

### Штаб-квартира Delta IABG

Delta Electronics, Inc.  
Taoyuan Technology Center  
No.18, Xinglong Rd., Taoyuan City,  
Taoyuan County 33068, Taiwan  
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

### Азия

**Delta Electronics (Jiangsu) Ltd.**  
Wujiang Plant 3  
1688 Jiangxing East Road,  
Wujiang Economic Development Zone  
Wujiang City, Jiang Su Province, P.R.C. 215200  
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

**Delta Greentech (China) Co., Ltd.**  
238 Min-Xia Road, Pudong District,  
Shanghai, P.R.C. 201209  
TEL: 86-21-58635678 / FAX: 86-21-58630003

**Delta Electronics (Japan), Inc.**  
Tokyo Office  
2-1-14 Minato-ku Shibadaimon,  
Tokyo 105-0012, Japan  
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

**Delta Electronics (Korea), Inc.**  
1511, Byucksan Digital Valley 6-cha, Gasan-dong,  
Geumcheon-gu, Seoul, Korea, 153-704  
TEL: 82-2-515-5303 / FAX: 82-2-515-5302

**Delta Electronics Int'l (S) Pte Ltd.**  
4 Kaki Bukit Ave 1, #05-05, Singapore 417939  
TEL: 65-6747-5155 / FAX: 65-6744-9228

**Delta Electronics (India) Pvt. Ltd.**  
Plot No 43 Sector 35, HSIDC  
Gurgaon, PIN 122001, Haryana, India  
TEL : 91-124-4874900 / FAX : 91-124-4874945

### Украина

ООО «Системы реального времени -  
Украина»  
ул.Святослава Хороброго, 29А  
Днепр, 49000  
Тел.: 380-56-239-22-23 / Факс: 380 56 232 47 59

### Америка

**Delta Products Corporation (USA)**  
Raleigh Office  
P.O. Box 12173,5101 Davis Drive,  
Research Triangle Park, NC 27709, U.S.A.  
TEL: 1-919-767-3800 / FAX: 1-919-767-8080

**Delta Greentech (Brasil) S.A.**  
Sao Paulo Office  
Rua Itapeva, 26 - 3° andar Edificio Itapeva One-Bela Vista  
01332-000-São Paulo-SP-Brazil  
TEL: 55 11 3568-3855 / FAX: 55 11 3568-3865

### Европа

**Delta Electronics (Netherlands) B.V.**  
Eindhoven Office  
De Witbogt 20, 5652 AG Eindhoven, The Netherlands  
TEL : +31 (0)40-8003800 / FAX : +31 (0)40-8003898

Руководство по программированию ПЛК серии AS



## Серия AS Руководство по программированию

# Содержание

## Глава 1 Введение

<b>1.1 Общая информация</b> .....	<b>1-2</b>
1.1.1    Руководства .....	1-2
1.1.2    Аппаратные средства .....	1-2
<b>1.2 Программное обеспечение</b> .....	<b>1-6</b>
1.2.1    Редактирование программ .....	1-6
1.2.2    Программные средства.....	1-7

## Глава 2 Программные объекты

<b>2.1 Описание объектов</b> .....	<b>2-2</b>
2.1.1    Таблица объектов .....	2-2
2.1.2    Базовая структура сохранения входов/выходов .....	2-3
2.1.3    Связь между типом объекта и операцией ПЛК .....	2-3
2.1.4    Энергонезависимые области в диапазоне объектов.....	2-4
<b>2.2. Функции объектов</b> .....	<b>2-5</b>
2.2.1    Значения и константы.....	2-5
2.2.2    Числа с плавающей запятой .....	2-7
2.2.3    Строковые переменные.....	2-8
2.2.4    Входные реле (X) .....	2-10
2.2.5    Выходные реле (Y) .....	2-10
2.2.6    Вспомогательные реле (M) .....	2-11
2.2.7    Специальные вспомогательные реле (SM) .....	2-11
2.2.8    Условия обновления состояния специальных вспомогат. реле.....	2-43
2.2.9    Шаговые реле (S) .....	2-52
2.2.10   Таймеры (T) .....	2-52
2.2.11   Счетчики.....	2-54
2.2.12   32-битные счетчики (HC) .....	2-55

2.2.13	Регистры данных (D) .....	2-57
2.2.14	Специальные регистры данных (SR) .....	2-57
2.2.15	Условия обновления специальных регистров данных .....	2-84
2.2.16	Дополнительная информация по SM и SR .....	2-87
2.2.17	Индексные регистры (E) .....	2-98
2.2.18	Фаловые регистры (FR) .....	2-98

### **Глава 3 Сводная таблица инструкций**

<b>3.1</b>	<b>Инструкции</b> .....	<b>3-2</b>
3.1.1	Базовые инструкции .....	3-2
3.1.2	Прикладные инструкции .....	3-2
<b>3.2</b>	<b>Сводная таблица инструкций</b> .....	<b>3-3</b>
3.2.1	Базовые инструкции .....	3-3
3.2.2	Прикладные инструкции (Сортировка по номерам) .....	3-4
3.2.3	Прикладные инструкции (Сортировка по алфавиту) .....	3-4
3.2.4	Таблица объектов .....	3-5
<b>3.3</b>	<b>Таблица базовых инструкций</b> .....	<b>3-6</b>
<b>3.4</b>	<b>Таблица прикладных инструкций</b> .....	<b>3-8</b>
3.4.1	Прикладные инструкции (сортировка по номерам) .....	3-8
3.4.2	Прикладные инструкции (сортировка по алфавиту) .....	3-36

### **Глава 4 Структура инструкций**

<b>4.1</b>	<b>Состав прикладных инструкций API</b> .....	<b>4-2</b>
<b>4.2</b>	<b>Описание использования операндов</b> .....	<b>4-5</b>
<b>4.3</b>	<b>Ограничения на применение инструкций</b> .....	<b>4-6</b>
<b>4.4</b>	<b>Индексные регистры</b> .....	<b>4-7</b>
<b>4.5</b>	<b>Регистры метки</b> .....	<b>4-9</b>

<b>4.6</b>	<b>Регистры метки таймеров</b> .....	<b>4-11</b>
<b>4.7</b>	<b>Регистры метки 16-битного счетчика</b> .....	<b>4-12</b>
<b>4.8</b>	<b>Регистры метки 32-битного счетчика</b> .....	<b>4-14</b>
<b>4.9</b>	<b>Файловые регистры</b> .....	<b>4-15</b>

## **Глава 5 Базовые инструкции**

<b>5.1</b>	<b>Таблица базовых инструкций</b> .....	<b>5-2</b>
<b>5.2</b>	<b>Описание базовых инструкций</b> .....	<b>5-3</b>

## **Глава 6 Прикладные инструкции**

<b>6.1</b>	<b>Инструкции сравнения</b> .....	<b>6-3</b>
6.1.1	Описание инструкций сравнения .....	6-3
<b>6.2</b>	<b>Арифметические инструкции</b> .....	<b>6-43</b>
6.2.1	Описание арифметических инструкций .....	6-43
<b>6.3</b>	<b>Инструкции преобразования данных</b> .....	<b>6-74</b>
6.3.1	Описание инструкций преобразования данных .....	6-74
<b>6.4</b>	<b>Инструкции передачи данных</b> .....	<b>6-115</b>
6.4.1	Описание инструкций передачи данных .....	6-115
<b>6.5</b>	<b>Инструкции перехода</b> .....	<b>6-143</b>
6.5.1	Описание инструкций перехода .....	6-143
<b>6.6</b>	<b>Исполняемые программой инструкции</b> .....	<b>6-151</b>
6.6.1	Описание исполняемых программой инструкций .....	6-151
<b>6.7</b>	<b>Инструкции обновления входов/выходов</b> .....	<b>6-163</b>
6.7.1	Описание инструкций обновления входов/выходов .....	6-163
<b>6.8</b>	<b>Вспомогательные инструкции</b> .....	<b>6-168</b>
6.8.1	Описание вспомогательных инструкций .....	6-168
<b>6.9</b>	<b>Логические инструкции</b> .....	<b>6-216</b>
6.9.1	Описание логических инструкций .....	6-216

<b>6.10 Инструкции вращения</b> .....	<b>6-237</b>
6.10.1 Описание инструкций вращения .....	6-237
<b>6.11 Инструкции таймеров и счетчиков</b> .....	<b>6-248</b>
6.11.1 Описание инструкций таймеров и счетчиков .....	6-248
<b>6.12 Инструкции сдвига</b> .....	<b>6-283</b>
6.12.1 Описание инструкций сдвига .....	6-283
<b>6.13 Инструкции по обработке данных</b> .....	<b>6-320</b>
6.13.1 Описание инструкций по обработке данных .....	6-320
<b>6.14 Инструкции по созданию структуры</b> .....	<b>6-377</b>
6.14.1 Описание инструкций по созданию структуры .....	6-377
<b>6.15 Инструкции модулей</b> .....	<b>6-385</b>
6.15.1 Описание инструкций модулей.....	6-385
<b>6.16 Инструкции значений с плавающей запятой</b> .....	<b>6-420</b>
6.16.1 Описание инструкций значения с плавающей запятой .....	6-420
<b>6.17 Инструкции для часов реального времени</b> .....	<b>6-455</b>
6.17.1 Описание инструкций для часов реального времени.....	6-455
<b>6.18 Инструкции периферийных устройств</b> .....	<b>6-485</b>
6.18.1 Описание инструкций периферийных устройств .....	6-485
<b>6.19 Инструкции коммуникации</b> .....	<b>6-501</b>
6.19.1 Описание инструкций коммуникации .....	6-501
6.19.2 Описание флагов и регистров для коммуникации .....	6-584
<b>6.20 Прочие инструкции</b> .....	<b>6-587</b>
6.20.1 Описание прочих инструкций.....	6-587
<b>6.21 Инструкции по обработке строковых переменных</b> .....	<b>6-599</b>
6.21.1 Описание инструкций по обработке строковых переменных.....	6-599
<b>6.22 Инструкции для Ethernet</b> .....	<b>6-659</b>
6.22.1 Описание инструкций для Ethernet .....	6-659
<b>6.23 Инструкции для карты памяти</b> .....	<b>6-701</b>

6.23.1	Описание инструкций для карты памяти .....	6-701
<b>6.24</b>	<b>Инструкции управления задачами .....</b>	<b>6-718</b>
6.24.1	Описание инструкций управления задачами .....	6-718
<b>6.25</b>	<b>Инструкции SFC .....</b>	<b>6-722</b>
6.25.1	Описание инструкций SFC.....	6-722
<b>6.26</b>	<b>Инструкции высокочастотных выходов .....</b>	<b>6-729</b>
6.26.1	Описание инструкций высокочастотных выходов.....	6-729
<b>6.27</b>	<b>Инструкции коммуникации CANopen.....</b>	<b>6-825</b>
6.27.1	Описание инструкций коммуникации CANopen .....	6-825

## **Глава 7 Поиск и устранение неисправностей**

<b>7.1</b>	<b>Поиск неисправностей .....</b>	<b>7-3</b>
7.1.1	Основные шаги по поиску неисправностей .....	7-3
7.1.2	Сброс ошибок.....	7-3
7.1.3	Блок-схема устранения неисправностей .....	7-4
7.1.4	Функция System Log .....	7-5
<b>7.2</b>	<b>Поиск неисправностей для модулей ЦПУ .....</b>	<b>7-6</b>
7.2.1	Светодиодный индикатор ERROR включен .....	7-6
7.2.2	Светодиодный индикатор ERROR мигает с частотой 0,5 сек .....	7-6
7.2.3	Светодиодный индикатор ERROR мигает с частотой 0,2 сек .....	7-8
7.2.4	Светодиодный индикатор ERROR медленно мигает каждые 3 сек, загораясь на 1 сек .....	7-8
7.2.5	Светодиодный индикатор BAT. LOW включен .....	7-8
7.2.6	Светодиодный индикатор BAT. LOW мигает с частотой 0,5 сек.....	7-8
7.2.7	Светодиодные индикаторы RUN и ERROR одновременно мигают с частотой 0,5 сек. ....	7-9
7.2.8	Светодиодные индикаторы RUN и ERROR попеременно мигают с частотой 0,5 сек. ....	7-9
7.2.9	Прочие ошибки (без светодиодной индикации) .....	7-9
<b>7.3</b>	<b>Поиск неисправностей для модулей входов/выходов .....</b>	<b>7-16</b>
7.3.1	Поиск неисправностей для модулей аналоговых входов/выходов (AD/DA/XA) и температурных модулей (RTD/TC) .....	7-16
7.3.2	Поиск неисправностей для весового модуля AS02LC .....	7-17
7.3.3	Поиск неисправностей для модуля коммуникации AS00SCM .....	7-17

7.3.4	Поиск неисправностей для удаленного модуля AS00SCM.....	7-19
<b>7.4</b>	<b>Коды ошибок и светодиодная индикация для модулей ЦПУ.....</b>	<b>7-20</b>
7.4.1	Коды ошибок и светодиодная индикация для модулей ЦПУ .....	7-21
7.4.2	Коды ошибок и светодиодная индикация для модулей аналоговых входов / выходов и температурных модулей.....	7-26
7.4.3	Коды ошибок и светодиодная индикация для весового модуля AS02LC .....	7-26
7.4.4	Коды ошибок и светодиодная индикация для модуля коммуникации AS00SCM.....	7-27
7.4.5	Коды ошибок и светодиодная индикация для удаленного модуля AS00SCM.....	7-27

Перевод, адаптация, обновление выполнены компанией «СТОИК»  
тел. (495) 661-24-41 / 661-24-61, <http://www.deltronics.ru>

---

# Глава 1 Введение

## Содержание

<b>1.1 Общая информация .....</b>	<b>1-2</b>
1.1.1 Руководства .....	1-2
1.1.2 Аппаратные средства.....	1-2
<b>1.2 Программное обеспечение .....</b>	<b>1-6</b>
1.2.1 Редактирование программ.....	1-6
1.2.2 Программные средства .....	1-8



## 1.1 Общая информация

В данном Руководстве рассматриваются вопросы программирования контроллеров серии AS и описываются операнды и прикладные инструкции для программирования.

Перевод, адаптация и обновление выполнены компанией СТОИК (тел. (495) 661-24-41, <http://www.deltronics.ru>)

### 1.1.1 Руководства

Delta Electronics предоставляет пользователям руководства по эксплуатации ПЛК серии AS в соответствии с различными требованиями применений.

- **Руководство по аппаратной части AS:** Знакомит с аппаратными спецификациями, адресацией, электромонтажом, техническим обслуживанием, поиском и устранением неисправностей и т.п.
- **Руководство по эксплуатации AS:** Знакомит с конфигурацией аппаратных средств, настройкой соединения, работой модуля CPU, настройкой программного обеспечения и т.д.
- **Руководство по программированию AS:** Знакомит с программными объектами и инструкциями.
- **Руководство на модули расширения для AS:** Знакомит со спецификациями модулей, установкой, настройкой, поиском и устранением неисправностей и т.д.
- **Руководство пользователя ISPSOft:** Знакомит с использованием среды программирования ISPSOft.
- **Quick Start (Быстрый старт) AS:** Помогает пользователям за короткое время создавать и использовать систему. **Quick Start AS** не только знакомит пользователей с основной структурой системы, но и учит их с помощью простых примеров пошагово писать и загружать программу в модуль CPU, включающую в себя переменные и функциональные блоки. Пользователи смогут испытать удобство, которые содержат новые функции. Если во время работы системы возникла ошибка, обратитесь к разделу 3.3.2 для получения дополнительной информации об отладке программы. (Для получения дополнительной информации о поиске неисправностей обратитесь, пожалуйста, к главе 12 Руководства по эксплуатации AS).

### 1.1.2 Аппаратные средства

Наименование	Обозначение	Описание
Модуль питания	AS-PS02	Вход: 100~240 В AC, 50/60 Гц Выход: 24 В DC / 2 А, 48 Вт (только питание ПЛК)
	AS-PS02A	Вход: 100~240 В AC, 50/60 Гц Выход: 24 В DC / 1,5 А, 36 Вт (только питание ПЛК) Выход: 24 В DC / 0,5 А, 12 В (питание внешних устройств)
Модули CPU	AS332P-A	Модуль CPU, выход PNP, 2x RS-485, 1x USB, 1x MicroSD, 2 функциональных платы (опция), 16DI/16DO, расширение до 1024 I/O, объем программы:128К шагов
	AS332T-A	Модуль CPU, выход NPN, 1x Ethernet, 2x RS-485, 1x USB, 1x MicroSD, 2 функциональных платы (опция), 16DI/16DO, расширение до 1024 I/O, объем программы:128К шагов
	AS324MT-A	Модуль CPU, выход NPN, 1x Ethernet, 2x RS-485, 1x USB, 1x MicroSD, 2 функциональных платы (опция), 12DI/12DO, расширение до 1024 I/O, объем программы:128К шагов
Модули дискретных входов/выходов	AS08AM10N-A	24 В DC 5 мА 8 входов Пружинный клеммный блок

Наименование	Обозначение	Описание
	AS08AN01P-A	5 ~ 30 В DC 0.5 А 8 выходов Выходы: Sourcing Пружинный клеммный блок
	AS08AN01R-A	240 В AC / 24 В DC 2 А 8 выходов Выходы: Реле Пружинный клеммный блок
	AS08AN01T-A	5 ~ 30VDC 0.5А 8 выходов Выходы: NPN Пружинный клеммный блок
	AS16AM10N-A	24 В DC 5 мА 16 входов Пружинный клеммный блок
	AS16AN01P-A	5 ~ 30 В DC 0.5 А 16 выходов Выходы: PNP Пружинный клеммный блок
	AS16AN01R-A	240 В AC / 24 В DC 2 А 16 выходов Выходы: Реле Пружинный клеммный блок
	AS16AN01T-A	5 ~ 30VDC 0.5А 16 выходов Выходы: NPN Пружинный клеммный блок
	AS16AP11P-A	24 В DC 5 мА 8 входов 5 ~ 30 В DC 0.5 А 8 выходов Выходы: PNP Пружинный клеммный блок
	AS16AP11R-A	24VDC 5mA 8 входов 240 В AC/24 В DC 2А 8 выходов Выходы: Реле Пружинный клеммный блок

Наименование	Обозначение	Описание
	AS16AP11T-A	24 В DC 5 мА 8 входов 5 ~ 30 В DC 0.5 А 8 выходов Выходы: NPN Пружинный клеммный блок
	AS32AM10N-A	24 В DC 3.2 мА 32 входов Разъем IDC-40
	AS32AN02T-A	5 ~ 30 В DC 0.1 А 32 выхода Выходы: NPN Разъем IDC-40
	AS64AM10N-A	24 В DC 3.2 мА 64 входом Разъем IDC-40
	AS64AN02T-A	5 ~ 30 В DC 0.1 А 64 выходов Выходы: NPN Разъем IDC-40
Модули аналоговых входов/ выходов	AS04AD-A	4-канальный модуль аналоговых входов Аппаратное разрешение: 16 бит 0~10 В, 0/1~5 В, -5~+5 В, -10~+10 В, 0/4~20 мА, -20~+20 мА Время преобразования: 2 мс/канал
	AS04DA-A	4-канальный модуль аналоговых выходов Аппаратное разрешение: 12 бит -10~+10 В, 0~20 мА, 4~20 мА Время преобразования: 2 мс/канал
	AS06XA-A	4-канальный модуль аналоговых входов Аппаратное разрешение: 16 бит 0~10 В, 0/1~5 В, -5~+5 В, -10~+10 В, 0/4~20 мА, -20~+20 мА Время преобразования: 2 мс/канал 2-канальный модуль аналоговых выходов Аппаратное разрешение: 12 бит -10~+10 В, 0~20 мА, 4~20 мА Время преобразования: 2 мс/канал
Температурные модули	AS04RTD-A	4-канальный, 2/3-проводные термодатчики Тип: Pt100 / Ni100 / Pt1000 / Ni1000 / JPt100 / LG-Ni1000 / Cu50 / Cu100 / 0~300Ω / 0~3000Ω входной импеданс Разрешение: 0.1°C/0.1°F (16 бит) Время преобразования: 200 мс/канал
	AS04TC-A	4-канальный, термопары Тип: J, K, R, S, T, E, N, B и -100~+100 мВ Разрешение: 0.1°C/0.1°F (24 бит) Время преобразования: 200 мс/канал

Наименование	Обозначение	Описание
Весовой модуль	AS02LC-A	2-канальный, 4/6-проводной датчик Собственное значение датчика: 1,2,4,6,20,40,80 мВ/В Высокоточный 1/10000 за цикл 50 мс Разрешение АЦП: 24 бит Время преобразования: 2.5 ~ 400 мс (9 вариантов выбора)
Сетевой модуль	AS00SCM-A	Модуль последовательной связи, 2 порта связи, применимы коммуникационные платы, поддержка протокола MODBUS
Модуль удаленных входов/выходов	AS00SCM-A + AS-FCOPM	Применяется совместно с функциональной платой AS-FCOPM
Функциональные платы	AS-F232	Коммуникационный порт, RS232, работает в режимах master или slave
	AS-F422	Коммуникационный порт, RS422, работает в режимах master или slave
	AS-F485	Коммуникационный порт, RS485, работает в режимах master или slave
	AS-FCOPM	Порт CANopen, поддержка DS301, модулей удаленного управления AS и сервоприводов Delta
	AS-F2AD	2-канальная плата аналоговых входов 0~10 В (12 бит), 4~20 мА (11 бит) Время преобразования: 3 мс/канал
	AS-F2DA	2-канальная плата аналоговых входов 0~10 В, 4~20 мА (12 бит) Время преобразования: 2 мс/канал
Кабели для программирования	UC-PRG015-01A (1.5M)	Для соединения ПЛК и ПК через mini USB порт, применяется с AS332T-A, AS332P-A, AS324MT-A
	UC-PRG030-01A (3M)	Для соединения ПЛК и ПК через mini USB порт, применяется с AS332T-A, AS332P-A, AS324MT-A
	UC-PRG030-20A (3M)	Для соединения ПЛК и ПК через порт RJ45, применяется с AS332T-A, AS332P-A, AS324MT-A
Кабели входов / выходов	UC-ET010-24B (1M) UC-ET020-24B (2M) UC-ET030-24B (3M)	IDC-40, 40-Pin ↔ 40-Pin, экранированный, применяется с AS32AM10N-A, AS32AN02T-A, AS64AM10N-A, AS64AN02T-A
	UC-ET010-24D (1M) UC-ET020-24D (2M) UC-ET030-24D (3M)	IDC-40, 40-Pin ↔ 40-Pin, экранированный, применяется с AS332T-A, AS332P-A, AS324MT-A, AS32AM10N-A, AS32AN02T-A, AS64AM10N-A, AS64AN02T-A
Внешние клеммные блоки	UB-10-ID16A	16 входов/выходов, разъем 20-Pin IDC-40, применяется с AS332T-A, AS332P-A, AS324MT-A, AS32AM10N-A, AS32AN02T-A, AS64AM10N-A, AS64AN02T-A
	UB-10-ID32A	32 входов/выходов, разъем 40-Pin IDC-40, применяется с AS32AM10N-A, AS64AM10N-A
	UB-10-OR16A	16 релейных выходов, разъем 20-Pin IDC-40, NPN, применяется с AS332T-A, AS32AN02T-A, AS64AN02T-A
	UB-10-OR16B	16 релейных выходов, разъем 20-Pin IDC-40, PNP, применяется с AS332P-A
	UB-10-OT32A	32 транзисторных выхода, разъем 40-Pin IDC-40, NPN, применяется с AS32AN02T-A, AS64AN02T-A
Сетевые кабели	UC-CMC003-01A (0.3M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC005-01A (0.5M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC010-01A (1M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC015-01A (1.5M)	Кабель связи CANopen, применяется с AS-FCOPM

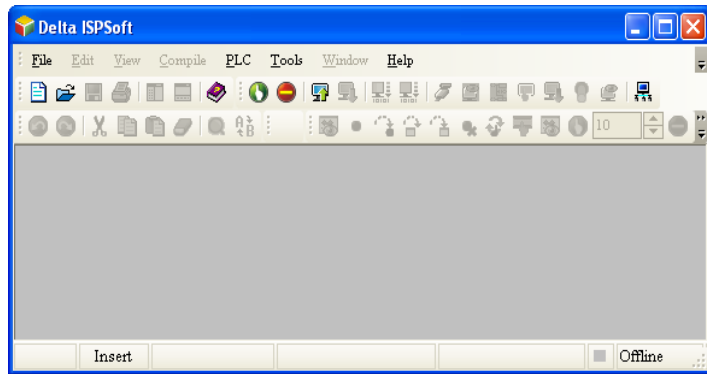
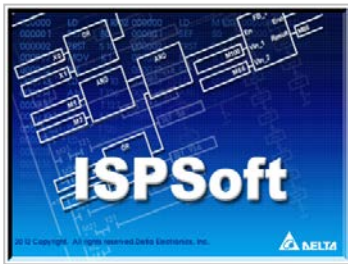
1

Наименование	Обозначение	Описание
	UC-CMC020-01A (2M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC030-01A (3M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC050-01A (5M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC100-01A (10M)	Кабель связи CANopen, применяется с AS-FCOPM
	UC-CMC200-01A (20M)	Кабель связи CANopen, применяется с AS-FCOPM

## 1.2 Программное обеспечение

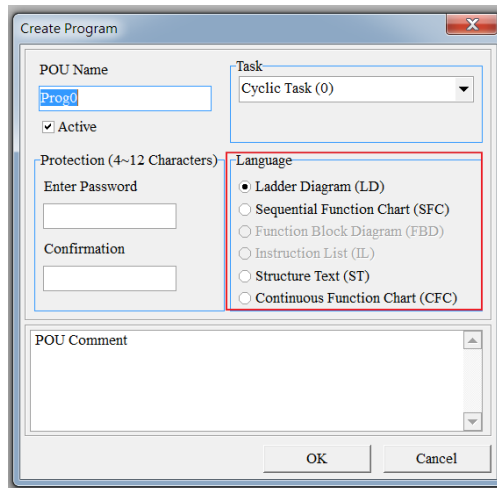
### 1.2.1 Редактирование программ

Запуск ПО для создания и редактирования программ ISPSoft:

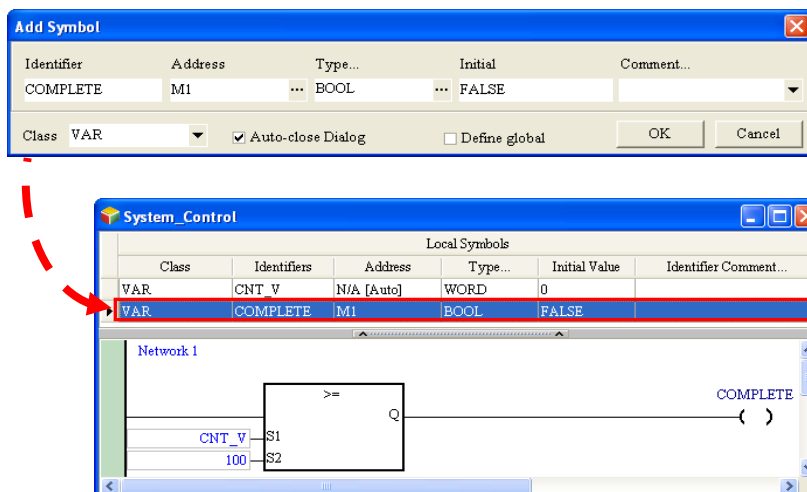


- ПЛК серии AS поддерживают 4 языка программирования: Structure Text (ST), Ladder Diagram (LD), Sequential Function Chart (SFC) и Continuous Function Chart (CFC).

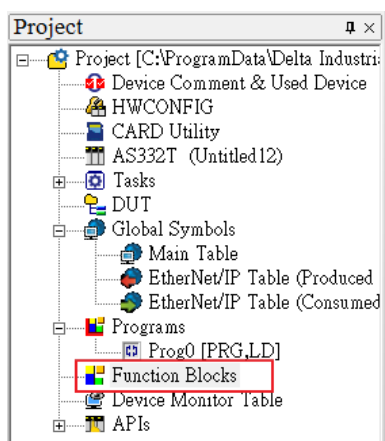
ПРИМЕЧАНИЕ: язык CFC доступен в версии ПО ISPSoft 3.01 и выше.



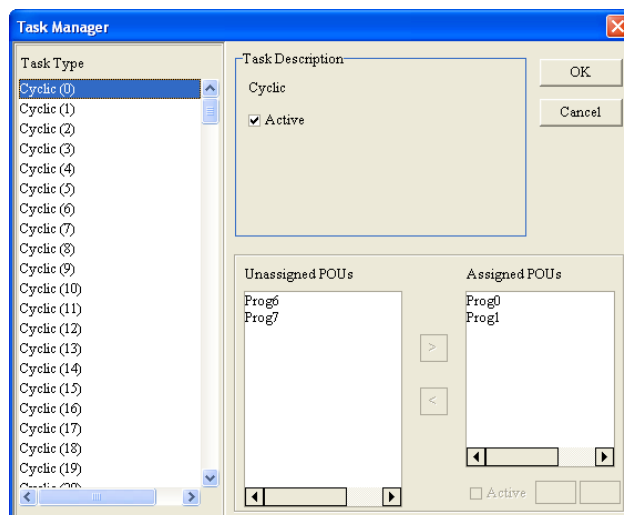
- Использование переменных, которые позволяют пользователю определять символ для замены имени устройства ПЛК, не только улучшает читаемость программы, но и экономит пользователю много времени при задании адреса устройства.



- Внедрение модуля POU (Program Organization Unit) не только делит основную программу на несколько программных единиц, но также заменяет традиционные подпрограммы функциями и функциональными блоками. Структура программы становится более модульной и более управляемой.



- Концепция задач, которые используются для управления порядком выполнения программ, расширяет разработку программы до уровня управления проектами, при этом значительно упрощается управление программами, имеющими значительный объем.



## 1.2.2 Программные средства

**1**

Модули POU (Program Organization Units) являются базовыми элементами программы ПЛК. В отличие от традиционных программ для ПЛК, согласно стандарту IEC 61131-3 объемная программа может быть разделена на меньшие элементы. Этими элементами являются модули POU. Модули POU могут быть трех типов.

1. Программа (PROG): Модуль POU данного типа. Разработчик может задать для выполнения модуля POU типа Программа для циклического выполнения или прерывание и упорядочить сканирование в списке задач для модулей POU типа Программа.
2. Функциональный блок (FB): Назначение функционального блока (FB) аналогично назначению подпрограммы. Программа, записанная внутри функционального блока, выполняется после того, как функциональный блок вызывается модулем POU типа Программа и вводятся соответствующие параметры.
3. Функция (FC): Назначение функции (FC) близко к назначению макрокоманды. Пользователи могут записывать значительное число прикладных инструкций или программных функций в данный модуль POU, а затем использовать их в модулях POU типа Программа или Функциональный блок.

Задача - это функция, которая предусматривает, что программа выполняется в определенном порядке или в соответствии с определенным условием прерывания. Смысл задачи заключается в том, что она предоставляет каждому модулю POU типа Программа конкретный алгоритм выполнения, задает порядок выполнения для модуля POU типа Программа или способ его активизации.

Во многих случаях, не все модули POU типа Программа будут выполняться в проекте. Выполнять ли данный модуль POU или нет, а также, каким образом его выполнять, зависит от назначенной задачи. Если для модуля POU типа Программа задача не назначена, он будет сохранен как обычный исходный код проекта вместо компиляции в качестве кода выполнения в ПЛК. Кроме того, необходимо обозначить только модули POU типа Программа. Выполнение функциональных блоков или функций зависит от приоритетного модуля POU, который к ним обращается. Существует три типа задач.

1. Циклическая задача: Модули POU типа Программа, которым назначена задача данного типа, будут циклически сканироваться и выполняться по порядку.
2. Задача прерывания по времени: Если время прерывания достигнуто, все модули POU типа Программа, которым назначена задача прерывания, будут выполняться по порядку.
3. Задача условного прерывания: Условное прерывание можно разделить на несколько типов. Например, внешнее прерывание, прерывание по сигналу на входах/выходах и т. д. Пользователь должен знать типы прерываний, поддерживаемые ПЛК, до того, как будет создан проект. Если модулю POU типа Программа назначается задача условного прерывания, функция модуля POU будет аналогична подпрограмме прерывания. Если условие прерывания выполнено, например, на клемме внешнего прерывания есть сигнал, все модули POU типа Программа, которым назначены задачи данного типа, будут выполняться по порядку.

## Глава 2 Программные объекты

### Содержание

<b>2.1 Описание объектов</b> .....	<b>2-2</b>
2.1.1 Таблица объектов .....	2-2
2.1.2 Базовая структура сохранения входов/выходов .....	2-3
2.1.3 Связь между типом объекта и операцией ПЛК .....	2-3
2.1.4 Энергонезависимые области в диапазоне объектов .....	2-4
<b>2.2. Функции объектов</b> .....	<b>2-5</b>
2.2.1 Значения и константы .....	2-5
2.2.2 Числа с плавающей запятой .....	2-7
2.2.3 Строковые переменные .....	2-8
2.2.4 Входные реле (X) .....	2-10
2.2.5 Выходные реле (Y) .....	2-10
2.2.6 Вспомогательные реле (M) .....	2-11
2.2.7 Специальные вспомогательные реле (SM) .....	2-11
2.2.8 Условия обновления состояния специальных вспомогательных реле .....	2-54
2.2.9 Шаговые реле (S) .....	2-63
2.2.10 Таймеры (T) .....	2-63
2.2.11 Счетчики .....	2-65
2.2.12 32-битные счетчики (HC) .....	2-66
2.2.13 Регистры данных (D) .....	2-68
2.2.14 Специальные регистры данных (SR) .....	2-69
2.2.15 Условия обновления специальных регистров данных .....	2-101
2.2.16 Дополнительная информация по SM и SR .....	2-104
2.2.17 Индексные регистры (E) .....	2-116
2.2.18 Фаловые регистры (FR) .....	2-117



## 2.1 Описание объектов

В данном Разделе представлены программные объекты и средства – специальные реле, специальные регистры данных, основные и прикладные инструкции для программирования ПЛК серии AS.

### 2.1.1 Таблица объектов

Тип	Наименование объекта		Количество объектов	Диапазон
<b>Битовые объекты</b>	Входное реле	X	1024	X0.0~X63.15
	Выходное реле	Y	1024	Y0.0~Y63.15
	Регистр данных	D	48,0000	D0.0~D29999.15
		W	48,0000	W0.0~W29999.15 *4
	Вспомогательное реле	M	8192	M0~M8191
	Специальное вспомогательное реле	SM	2048	SM0~SM4095
	Шаговое реле	S	2048	S0~S2047
	Таймер	T	512	T0~T511
	Счетчик	C	512	C0~C511
32-битный счетчик	HC	256	HC0~HC255	
<b>Словные объекты</b>	Входное реле	X	64	X0~X63
	Выходное реле	Y	64	Y0~Y63
	Регистр данных	D	30000	D0~D29999
		W	30000	W0~W29999 *4
	Специальное вспомогательное реле	SR	2048	SR0~SR2047
	Файловый регистр	FR	65536	FR0~FR65535
	Таймер	T	512	T0~T511
	Счетчик	C	512	C0~C511
	32-битный счетчик	HC	256 ( 512 words )	HC0~HC255
Индексный регистр	E	10	E0~E9	
		5	E10~E14 *4	
<b>Константы</b>	Десятичный формат	K	16 бит: -32768~32767 32 бит: -2147483648~2147483647	
<b>Константы</b>	Шестнадцатеричный формат	16#	16 бит: 16#0~16#FFFF 32 бит: 16#0~16#FFFFFFFF	
	С плавающей точкой	F	32 бит: $\pm 1.17549435^{-38}$ ~ $\pm 3.40282347^{+38}$	
<b>Символьная переменная</b>	Символьная переменная	“\$”	1~31 символов	

\*1: В списках объектов в Главах 5 и 6 настоящего Руководства по программированию десятичные значения обозначаются буквой K. Например, для K50 в ISPSOft может быть задан только номер 50.

\*2: В списках объектов в Главах 5 и 6 настоящего Руководства по программированию числа с плавающей запятой обозначаются F / DF, а в ISPSOft они представлены десятичными точками; для F500 с плавающей запятой следует ввести 500,0.

\*3: В Главах 5 и 6 настоящего Руководства по программированию строки обозначаются символом «\$», например, в ISPSOft для строки 1234, следует ввести «1234».

\*4: Используется только для редактирования в ISPSOft.

### 2.1.2 Базовая структура сохранения входов/выходов

Объект	Функция	Доступность битов	Доступность слов	Модифицирование ISPSoft	Принудительное включение / выключение битов
X	Входное реле	ОК	ОК	ОК	ОК
Y	Выходное реле	ОК	ОК	ОК	ОК
M	Вспомогательное реле	ОК	-	ОК	-
SM	Специальное вспомогательное реле	ОК	-	ОК	-
S	Шаговое реле	ОК	-	ОК	-
T	Таймер	ОК	ОК	ОК	-
C	Счетчик	ОК	ОК	ОК	-
HC	32-битный счетчик	ОК	ОК	ОК	-
D	Регистр данных	ОК	ОК	ОК	ОК
SR	Специальный регистр данных	-	ОК	ОК	-
FR	Файловый регистр	-	ОК* <sup>1</sup>	-	-
E	Индексный регистр	-	ОК	ОК	-

\*1: Используется для записи инструкции в FR.

### 2.1.3 Связь между типом объекта и операцией ПЛК

Операция ПЛК \ Тип объекта		Энергозависимые		Энергонезависимые	
		Объект Y	Другие	Файловый регистр	Другие
Питание: ВЫКЛ→ВКЛ		Очистка	Очистка	Сохранение	Сохранение
Сброс на заводские настройки		Очистка	Очистка	Очистка	Очистка
СТОП ↓ ПУСК* <sup>1</sup>	Очистка энергозависимой области памяти	Очистка	Очистка	Сохранение	Сохранение
	Состояние энергозависимой области памяти сохраняется	Сохранение	Сохранение	Сохранение	Сохранение
ПУСК ↓ СТОП* <sup>1</sup>	Очистка состояния объекта Y	Очистка	Сохранение	Сохранение	Сохранение

Операция ПЛК	Тип объекта	Энергозависимые		Энергонезависимые	
		Объект Y	Другие	Файловый регистр	Другие
	Состояние объекта Y сохраняется	Сохранение	Сохранение	Сохранение	Сохранение
	<b>SM204 ВКЛЮЧЕН</b> (очистка всех энергозависимых областей)	Очистка	Очистка	Сохранение	Сохранение
	<b>SM205 ВКЛЮЧЕН</b> (очистка всех энергонезависимых областей)	Сохранение	Сохранение	Сохранение	Очистка

\*1: Для задания состояния перейдите в раздел HWCONFIG в ПО ISPSOft. По умолчанию ПЛК СТОП -> ПУСК относится к «очищению энергозависимой области». По умолчанию ПЛК ПУСК -> СТОП относится к «очищению состояния объекта Y».

#### 2.1.4 Энергонезависимые области в диапазоне объектов

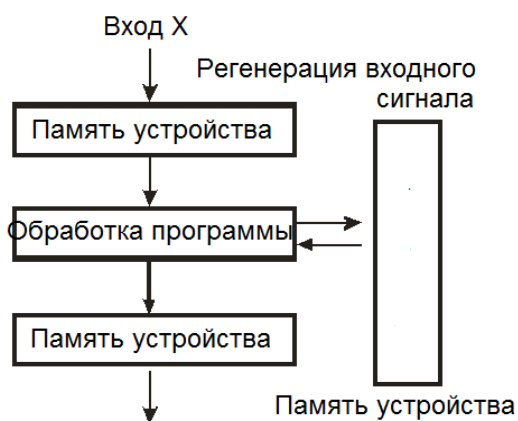
Объект	Функция	Диапазон	Энергонезависимая область
X	Входное реле	X0~X63	Нет
Y	Выходное реле	Y0~Y63	Нет
M* <sup>1</sup>	Вспомогательное реле	M0~M8191	По умолчанию M6000~M8191
SM	Специальное вспомогательное реле	SM0~SM2047	Некоторые объекты энергонезависимы и не могут быть изменены. Пожалуйста, обратитесь к перечню специальных вспомогательных реле для получения дополнительной информации
S* <sup>1</sup>	Шаговое реле	S0~S1023	По умолчанию S512~S1023
T	Таймер	T0~T511	Нет
C* <sup>1</sup>	Счетчик	C0~C511	По умолчанию C448~C511
HC* <sup>1</sup>	32-битный счетчик	HC0~HC255	По умолчанию HC128~HC255
D* <sup>1</sup>	Регистр данных	D0~D29999	По умолчанию D20000~D29999
		W0~W29999	* <sup>2</sup>
FR	Файловый регистр	FR0~FR65535	Все
SR	Специальный регистр данных	SR0~SR2047	Некоторые объекты энергонезависимы и не могут быть изменены. Пожалуйста, обратитесь к перечню специальных регистров данных для получения дополнительной информации
E	Индексный регистр	E0~E9	Нет
		E10~E14	* <sup>2</sup>

\*1: Для настройки области обратитесь к HWCONFIG в ПО ISPSOft. Длина области не должна превышать диапазон объекта.

\*2: Редактирование только в ПО ISPSOft.

## 2.2. Функции объектов

Процедура обработки сигнала в ПЛК:



Регенерация выходного сигнала и отправка сигнала на выход Y

- Регенерация входного сигнала
  1. Перед выполнением программы состояние внешнего входного записывается в специальную область памяти для входных сигналов.
  2. Когда программа выполнена, состояние входного сигнала в памяти не изменяется, даже если входной сигнал изменяется с состояния ВКЛ на ВЫКЛ или с ВЫКЛ на ВКЛ. Состояние входного сигнала будет обновлено при следующем сканировании.
- Обработка программы
 

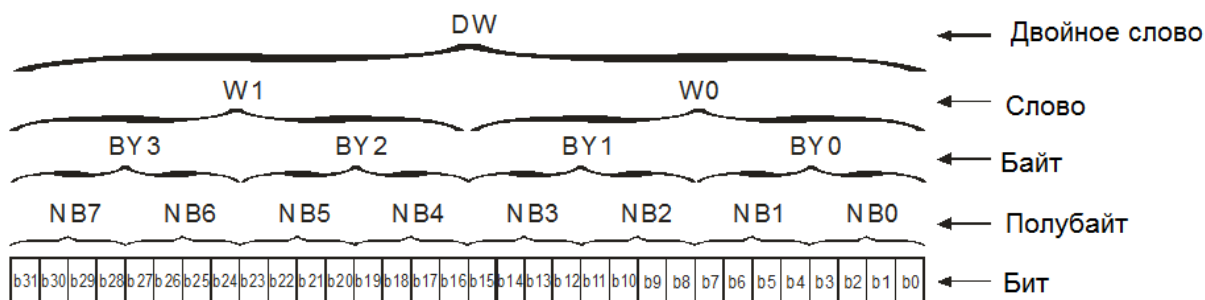
После обновления входного сигнала инструкции в программе выполняются в порядке от начального адреса программы, а результаты сохраняются в памяти устройства.
- Регенерация состояния выхода
 

После выполнения команды END состояние, сохраненное в памяти устройства, отправляется на указанный выход.

### 2.2.1 Значения и константы

Наименование	Описание
Бит	Бит - это базовый блок в двоичной системе. Его состояние равно 1 или 0
Полубайт	Полубайт состоит из четырех последовательных битов (например, b3 ~ b0). Полубайты можно использовать для представления чисел 0 ~ 9 в десятичной системе или чисел 0 ~ F в шестнадцатеричной системе
Байт	Байт состоит из двух последовательных полубайтов (т. е. 8 бит, b7 ~ b0). Байты могут использоваться для представления чисел 00 ~ FF в шестнадцатеричной системе
Слово	Слово состоит из двух последовательных байтов (то есть 16 бит, b15 ~ b0). Слова могут использоваться для представления чисел 0000 ~ FFFF в шестнадцатеричной системе
Двойное слово	Двойное слово состоит из двух последовательных слов (т. е. 32 бит, b31 ~ b0). Двойные слова могут использоваться для представления чисел 00000000 ~ FFFFFFFF в шестнадцатеричной системе

Соотношение между битами, полубайтами, байтами, словами и двойными словами в двоичной системе показано ниже.



ПЛК использует четыре типа значений для выполнения операций в соответствии с различными задачами:

1. Двоичные числа (BIN)

ПЛК использует двоичную систему для управления значениями.

2. Десятичные числа (DEC)

Десятичные числа в ПЛК используются как:

- Заданное значение таймера (Т) или счетчика (С/НС). Например, TMR C0 50 (**константа К**).
- Номер объекта. Например, M10 и T30 (номер объекта)
- Число перед или позади десятичной точки. Например, X0.0, Y0.11 и D10.0 (номера объектов).
- **Константа К**: Используется при выполнении прикладных инструкций. Например, MOV 123 D0 (**константа К**).

3. Двоично-десятичные значения (BCD)

Десятичное значение представлено полубайтом или четырьмя битами, и поэтому шестнадцать последовательных битов могут представлять четырехзначное десятичное значение.

4. Шестнадцатеричные числа (HEX)

Шестнадцатеричные числа в ПЛК используются как:

- **Константа 16#**: Используется при выполнении прикладных инструкций. Например, MOV 16#1A2B D0 (шестнадцатеричная константа).

Соответствие значений:

Двоичное (BIN)	Десятичное (DEC)	Двоично-десятичное (BCD)	Шестнадцатеричное (HEX)
Внутренние операции ПЛК	<b>Константа К</b> , Номер объекта	Соответствующие инструкции BCD	Константа 16#, Номер объекта
0000	0	0000	0
0001	1	0001	1
0010	2	0010	2
0011	3	0011	3
0100	4	0100	4
0101	5	0101	5
0110	6	0110	6
0111	7	0111	7

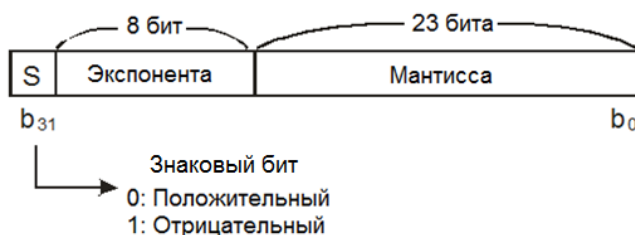
Двоичное (BIN)	Десятичное (DEC)	Двоично-десятичное (BCD)	Шестнадцатеричное (HEX)
1000	8	1000	8
1001	9	1001	9
1010	10	-	A
1011	11	-	B
1100	12	-	C
1101	13	-	D
1110	14	-	E
1111	15	-	F
10000	16	0001 0000	10
10001	17	0001 0001	11

### 2.2.2 Числа с плавающей запятой

Числа с плавающей запятой представлены десятичными точками в ISPSofT. Например, число с плавающей запятой 500 представляется как 500,0. Дополнительную информацию см. в разделе 2.2.2 настоящего Руководства.

#### 2.2.2.1 Число с плавающей запятой одинарной точности

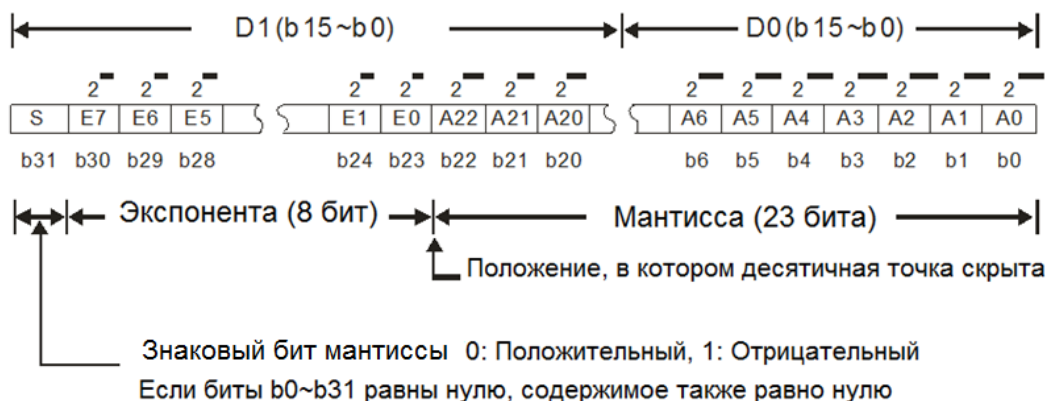
Число с плавающей запятой представлено 32-битным регистром. В представлении используется стандарт IEEE754, и формат выглядит следующим образом.



Уравнение:  $(-1)^S \times 2^{E-B} \times 1.M; B = 127$

Числа с плавающей запятой одинарной точности варьируются от  $\pm 2^{-126}$  до  $\pm 2 + 128$  и соответствуют диапазону от  $\pm 1.1755 \times 10^{-38}$  до  $\pm 3.4028 \times 10 + 38$ .

ПЛК серии AS использует два последовательных регистра для формирования 32-битного числа с плавающей запятой. Например (D1, D0).



2

**Пример 1:**

**23 представлено числом с плавающей запятой с одинарной точностью.**

**Шаг 1:** Преобразование 23 в двоичное число, т. е. 23.0 = 10111.

**Шаг 2:** Нормализация двоичного числа, то есть 10111 = 1.0111 × 2<sup>4</sup> (0111 - мантисса, а 4 - экспонента).

**Шаг 3:** Получение значения экспоненты.

$$\therefore E - B = 4 \rightarrow E - 127 = 4 \therefore E = 131 = 100000112$$

**Шаг 4:** Объедините знаковый бит, экспонента и мантиссу, чтобы сформировать число с плавающей запятой.

$$0\ 10000011\ 01110000000000000000_2 = 41B80000_{16}$$

**Пример 2:**

**-23 представляется числом с плавающей запятой с одинарной точностью.**

Шаги преобразования -23.0 в число с плавающей запятой те же, что и для преобразования 23.0 в число с плавающей запятой, за исключением того, что знаковый бит равен 1.

$$1\ 10000011\ 01110000000000000000_2 = C1B80000_{16}$$

**2.2.2.2 Десятичные числа с плавающей запятой**

- Поскольку числа с плавающей запятой с одинарной точностью и числа с плавающей запятой с двойной точностью трудно воспринимаются людьми, их можно преобразовать в десятичные числа с плавающей запятой, чтобы пользователю было легче оценить значения. Однако, в своей работе ПЛК использует именно числа с плавающей запятой с одинарной точностью и с двойной точностью.
- 32-битовое десятичное число с плавающей запятой представлено двумя последовательными регистрами. Константа хранится в регистре с меньшим номером, а экспонента хранится в регистре с большим номером. Например (D1, D0).

$$\text{Десятичное число с плавающей запятой} = [ \text{Константа D0} ]^{[ \text{Экспонента D1} ]} \cdot 10$$

Диапазон константы D0 = ±1,000 ~ ±9,999

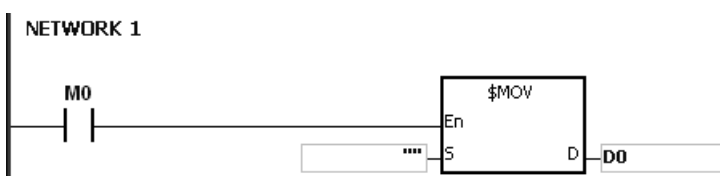
Диапазон экспоненты D1 = -41 ~ +35

Константа 100 не может существовать в диапазоне для D0, поэтому значение 100 представлено как 1000 × 10<sup>-1</sup>. Кроме того, 32-битовые десятичные числа с плавающей запятой варьируются в диапазоне от ± 1175 × 10<sup>-41</sup> до ± 402 × 10<sup>+35</sup>.

**2.2.3 Строковые переменные**

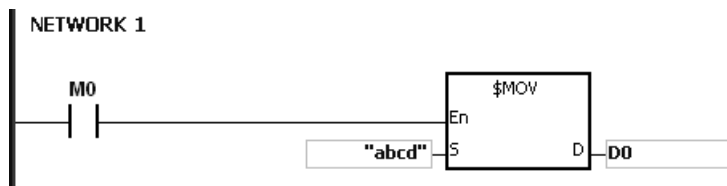
Строковые переменные могут обрабатывать коды ASCII (\*1). Полная строковая переменная начинается с символа начала и заканчивается символом окончания (NULL-код). Если вы вводите строковую переменную, вы можете ввести максимум 31 символ, и окончательный символ 16 # 00 будет добавлен в ISPSOft автоматически.

1. Строковая переменная без символов (NULL code) перемещена.



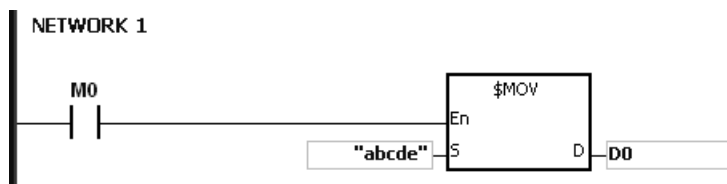
D0=0 (NULL)

2. Строковая переменная с четным номером.



D0	16#62 (b)	16#61 (a)
D1	16#64 (d)	16#63 (b)
D2	0 (NULL)	

3. Строковая переменная с нечетным номером.



D0	16#62 (b)	16#61 (a)
D1	16#64 (d)	16#63 (b)
D2	0 (NULL)	16#65 (e)

\*1: Таблица кодов ASCII

Hex*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Hex	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
ASCII	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒
Hex	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
ASCII	SP	!	"	#	\$	%	&	'	(	)	*	+	.	-	.	/
Hex	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
ASCII	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
Hex	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
ASCII	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Hex	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
ASCII	P	Q	R	S	T	U	V	W	X	Y	Z	☒	☒	☒	☒	☒
Hex	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
ASCII	`	a	b	c	d	e	f	g	h	i	j	k	l	M	n	o
Hex	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F



ASCII	p	q	r	s	t	u	v	w	x	y	z	{		}	~	☒
-------	---	---	---	---	---	---	---	---	---	---	---	---	--	---	---	---

Примечания: \*Hex – шестнадцатеричные символы,

☒ Не используется.

## 2.2.4 Входные реле (X)

### ● Функция входа

Вход подключается к устройству ввода (например, к внешним устройствам, таким как кнопочные переключатели, поворотные переключатели, цифровые переключатели и т. Д.), И входной сигнал считывается в ПЛК. Кроме того, контакт А или контакт В входа можно использовать несколько раз в программе, а состояние ВКЛ / ВЫКЛ входа зависит от состояния ВКЛ / ВЫКЛ устройства ввода.

### ● Номер входа (десятичное значение):

Для ПЛК номера входов начинаются с X0.0. Номера входов зависят от количества подключенных модулей дискретных входов / выходов, а нумеруются входы в соответствии с порядком, в котором модули дискретных входов / выходов подключены к модулю ЦПУ. Максимальное количество входов на ПЛК может достигать 8192 в диапазоне между X0.0 и X511.15.

### ● Тип входа

Входы подразделяются на два типа.

1. Регенерированный вход: перед выполнением программы данные подаются в ПЛК в соответствии с состояниями входов, которые регенерируются. Например, LD X0.0.
2. Прямой вход: во время выполнения инструкций данные подаются в ПЛК в соответствии с состояниями входов. Например, LD DX0.0.

## 2.2.5 Выходные реле (Y)

### ● Функция выхода

Выход посылает сигнал ВКЛ / ВЫКЛ нагрузку, подключенную к выходу. Нагрузка может быть внешней сигнальной лампой, цифровым дисплеем или электромагнитным клапаном. Существует четыре типа выходов. Это реле, транзисторы (NPN и PNP) и TRIAC (тиристоры). Контакт А или контакт В выхода можно использовать несколько раз в программе, но выход Y должен использоваться в программе только один раз. В противном случае, согласно принципу сканирования программы в ПЛК, состояние выхода зависит от нагрузки, подключенной к последнему выходу Y в программе.

### ● Номер выхода (десятичное значение)

Для ПЛК номера выходов начинаются с Y0.0. зависят от количества подключенных модулей дискретных входов / выходов, а нумеруются входы в соответствии с порядком, в котором модули дискретных входов / выходов подключены к модулю ЦПУ. Максимальное количество выходов на ПЛК может достигать 1024 в диапазоне между Y0.0 и Y63.15.

Выход, который практически не используется, может использоваться как общий.

### ● Тип выхода

Выходы подразделяются на два типа.

3. Регенерированный выход: перед выполнением программы данные выводятся из ПЛК в соответствии с

состояниями выходов, которые регенерируются. Например, OUT Y0.0.

1. Прямой выход: во время выполнения инструкций данные выводятся из ПЛК в соответствии с состояниями выходов. Например, OUT DY0.0.

## 2.2.6 Вспомогательные реле (M)

Вспомогательное реле имеет контакты А и В. Его можно использовать несколько раз в программе. Пользователи могут комбинировать контуры управления с помощью вспомогательного реле, но не могут управлять внешней нагрузкой с помощью вспомогательного реле. Вспомогательные реле можно разделить на два типа в соответствии с их атрибутами.

1. Для общего применения: Если при работе ПЛК происходит отключение электропитания, вспомогательное реле для общего применения будет сброшено в состояние ВЫКЛ. Когда питание восстановится, состояние вспомогательного реле для общего применения останется выключенным.
2. Для энергонезависимых областей: Если при работе ПЛК происходит отключение электропитания, состояние вспомогательного энергонезависимого реле будет зафиксировано. Когда питание восстановится, состояние вспомогательного энергонезависимого реле останется тем же.

## 2.2.7 Специальные вспомогательные реле (SM)

Каждое специальное вспомогательное реле имеет свою специфическую функцию. Не используйте специальные вспомогательные реле, функции которых не определены.

Специальные вспомогательные реле и их функции перечислены ниже. Для номеров SM, обозначенных «\*», пользователи могут обратиться за дополнительной информацией о специальных вспомогательных реле / специальных регистрах данных. Также в таблице указатель “R” в таблице означает, что данные можно только считывать, указатель “R/W” означает, что данные можно считывать и записывать. Указатель “–” означает, что состояние реле не изменяется, указатель “#” показывает, что установка реле зависит от состояния ПЛК, но можно считывать их значения.

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM0	Ошибка выполнения операции или значение операнда выходит за пределы допустимого диапазона	○	OFF	OFF	–	N	R	OFF
SM1	Ошибка выполнения операции или значения операнда заблокирована.	○	OFF	OFF	–	N	R	OFF
SM5	Ошибка проверки инструкции	○	OFF	OFF	–	N	R	OFF
SM6	Потеря данных в энергонезависимой области	○	OFF	–	–	N	R/W	OFF
SM7	Недостаточное напряжение питания (24В)	○	OFF	–	–	N	R	OFF
*SM8	Ошибка сторожевого таймера	○	OFF	–	–	N	R	OFF
SM9	Системная ошибка	○	OFF	–	–	N	R	OFF
SM10	Ошибка шины входов/выходов	○	OFF	–	–	N	R	OFF
*SM22	Очистка журнала ошибок	○	OFF	OFF	OFF	N	R/W	OFF
SM23	Очистка загруженного журнала	○	OFF	OFF	OFF	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM24	Очистка журнала состояния ПЛК	○	OFF	OFF	OFF	N	R/W	OFF
SM25	Активация флага онлайн-редактирования	○	OFF	-	-	N	R	OFF
SM26	Активация флага отладки	○	OFF	-	-	N	R	OFF
SM28	Ошибка использования высокоскоростной инструкцией точки одновременно как входа и выхода	○	OFF	OFF	OFF	N	R/W	OFF
SM30	Ошибка в удаленном модуле	○	OFF	-	-	N	R	OFF
SM34	Неверный пароль	○	OFF	-	-	N	R/W	OFF
*SM36	Запись на карту памяти. При этом ПЛК работает в соответствии со значением в SR36.	○	OFF	-	-	N	R/W	OFF
SM76	Передача данных через функциональную плату 1	○	OFF	OFF	-	N	R/W	OFF
SM77	Передача данных через функциональную плату 2	○	OFF	OFF	-	N	R/W	OFF
SM78	Ожидание ответа на функциональной плате 1	○	OFF	OFF	-	N	R	OFF
SM79	Ожидание ответа на функциональной плате 2	○	OFF	OFF	-	N	R	OFF
SM80	Завершение приема на функциональной плате 1	○	OFF	OFF	-	N	R/W	OFF
SM81	Завершение приема на функциональной плате 2	○	OFF	OFF	-	N	R/W	OFF
SM82	Ошибка получения данных на функциональной плате 1 с помощью инструкций MODRW или RS.	○	OFF	OFF	-	N	R	OFF
SM83	Ошибка получения данных на функциональной плате 2 с помощью инструкций MODRW или RS.	○	OFF	OFF	-	N	R	OFF
SM84	Нет полученных данных на функциональной плате 1 через заданное время.	○	OFF	OFF	-	N	R/W	OFF
SM85	Нет полученных данных на функциональной плате 2 через заданное время.	○	OFF	OFF	-	N	R/W	OFF
SM86	Выбор режима работы функциональной платы 1 ON: 8-битовый режим OFF: 16-битовый режим	○	OFF	-	-	N	R/W	OFF
SM87	Выбор режима работы функциональной платы 2 ON: 8-битовый режим OFF: 16-битовый режим	○	OFF	-	-	N	R/W	OFF
SM90	Изменение протокола связи на функциональной плате 1	○	OFF	-	-	N	R/W	OFF
SM91	Изменение протокола связи на функциональной плате 2	○	OFF	-	-	N	R/W	OFF
SM94	Изменение управления светодиодными индикаторами на COM1	○	-	-	-	H	R/W	OFF
SM95	Изменение управления светодиодными индикаторами на COM2	○	-	-	-	H	R/W	OFF
*SM96	Передача данных через COM1.	○	OFF	OFF	-	N	R/W	OFF
*SM97	Передача данных через COM2.	○	OFF	OFF	-	N	R/W	OFF
*SM98	Ожидание приема данных через COM1	○	OFF	OFF	-	N	R	OFF
*SM99	Ожидание приема данных через COM2	○	OFF	OFF	-	N	R	OFF
*SM100	Прием данных через COM1 выполнен.	○	OFF	OFF	-	N	R/W	OFF
*SM101	Прием данных через COM2 выполнен.	○	OFF	OFF	-	N	R/W	OFF
*SM102	Ошибка получения данных на COM1 с помощью инструкций MODRW или RS.	○	OFF	OFF	-	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM103	Ошибка получения данных на COM2 с помощью инструкций MODRW или RS.	○	OFF	OFF	–	N	R/W	OFF
*SM104	Нет полученных данных на COM1 через заданное время.	○	OFF	OFF	–	N	R/W	OFF
*SM105	Нет полученных данных на COM2 через заданное время.	○	OFF	OFF	–	N	R/W	OFF
*SM106	Выбор режима работы COM1 ON: 8-битовый режим OFF: 16-битовый режим	○	OFF	–	–	N	R/W	OFF
*SM107	Выбор режима работы COM2 ON: 8-битовый режим OFF: 16-битовый режим	○	OFF	–	–	N	R/W	OFF
SM166	Запущен VR0 (работа с SR166)	○	OFF	–	–	N	R/W	OFF
SM167	Запущен VR1 (работа с SR167)	○	OFF	–	–	N	R/W	OFF
SM168	Создано подключение для функциональной платы 1	○	–	–	–	N	R	OFF
SM169	Функциональная плата 1 работает	○	OFF	–	–	N	R	OFF
SM170	Создано подключение для функциональной платы 2	○	–	–	–	N	R	OFF
SM171	Функциональная плата 2 работает	○	OFF	–	–	N	R	OFF
*SM204	Очищены все энергозависимые области	○	OFF	OFF	OFF	N	R/W	OFF
*SM205	Очищены все энергонезависимые области	○	OFF	OFF	OFF	N	R/W	OFF
SM206	Подавление всех выходов	○	OFF	–	–	N	R/W	OFF
*SM209	Изменен протокол связи на COM1	○	OFF	OFF	OFF	N	R/W	OFF
*SM210	Изменение режима работы COM1 между ASCII и RTU ON: Режим RTU	○	–	–	–	H	R/W	OFF
*SM211	Изменен протокол связи на COM2	○	OFF	OFF	OFF	N	R/W	OFF
*SM212	Изменение режима работы COM2 между ASCII и RTU ON: Режим RTU	○	–	–	–	H	R/W	OFF
SM215	Рабочее состояние ПЛК	○	OFF	ON	OFF	N	R/W	OFF
SM218	Неисправность часов реального времени	○	–	–	–	N	R	OFF
SM219	Батарея часов реального времени разряжена	○	–	–	–	N	R	OFF
*SM220	Калибровка часов реального времени в пределах ±30 секунд	○	OFF	OFF	–	N	R/W	OFF
SM270	Флаг изменения направления входа для MPG 1 (X0.0/X0.1)	○	OFF	OFF	–	N	R/W	OFF
SM271	Флаг изменения направления входа для MPG 2 (X0.2/X0.3)	○	OFF	OFF	–	N	R/W	OFF
SM272	Флаг изменения направления входа для MPG 3 (X0.4/X0.5)	○	OFF	OFF	–	N	R/W	OFF
SM273	Флаг изменения направления входа для MPG 4 (X0.6/X0.7)	○	OFF	OFF	–	N	R/W	OFF
SM274	Флаг изменения направления входа для MPG 5 (X0.8/X0.9)	○	OFF	OFF	–	N	R/W	OFF
SM275	Флаг изменения направления входа для MPG 6 (X0.10/X0.11)	○	OFF	OFF	–	N	R/W	OFF
SM221	Переключение на летнее время	○	–	–	–	H	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM281	Флаг изменения направления входа для высокоскоростного счетчика 1	○	OFF	OFF	–	N	R/W	OFF
SM282	Флаг изменения направления входа для высокоскоростного счетчика 2	○	OFF	OFF	–	N	R/W	OFF
SM283	Флаг изменения направления входа для высокоскоростного счетчика 3	○	OFF	OFF	–	N	R/W	OFF
SM284	Флаг изменения направления входа для высокоскоростного счетчика 4	○	OFF	OFF	–	N	R/W	OFF
SM285	Флаг изменения направления входа для высокоскоростного счетчика 5	○	OFF	OFF	–	N	R/W	OFF
SM286	Флаг изменения направления входа для высокоскоростного счетчика 6	○	OFF	OFF	–	N	R/W	OFF
SM287	Флаг изменения направления входа для высокоскоростного счетчика 7	○	OFF	OFF	–	N	R/W	OFF
SM288	Флаг изменения направления входа для высокоскоростного счетчика 8	○	OFF	OFF	–	N	R/W	OFF
SM291	Флаг сброса входа для высокоскоростного счетчика 1	○	OFF	OFF	–	N	R/W	OFF
SM292	Флаг сброса входа для высокоскоростного счетчика 2	○	OFF	OFF	–	N	R/W	OFF
SM293	Флаг сброса входа для высокоскоростного счетчика 3	○	OFF	OFF	–	N	R/W	OFF
SM294	Флаг сброса входа для высокоскоростного счетчика 4	○	OFF	OFF	–	N	R/W	OFF
SM295	Флаг сброса входа для высокоскоростного счетчика 5	○	OFF	OFF	–	N	R/W	OFF
SM296	Флаг сброса входа для высокоскоростного счетчика 6	○	OFF	OFF	–	N	R/W	OFF
SM300	Задание режима счета для HC200. (HC200 считает при включенном SM300)	○	OFF	OFF	–	N	R/W	OFF
SM301	Задание режима счета для HC201. (HC201 считает при включенном SM301)	○	OFF	–	–	N	R	OFF
SM302	Задание режима счета для HC202. (HC202 считает при включенном SM302)	○	OFF	–	–	N	R	OFF
SM303	Задание режима счета для HC203. (HC203 считает при включенном SM303)	○	OFF	–	–	N	R	OFF
SM304	Задание режима счета для HC204. (HC204 считает при включенном SM304)	○	OFF	OFF	–	N	R/W	OFF
SM305	Задание режима счета для HC205. (HC205 считает при включенном SM305)	○	OFF	–	–	N	R	OFF
SM306	Задание режима счета для HC206. (HC206 считает при включенном SM306)	○	OFF	–	–	N	R	OFF
SM307	Задание режима счета для HC207. (HC207 считает при включенном SM307)	○	OFF	–	–	N	R	OFF
SM308	Задание режима счета для HC208. (HC208 считает при включенном SM308)	○	OFF	OFF	–	N	R/W	OFF
SM309	Задание режима счета для HC209. (HC209 считает при включенном SM309)	○	OFF	–	–	N	R	OFF
SM310	Задание режима счета для HC210. (HC210 считает при включенном SM310)	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM311	Задание режима счета для HC211. (HC211 считает при включенном SM311)	○	OFF	–	–	N	R	OFF
SM312	Задание режима счета для HC212. (HC212 считает при включенном SM312)	○	OFF	OFF	–	N	R/W	OFF
SM313	Задание режима счета для HC213. (HC213 считает при включенном SM313)	○	OFF	–	–	N	R	OFF
SM314	Задание режима счета для HC214. (HC214 считает при включенном SM314)	○	OFF	–	–	N	R	OFF
SM315	Задание режима счета для HC215. (HC215 считает при включенном SM315)	○	OFF	–	–	N	R	OFF
SM316	Задание режима счета для HC216. (HC216 считает при включенном SM316)	○	OFF	OFF	–	N	R/W	OFF
SM317	Задание режима счета для HC217. (HC217 считает при включенном SM317)	○	OFF	–	–	N	R	OFF
SM318	Задание режима счета для HC218. (HC218 считает при включенном SM318)	○	OFF	–	–	N	R	OFF
SM319	Задание режима счета для HC219. (HC219 считает при включенном SM319)	○	OFF	–	–	N	R	OFF
SM320	Задание режима счета для HC220. (HC220 считает при включенном SM320)	○	OFF	OFF	–	N	R/W	OFF
SM321	Задание режима счета для HC221. (HC221 считает при включенном SM321)	○	OFF	–	–	N	R	OFF
SM322	Задание режима счета для HC222. (HC222 считает при включенном SM322)	○	OFF	–	–	N	R	OFF
SM323	Задание режима счета для HC223. (HC223 считает при включенном SM323)	○	OFF	–	–	N	R	OFF
SM332	Задание режима счета для HC232. (HC232 считает при включенном SM332)	○	OFF	OFF	–	N	R/W	OFF
SM333	Задание режима счета для HC233. (HC233 считает при включенном SM333)	○	OFF	–	–	N	R	OFF
SM334	Задание режима счета для HC234. (HC234 считает при включенном SM334)	○	OFF	–	–	N	R	OFF
SM335	Задание режима счета для HC235. (HC235 считает при включенном SM335)	○	OFF	–	–	N	R	OFF
SM336	Задание режима счета для HC236. (HC236 считает при включенном SM336)	○	OFF	OFF	–	N	R/W	OFF
SM337	Задание режима счета для HC237. (HC237 считает при включенном SM337)	○	OFF	–	–	N	R	OFF
SM338	Задание режима счета для HC238. (HC238 считает при включенном SM338)	○	OFF	–	–	N	R	OFF
SM339	Задание режима счета для HC239. (HC239 считает при включенном SM339)	○	OFF	–	–	N	R	OFF
SM340	Задание режима счета для HC240. (HC240 считает при включенном SM340)	○	OFF	OFF	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM341	Задание режима счета для HC241. (HC241 считает при включенном SM341)	○	OFF	–	–	N	R	OFF
SM342	Задание режима счета для HC242. (HC242 считает при включенном SM342)	○	OFF	OFF	–	N	R/W	OFF
SM343	Задание режима счета для HC243. (HC243 считает при включенном SM343)	○	OFF	–	–	N	R	OFF
SM344	Задание режима счета для HC244. (HC244 считает при включенном SM344)	○	OFF	OFF	–	N	R/W	OFF
SM345	Задание режима счета для HC245. (HC245 считает при включенном SM345)	○	OFF	–	–	N	R	OFF
SM346	Задание режима счета для HC246. (HC246 считает при включенном SM346)	○	OFF	OFF	–	N	R/W	OFF
SM347	Задание режима счета для HC247. (HC247 считает при включенном SM347)	○	OFF	–	–	N	R	OFF
SM348	Задание режима счета для HC248. (HC248 считает при включенном SM348)	○	OFF	OFF	–	N	R/W	OFF
SM349	Задание режима счета для HC249. (HC249 считает при включенном SM349)	○	OFF	–	–	N	R	OFF
SM350	Задание режима счета для HC250. (HC250 считает при включенном SM350)	○	OFF	OFF	–	N	R/W	OFF
SM351	Задание режима счета для HC251. (HC251 считает при включенном SM351)	○	OFF	–	–	N	R	OFF
SM352	Задание режима счета для HC252. (HC252 считает при включенном SM352)	○	OFF	OFF	–	N	R/W	OFF
SM353	Задание режима счета для HC253. (HC253 считает при включенном SM353)	○	OFF	OFF	–	N	R/W	OFF
*SM400	НО контакт	○	OFF	ON	OFF	N	R	OFF
*SM401	НЗ контакт	○	OFF	OFF	ON	N	R	OFF
*SM402	Импульс при переводе ПЛК в работу	○	OFF	ON	OFF	N	R	OFF
*SM403	Отсутствие импульса при переводе ПЛК в Работу	○	OFF	OFF	ON	N	R	OFF
*SM404	Тактовый импульс 10 мс: 5 мс импульс включен, 5 мс выключен	○	OFF	–	–	N	R	OFF
*SM405	Тактовый импульс 100 мс: 50 мс импульс включен, 50 мс выключен	○	OFF	–	–	N	R	OFF
*SM406	Тактовый импульс 200 мс: 100 мс импульс включен, 100 мс выключен	○	OFF	–	–	N	R	OFF
*SM407	Тактовый импульс 10 сек: 500 мс импульс включен, 500 мс выключен	○	OFF	–	–	N	R	OFF
*SM450	Режим видимости карты памяти ВКЛ: Карта памяти невидима. ВЫКЛ: Карта памяти видима.	○	–	–	–	N	R	OFF
*SM452	Режим видимости данных на карте памяти. ВКЛ: Данные видны. ВЫКЛ: Данные не видны.	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM453	Ошибка во время работы карты памяти. ВКЛ: Ошибка произошла.	○	OFF	–	–	N	R	OFF
SM454	Включение/отключение регистратора данных. (ВКЛ: включен, ВЫКЛ: выключен)	○	OFF	–	–	N	R/W	OFF
SM455	Процесс сохранения образцов в регистраторе (ВКЛ: буфер полон или идет процесс)	○	OFF	–	–	N	R	OFF
*SM456	Использование регистратора и карты памяти (ВКЛ: в соответствии со значением в SR902)	○	OFF	–	–	N	R/W	OFF
SM457	Состояние образцов параметров в регистраторе (ВКЛ: образец параметра задан)	○	–	–	–	N	R	OFF
SM460	Вывод значения для Y0.0 оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R	OFF
SM461	Вывод для Y0.0 оси 1 (Y0.0/Y0.1) выполнен.	○	OFF	OFF	–	N	R/W	OFF
SM462	Изменение направления оси 1 (Y0.1)	○	OFF	OFF	–	N	R/W	OFF
*SM463	Остановка выхода Y0.0 оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM464	Включено положительное максимальное значение оси 1 (Y0.0/Y0.1)	○	–	–	–	Y	R/W	OFF
SM465	Сигнал достижения положительного предела оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM466	Включено отрицательное максимальное значение оси 1 (Y0.0/Y0.1)	○	–	–	–	Y	R/W	OFF
SM467	Сигнал достижения отрицательного предела оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
*SM468	Включена S-образная рамповая кривая для оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM469	Включен фиксированный наклон ramпы для оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM470	Завершение вывода с автоматическим сбросом для Y0.0 оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM471	Выполнение прерывания I500 при остановке импульсного выхода для оси 1 (Y0.0/Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM472	Вывод значения для Y0.1	○	OFF	OFF	–	N	R	OFF
SM473	Вывод для Y0.1 выполнен.	○	OFF	OFF	–	N	R/W	OFF
*SM474	Остановка вывода на Y0.1.	○	OFF	OFF	–	N	R/W	OFF
SM475	Автоматический сброс при выполнении на Y0.1	○	OFF	OFF	–	N	R/W	OFF
*SM476	Немедленный сброс вывода на Y0.0 оси 1 (Y0.0/Y0.1) при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
*SM477	Немедленный сброс вывода на Y0.1 при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM478	Изменение целевой позиции при выводе на Y0.0 / ось 1 (Y0.0 / Y0.1)	○	OFF	OFF	–	N	R/W	OFF
SM479	Изменение целевой позиции при выводе на Y0.1	○	OFF	OFF	–	N	R/W	OFF
SM480	Вывод значения для Y0.2 оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R	OFF
SM481	Вывод для Y0.2 оси 2 (Y0.2/Y0.3) выполнен.	○	OFF	OFF	–	N	R/W	OFF



SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM482	Изменение направления оси 2 (Y0.3)	○	OFF	OFF	–	N	R/W	OFF
*SM483	Остановка выхода Y0.2 оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R/W	OFF
SM484	Включено положительное максимальное значение оси 2 (Y0.2/Y0.3)	○	–	–	–	Y	R/W	OFF
SM485	Сигнал достижения положительного предела оси 2 (Y0.2/Y0.2)	○	OFF	OFF	–	N	R/W	OFF
SM486	Включено отрицательное максимальное значение оси 2 (Y0.2/Y0.3)	○	–	–	–	Y	R/W	OFF
SM487	Сигнал достижения отрицательного предела оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R/W	OFF
*SM488	Включена S-образная рамповая кривая для оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R/W	OFF
SM489	Включен фиксированный наклон рампы для оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R/W	OFF
SM490	Завершение вывода с автоматическим сбросом для Y0.2 оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R/W	OFF
SM491	Выполнение прерывания I501 при остановке импульсного выхода для оси 2 (Y0.2/Y0.3)	○	OFF	OFF	–	N	R/W	OFF
SM492	Вывод значения для Y0.3.	○	OFF	OFF	–	N	R	OFF
SM493	Вывод на Y0.3 выполнен.	○	OFF	OFF	–	N	R/W	OFF
*SM494	Остановка вывода на Y0.3.	○	OFF	OFF	–	N	R/W	OFF
SM495	Автоматический сброс при выполнении на Y0.3	○	OFF	OFF	–	N	R/W	OFF
*SM496	Немедленный сброс вывода на Y0.2 оси 2 (Y0.2/Y0.3) при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
*SM497	Немедленный сброс вывода на Y0.3 при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM498	Изменение целевой позиции при выводе на Y0.2 / ось 2 (Y0.2 / Y0.3)	○	OFF	OFF	–	N	R/W	OFF
SM499	Изменение целевой позиции при выводе на Y0.3	○	OFF	OFF	–	N	R/W	OFF
SM500	Вывод значения для Y0.4 оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R	OFF
SM501	Вывод Y0.4 оси 3 (Y0.4/Y0.5) выполнен.	○	OFF	OFF	–	N	R/W	OFF
SM502	Изменение направления оси 3 (Y0.5)	○	OFF	OFF	–	N	R/W	OFF
*SM503	Остановка выхода Y0.4 оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF
SM504	Включено положительное максимальное значение оси 3 (Y0.4/Y0.5)	○	–	–	–	Y	R/W	OFF
SM505	Сигнал достижения положительного предела оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF
SM506	Включено отрицательное максимальное значение оси 3 (Y0.4/Y0.5)	○	–	–	–	Y	R/W	OFF
SM507	Сигнал достижения отрицательного предела оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF
*SM508	Включена S-образная рамповая кривая для оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM509	Включен фиксированный наклон рампы для оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF
SM510	Завершение вывода с автоматическим сбросом для Y0.4 оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF
SM511	Выполнение прерывания I502 при остановке импульсного выхода для оси 3 (Y0.4/Y0.5)	○	OFF	OFF	–	N	R/W	OFF
SM512	Вывод значения для Y0.5	○	OFF	OFF	–	N	R	OFF
SM513	Вывод на Y0.5 выполнен.	○	OFF	OFF	–	N	R/W	OFF
*SM514	Остановка вывода на Y0.5.	○	OFF	OFF	–	N	R/W	OFF
SM515	Автоматический сброс при выполнении на Y0.5	○	OFF	OFF	–	N	R/W	OFF
*SM516	Немедленный сброс вывода на Y0.4 оси 3 (Y0.4/Y0.5) при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
*SM517	Немедленный сброс вывода на Y0.5 при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM518	Изменение целевой позиции при выводе на Y0.4 / ось 3 (Y0.4 / Y0.5)	○	OFF	OFF	–	N	R/W	OFF
SM519	Изменение целевой позиции при выводе на Y0.5	○	OFF	OFF	–	N	R/W	OFF
SM520	Вывод значения для Y0.6 оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R	OFF
SM521	Вывод Y0.6 оси 4 (Y0.6/Y0.7) выполнен.	○	OFF	OFF	–	N	R/W	OFF
SM522	Изменение направления оси 4 (Y0.7)	○	OFF	OFF	–	N	R/W	OFF
*SM523	Остановка выхода Y0.6 оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM524	Включено положительное максимальное значение оси 4 (Y0.6/Y0.7)	○	–	–	–	Y	R/W	OFF
SM525	Сигнал достижения положительного предела оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM526	Включено отрицательное максимальное значение оси 4 (Y0.6/Y0.7)	○	–	–	–	Y	R/W	OFF
SM527	Сигнал достижения отрицательного предела оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
*SM528	Включена S-образная рамповая кривая для оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM529	Включен фиксированный наклон рампы для оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM530	Завершение вывода с автоматическим сбросом для Y0.6 оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM531	Выполнение прерывания I503 при остановке импульсного выхода для оси 4 (Y0.6/Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM532	Вывод значения для Y0.7	○	OFF	OFF	–	N	R	OFF
SM533	Вывод на Y0.7 выполнен.	○	OFF	OFF	–	N	R/W	OFF
*SM534	Остановка вывода на Y0.7.	○	OFF	OFF	–	N	R/W	OFF
SM535	Автоматический сброс при выполнении на Y0.7	○	OFF	OFF	–	N	R/W	OFF
*SM536	Немедленный сброс вывода на Y0.6 оси 4 (Y0.6/Y0.7) при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM537	Немедленный сброс вывода на Y0.7 при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM538	Изменение целевой позиции при выводе на Y0.6 / ось 4 (Y0.6 / Y0.7)	○	OFF	OFF	–	N	R/W	OFF
SM539	Изменение целевой позиции при выводе на Y0.7	○	OFF	OFF	–	N	R/W	OFF
SM540	Вывод значения для Y0.8 оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R	OFF
SM541	Вывод для Y0.8 оси 5 (Y0.8/Y0.9) выполнен.	○	OFF	OFF	–	N	R/W	OFF
SM542	Изменение направления оси 5 (Y0.9)	○	OFF	OFF	–	N	R/W	OFF
*SM543	Остановка выхода Y0.8 оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM544	Включено положительное максимальное значение оси 5 (Y0.8/Y0.9)	○	–	–	–	Y	R/W	OFF
SM545	Сигнал достижения положительного предела оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM546	Включено отрицательное максимальное значение оси 5 (Y0.8/Y0.9)	○	–	–	–	Y	R/W	OFF
SM547	Сигнал достижения отрицательного предела оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
*SM548	Включена S-образная рамповая кривая для оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM549	Включен фиксированный наклон ramпы для оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM550	Завершение вывода с автоматическим сбросом для Y0.8 оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM551	Выполнение прерывания I504 при остановке импульсного выхода для оси 5 (Y0.8/Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM552	Вывод значения для Y0.9	○	OFF	OFF	–	N	R	OFF
SM553	Вывод на Y0.9 выполнен.	○	OFF	OFF	–	N	R/W	OFF
*SM554	Остановка вывода на Y0.9.	○	OFF	OFF	–	N	R/W	OFF
SM555	Автоматический сброс при выполнении на Y0.9	○	OFF	OFF	–	N	R/W	OFF
*SM556	Немедленный сброс вывода на Y0.8 оси 5 (Y0.8/Y0.9) при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
*SM557	Немедленный сброс вывода на Y0.9 при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM558	Изменение целевой позиции при выводе на Y0.8 / ось 5 (Y0.8 / Y0.9)	○	OFF	OFF	–	N	R/W	OFF
SM559	Изменение целевой позиции при выводе на Y0.9	○	OFF	OFF	–	N	R/W	OFF
SM560	Вывод значения для Y0.10 оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R	OFF
SM561	Вывод для Y0.10 оси 6 (Y0.10/Y0.11) выполнен.	○	OFF	OFF	–	N	R/W	OFF
SM562	Изменение направления оси 6 (Y0.11)	○	OFF	OFF	–	N	R/W	OFF
*SM563	Остановка выхода Y0.10 оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM564	Включено положительное максимальное значение оси 6 (Y0.10/Y0.11)	○	–	–	–	Y	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM565	Сигнал достижения положительного предела оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM566	Включено отрицательное максимальное значение оси 6 (Y0.10/Y0.11)	○	–	–	–	Y	R/W	OFF
SM567	Сигнал достижения отрицательного предела оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
*SM568	Включена S-образная рамповая кривая для оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM569	Включен фиксированный наклон рампы для оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM570	Завершение вывода с автоматическим сбросом для Y0.10 оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM571	Выполнение прерывания I505 при остановке импульсного выхода для оси 6 (Y0.10/Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM572	Вывод значения для Y0.11	○	OFF	OFF	–	N	R	OFF
SM573	Вывод на Y0.11 выполнен.	○	OFF	OFF	–	N	R/W	OFF
*SM574	Остановка вывода на Y0.11.	○	OFF	OFF	–	N	R/W	OFF
SM575	Автоматический сброс при выполнении на Y0.11	○	OFF	OFF	–	N	R/W	OFF
*SM576	Немедленный сброс вывода на Y0.10 оси 6 (Y0.10/Y0.11) при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
*SM577	Немедленный сброс вывода на Y0.11 при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM578	Изменение целевой позиции при выводе на Y0.10 / ось 6 (Y0.10 / Y0.11)	○	OFF	OFF	–	N	R/W	OFF
SM579	Изменение целевой позиции при выводе на Y0.11	○	OFF	OFF	–	N	R/W	OFF
SM580	Все выходы немедленно выключаются при остановке или отключении инструкции	○	OFF	OFF	–	N	R/W	OFF
SM600	Флаг нуля	○	OFF	–	–	N	R	OFF
SM601	Флаг заимствования	○	OFF	–	–	N	R	OFF
SM602	Флаг переноса	○	OFF	–	–	N	R	OFF
SM604	Задание режима работы инструкции SORT. ВКЛ: По убыванию ВЫКЛ: По возрастанию	○	OFF	–	–	N	R/W	OFF
SM605	Определение рабочего режима команды SMOV	○	OFF	–	–	N	R/W	OFF
SM606	8-битовый или 16-битовый рабочий режим	○	OFF	–	–	N	R/W	OFF
SM607	Флаг матричного сравнения. ВКЛ: Сравнение одинаковых значений ВЫКЛ: Сравнение разных значений	○	OFF	–	–	N	R/W	OFF
SM608	Окончание матричного сравнения. После сравнение последнего бита включается SM608	○	OFF	–	–	N	R	OFF
SM609	Когда SM609 включен, сравнение идет с бита 0.	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM610	Это флаг поиска битов матрицы. Когда искомые биты совпадают, сравнение немедленно останавливается и включается SM610	○	OFF	–	–	N	R	OFF
SM611	Это флаг ошибки указателя матрицы. Когда значение указателя превышает диапазон сравнения, SM611 включен	○	OFF	–	–	N	R	OFF
SM612	Это флаг увеличения указателя матрицы. Текущее значение указателя увеличивается на единицу.	○	OFF	–	–	N	R/W	OFF
SM613	Это флаг сброса указателя матрицы. Текущее значение указателя сбрасывается до нуля.	○	OFF	–	–	N	R/W	OFF
SM614	Это флаг переноса для матрицы вращения / сдвига / вывода.	○	OFF	–	–	N	R	OFF
SM615	Это флаг заема для сдвига / вывода матрицы.	○	OFF	–	–	N	R/W	OFF
SM616	Это флаг направления для вращения / сдвига матрицы. Биты сдвигаются влево, когда SM616 выключен и биты сдвигаются вправо, когда SM616 включен.	○	OFF	–	–	N	R/W	OFF
SM617	Будут подсчитаны биты со значением 0 или 1.	○	OFF	–	–	N	R/W	OFF
SM618	Включен, когда результат подсчета матрицы равен 0.	○	OFF	–	–	N	R/W	OFF
SM619	Включен, когда выполняется инструкция EI.	○	OFF	OFF	–	N	R	OFF
SM620	Если результаты, полученные из сравнения с использованием команды CMPT #, состоят в том, что все устройства включены, SM620 включен	○	OFF	–	–	N	R	OFF
SM621	Режим счетчика HC0. (HC0 подсчитывает при включенном SM621)	○	OFF	–	–	N	R/W	OFF
SM622	Режим счетчика HC. (HC1 подсчитывает при включенном SM622)	○	OFF	–	–	N	R/W	OFF
SM623	Режим счетчика HC2. (HC2 подсчитывает при включенном SM623)	○	OFF	–	–	N	R/W	OFF
SM624	Режим счетчика HC3. (HC3 подсчитывает при включенном SM624)	○	OFF	–	–	N	R/W	OFF
SM625	Режим счетчика HC4. (HC4 подсчитывает при включенном SM625)	○	OFF	–	–	N	R/W	OFF
SM626	Режим счетчика HC5. (HC5 подсчитывает при включенном SM626)	○	OFF	–	–	N	R/W	OFF
SM627	Режим счетчика HC6. (HC6 подсчитывает при включенном SM627)	○	OFF	–	–	N	R/W	OFF
SM628	Режим счетчика HC7. (HC7 подсчитывает при включенном SM628)	○	OFF	–	–	N	R/W	OFF
SM629	Режим счетчика HC8. (HC8 подсчитывает при включенном SM629)	○	OFF	–	–	N	R/W	OFF
SM630	Режим счетчика HC9. (HC9 подсчитывает при включенном SM630)	○	OFF	–	–	N	R/W	OFF
SM631	Режим счетчика HC10. (HC10 подсчитывает при включенном SM631)	○	OFF	–	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM632	Режим счетчика HC11. (HC11 подсчитывает при включенном SM632)	○	OFF	–	–	N	R/W	OFF
SM633	Режим счетчика HC12. (HC12 подсчитывает при включенном SM633)	○	OFF	–	–	N	R/W	OFF
SM634	Режим счетчика HC13. (HC13 подсчитывает при включенном SM634)	○	OFF	–	–	N	R/W	OFF
SM635	Режим счетчика HC14. (HC14 подсчитывает при включенном SM635)	○	OFF	–	–	N	R/W	OFF
SM636	Режим счетчика HC15. (HC15 подсчитывает при включенном SM636)	○	OFF	–	–	N	R/W	OFF
SM637	Режим счетчика HC16. (HC16 подсчитывает при включенном SM637)	○	OFF	–	–	N	R/W	OFF
SM638	Режим счетчика HC17. (HC17 подсчитывает при включенном SM638)	○	OFF	–	–	N	R/W	OFF
SM639	Режим счетчика HC18. (HC18 подсчитывает при включенном SM639)	○	OFF	–	–	N	R/W	OFF
SM640	Режим счетчика HC19. (HC19 подсчитывает при включенном SM640)	○	OFF	–	–	N	R/W	OFF
SM641	Режим счетчика HC20. (HC20 подсчитывает при включенном SM641)	○	OFF	–	–	N	R/W	OFF
SM642	Режим счетчика HC21. (HC21 подсчитывает при включенном SM642)	○	OFF	–	–	N	R/W	OFF
SM643	Режим счетчика HC22. (HC22 подсчитывает при включенном SM643)	○	OFF	–	–	N	R/W	OFF
SM644	Режим счетчика HC23. (HC23 подсчитывает при включенном SM644)	○	OFF	–	–	N	R/W	OFF
SM645	Режим счетчика HC24. (HC24 подсчитывает при включенном SM645)	○	OFF	–	–	N	R/W	OFF
SM646	Режим счетчика HC25. (HC25 подсчитывает при включенном SM646)	○	OFF	–	–	N	R/W	OFF
SM647	Режим счетчика HC26. (HC26 подсчитывает при включенном SM647)	○	OFF	–	–	N	R/W	OFF
SM648	Режим счетчика HC27. (HC27 подсчитывает при включенном SM648)	○	OFF	–	–	N	R/W	OFF
SM649	Режим счетчика HC28. (HC28 подсчитывает при включенном SM649)	○	OFF	–	–	N	R/W	OFF
SM650	Режим счетчика HC29. (HC29 подсчитывает при включенном SM650)	○	OFF	–	–	N	R/W	OFF
SM651	Режим счетчика HC30. (HC30 подсчитывает при включенном SM651)	○	OFF	–	–	N	R/W	OFF
SM652	Режим счетчика HC31. (HC31 подсчитывает при включенном SM652)	○	OFF	–	–	N	R/W	OFF
SM653	Режим счетчика HC32. (HC32 подсчитывает при включенном SM653)	○	OFF	–	–	N	R/W	OFF
SM654	Режим счетчика HC33. (HC33 подсчитывает при включенном SM653)	○	OFF	–	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM655	Режим счетчика HC34. (HC34 подсчитывает при включенном SM655)	○	OFF	–	–	N	R/W	OFF
SM656	Режим счетчика HC35. (HC35 подсчитывает при включенном SM656)	○	OFF	–	–	N	R/W	OFF
SM657	Режим счетчика HC36. (HC36 подсчитывает при включенном SM657)	○	OFF	–	–	N	R/W	OFF
SM658	Режим счетчика HC37. (HC37 подсчитывает при включенном SM658)	○	OFF	–	–	N	R/W	OFF
SM659	Режим счетчика HC38. (HC38 подсчитывает при включенном SM659)	○	OFF	–	–	N	R/W	OFF
SM660	Режим счетчика HC39. (HC39 подсчитывает при включенном SM660)	○	OFF	–	–	N	R/W	OFF
SM661	Режим счетчика HC40. (HC40 подсчитывает при включенном SM661)	○	OFF	–	–	N	R/W	OFF
SM662	Режим счетчика HC41. (HC41 подсчитывает при включенном SM662)	○	OFF	–	–	N	R/W	OFF
SM663	Режим счетчика HC42. (HC42 подсчитывает при включенном SM663)	○	OFF	–	–	N	R/W	OFF
SM664	Режим счетчика HC43. (HC43 подсчитывает при включенном SM664)	○	OFF	–	–	N	R/W	OFF
SM665	Режим счетчика HC44. (HC44 подсчитывает при включенном SM665)	○	OFF	–	–	N	R/W	OFF
SM666	Режим счетчика HC45. (HC45 подсчитывает при включенном SM666)	○	OFF	–	–	N	R/W	OFF
SM667	Режим счетчика HC46. (HC46 подсчитывает при включенном SM667)	○	OFF	–	–	N	R/W	OFF
SM668	Режим счетчика HC47. (HC47 подсчитывает при включенном SM668)	○	OFF	–	–	N	R/W	OFF
SM669	Режим счетчика HC48. (HC48 подсчитывает при включенном SM669)	○	OFF	–	–	N	R/W	OFF
SM670	Режим счетчика HC49. (HC49 подсчитывает при включенном SM670)	○	OFF	–	–	N	R/W	OFF
SM671	Режим счетчика HC50. (HC50 подсчитывает при включенном SM671)	○	OFF	–	–	N	R/W	OFF
SM672	Режим счетчика HC51. (HC51 подсчитывает при включенном SM672)	○	OFF	–	–	N	R/W	OFF
SM673	Режим счетчика HC52. (HC52 подсчитывает при включенном SM673)	○	OFF	–	–	N	R/W	OFF
SM674	Режим счетчика HC53. (HC53 подсчитывает при включенном SM674)	○	OFF	–	–	N	R/W	OFF
SM675	Режим счетчика HC54. (HC54 подсчитывает при включенном SM675)	○	OFF	–	–	N	R/W	OFF
SM676	Режим счетчика HC55. (HC55 подсчитывает при включенном SM676)	○	OFF	–	–	N	R/W	OFF
SM677	Режим счетчика HC56. (HC56 подсчитывает при включенном SM677)	○	OFF	–	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM678	Режим счетчика HC57. (HC57 подсчитывает при включенном SM678)	○	OFF	–	–	N	R/W	OFF
SM679	Режим счетчика HC58. (HC58 подсчитывает при включенном SM679)	○	OFF	–	–	N	R/W	OFF
SM680	Режим счетчика HC59. (HC59 подсчитывает при включенном SM680)	○	OFF	–	–	N	R/W	OFF
SM681	Режим счетчика HC60. (HC60 подсчитывает при включенном SM681)	○	OFF	–	–	N	R/W	OFF
SM682	Режим счетчика HC61. (HC61 подсчитывает при включенном SM682)	○	OFF	–	–	N	R/W	OFF
SM683	Режим счетчика HC62. (HC62 подсчитывает при включенном SM683)	○	OFF	–	–	N	R/W	OFF
SM684	Режим счетчика HC63. (HC63 подсчитывает при включенном SM684)	○	OFF	–	–	N	R/W	OFF
SM685	Инструкция DSCLP использует операцию с плавающей точкой.	○	OFF	–	–	N	R/W	OFF
SM686	Режим инструкции RAMP	○	OFF	–	–	N	R/W	OFF
SM687	Выполнение инструкции RAMP завершено.	○	OFF	–	–	N	R/W	OFF
SM688	Выполнение инструкции INCD завершено.	○	OFF	–	–	N	R/W	OFF
SM690	Режим управления строкой	○	OFF	–	–	N	R/W	OFF
SM691	Входной режим инструкции НКУ 16-битовый. При включенном SM691 вход является шестнадцатеричным, при выключенном, клавиши A~F являются функциональными.	○	OFF	–	–	N	R/W	OFF
SM692	После выполнения инструкции НКУ, SM692 работает в сканирующем цикле.	○	OFF	–	–	N	R/W	OFF
SM693	После выполнения инструкции SEGL, SM693 работает в сканирующем цикле	○	OFF	–	–	N	R/W	OFF
SM694	После выполнения инструкции DSW, SM694 работает в сканирующем цикле	○	OFF	–	–	N	R/W	OFF
SM695	Флаг определения градусы/радианы ВКЛ: Градусы	○	OFF	–	–	N	R/W	OFF
SM749	Ошибка инициализации обмена данными через COM1	○	OFF	–	–	N	R/W	OFF
*SM750	Обмен данными через COM1 был запущен с помощью ISPSOft.	○	OFF	–	–	H	R/W	OFF
*SM752	Запущено подключение 1 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM753	Запущено подключение 2 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM754	Запущено подключение 3 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM755	Запущено подключение 4 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM756	Запущено подключение 5 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF



SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM757	Запущено подключение 6 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM758	Запущено подключение 7 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM759	Запущено подключение 8 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM760	Запущено подключение 9 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM761	Запущено подключение 10 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM762	Запущено подключение 11 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM763	Запущено подключение 12 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM764	Запущено подключение 13 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM765	Запущено подключение 14 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM766	Запущено подключение 15 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM767	Запущено подключение 16 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM768	Запущено подключение 17 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM769	Запущено подключение 18 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM770	Запущено подключение 19 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM771	Запущено подключение 20 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM772	Запущено подключение 21 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM773	Запущено подключение 22 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM774	Запущено подключение 23 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM775	Запущено подключение 24 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM776	Запущено подключение 25 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM777	Запущено подключение 26 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF
*SM778	Запущено подключение 27 для обмена данными через COM1	○	OFF	–	–	H	RW	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM779	Запущено подключение 28 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM780	Запущено подключение 29 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM781	Запущено подключение 30 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM782	Запущено подключение 31 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM783	Запущено подключение 32 для обмена данными через COM1	○	OFF	–	–	H	R/W	OFF
*SM784	Получены данные через COM1 по подключению 1	○	OFF	–	–	N	R	OFF
*SM785	Получены данные через COM1 по подключению 2	○	OFF	–	–	N	R	OFF
*SM786	Получены данные через COM1 по подключению 3	○	OFF	–	–	N	R	OFF
*SM787	Получены данные через COM1 по подключению 4	○	OFF	–	–	N	R	OFF
*SM788	Получены данные через COM1 по подключению 5	○	OFF	–	–	N	R	OFF
*SM789	Получены данные через COM1 по подключению 6	○	OFF	–	–	N	R	OFF
*SM790	Получены данные через COM1 по подключению 7	○	OFF	–	–	N	R	OFF
*SM791	Получены данные через COM1 по подключению 8	○	OFF	–	–	N	R	OFF
*SM792	Получены данные через COM1 по подключению 9	○	OFF	–	–	N	R	OFF
*SM793	Получены данные через COM1 по подключению 10	○	OFF	–	–	N	R	OFF
*SM794	Получены данные через COM1 по подключению 11	○	OFF	–	–	N	R	OFF
*SM795	Получены данные через COM1 по подключению 12	○	OFF	–	–	N	R	OFF
*SM796	Получены данные через COM1 по подключению 13	○	OFF	–	–	N	R	OFF
*SM797	Получены данные через COM1 по подключению 14	○	OFF	–	–	N	R	OFF
*SM798	Получены данные через COM1 по подключению 15	○	OFF	–	–	N	R	OFF
*SM799	Получены данные через COM1 по подключению 16	○	OFF	–	–	N	R	OFF
*SM800	Получены данные через COM1 по подключению 17	○	OFF	–	–	N	R	OFF
*SM801	Получены данные через COM1 по подключению 18	○	OFF	–	–	N	R	OFF
*SM802	Получены данные через COM1 по подключению 19	○	OFF	–	–	N	R	OFF
*SM803	Получены данные через COM1 по подключению 20	○	OFF	–	–	N	R	OFF
*SM804	Получены данные через COM1 по подключению 21	○	OFF	–	–	N	R	OFF
*SM805	Получены данные через COM1 по подключению 22	○	OFF	–	–	N	R	OFF
*SM806	Получены данные через COM1 по подключению 23	○	OFF	–	–	N	R	OFF
*SM807	Получены данные через COM1 по подключению 24	○	OFF	–	–	N	R	OFF
*SM808	Получены данные через COM1 по подключению 25	○	OFF	–	–	N	R	OFF
*SM809	Получены данные через COM1 по подключению 26	○	OFF	–	–	N	R	OFF
*SM810	Получены данные через COM1 по подключению 27	○	OFF	–	–	N	R	OFF
*SM811	Получены данные через COM1 по подключению 28	○	OFF	–	–	N	R	OFF
*SM812	Получены данные через COM1 по подключению 29	○	OFF	–	–	N	R	OFF
*SM813	Получены данные через COM1 по подключению 30	○	OFF	–	–	N	R	OFF
*SM814	Получены данные через COM1 по подключению 31	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM815	Получены данные через COM1 по подключению 32	○	OFF	–	–	N	R	OFF
*SM816	Произошла ошибка в подключении 1 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM817	Произошла ошибка в подключении 2 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM818	Произошла ошибка в подключении 3 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM819	Произошла ошибка в подключении 4 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM820	Произошла ошибка в подключении 5 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM821	Произошла ошибка в подключении 6 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM822	Произошла ошибка в подключении 7 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM823	Произошла ошибка в подключении 8 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM824	Произошла ошибка в подключении 9 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM825	Произошла ошибка в подключении 10 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM826	Произошла ошибка в подключении 11 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM827	Произошла ошибка в подключении 12 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM828	Произошла ошибка в подключении 13 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM829	Произошла ошибка в подключении 14 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM830	Произошла ошибка в подключении 15 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM831	Произошла ошибка в подключении 16 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM832	Произошла ошибка в подключении 17 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM833	Произошла ошибка в подключении 18 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM834	Произошла ошибка в подключении 19 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM835	Произошла ошибка в подключении 20 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM836	Произошла ошибка в подключении 21 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM837	Произошла ошибка в подключении 22 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM838	Произошла ошибка в подключении 23 для обмена данными через COM1	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM839	Произошла ошибка в подключении 24 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM840	Произошла ошибка в подключении 25 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM841	Произошла ошибка в подключении 26 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM842	Произошла ошибка в подключении 27 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM843	Произошла ошибка в подключении 28 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM844	Произошла ошибка в подключении 29 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM845	Произошла ошибка в подключении 30 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM846	Произошла ошибка в подключении 31 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
*SM847	Произошла ошибка в подключении 32 для обмена данными через COM1	○	OFF	–	–	N	R	OFF
SM861	Ошибка инициализации обмена данными через COM2	○	OFF	–	–	N	R/W	OFF
*SM862	Обмен данными через COM2 был запущен с помощью ISPSOft.	○	OFF	–	–	H	R/W	OFF
*SM864	Запущено подключение 1 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM865	Запущено подключение 2 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM866	Запущено подключение 3 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM867	Запущено подключение 4 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM868	Запущено подключение 5 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM869	Запущено подключение 6 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM870	Запущено подключение 7 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM871	Запущено подключение 8 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM872	Запущено подключение 9 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM873	Запущено подключение 10 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM874	Запущено подключение 11 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF
*SM875	Запущено подключение 12 для обмена данными через COM2	○	OFF	–	–	H	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM876	Запущено подключение 13 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM877	Запущено подключение 14 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM878	Запущено подключение 15 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM879	Запущено подключение 16 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM880	Запущено подключение 17 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM881	Запущено подключение 18 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM882	Запущено подключение 19 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM883	Запущено подключение 20 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM884	Запущено подключение 21 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM885	Запущено подключение 22 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM886	Запущено подключение 23 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM887	Запущено подключение 24 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM888	Запущено подключение 25 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM889	Запущено подключение 26 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM890	Запущено подключение 27 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM891	Запущено подключение 28 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM892	Запущено подключение 29 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM893	Запущено подключение 30 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM894	Запущено подключение 31 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM895	Запущено подключение 32 для обмена данными через COM2	○	OFF	–	–	H	RW	OFF
*SM896	Получены данные через COM2 по подключению 1	○	OFF	–	–	N	R	OFF
*SM897	Получены данные через COM2 по подключению 2	○	OFF	–	–	N	R	OFF
*SM898	Получены данные через COM2 по подключению 3	○	OFF	–	–	N	R	OFF
*SM899	Получены данные через COM2 по подключению 4	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM900	Получены данные через COM2 по подключению 5	○	OFF	–	–	N	R	OFF
*SM901	Получены данные через COM2 по подключению 6	○	OFF	–	–	N	R	OFF
*SM902	Получены данные через COM2 по подключению 7	○	OFF	–	–	N	R	OFF
*SM903	Получены данные через COM2 по подключению 8	○	OFF	–	–	N	R	OFF
*SM904	Получены данные через COM2 по подключению 9	○	OFF	–	–	N	R	OFF
*SM905	Получены данные через COM2 по подключению 10	○	OFF	–	–	N	R	OFF
*SM906	Получены данные через COM2 по подключению 11	○	OFF	–	–	N	R	OFF
*SM907	Получены данные через COM2 по подключению 12	○	OFF	–	–	N	R	OFF
*SM908	Получены данные через COM2 по подключению 13	○	OFF	–	–	N	R	OFF
*SM909	Получены данные через COM2 по подключению 14	○	OFF	–	–	N	R	OFF
*SM910	Получены данные через COM2 по подключению 15	○	OFF	–	–	N	R	OFF
*SM911	Получены данные через COM2 по подключению 16	○	OFF	–	–	N	R	OFF
*SM912	Получены данные через COM2 по подключению 17	○	OFF	–	–	N	R	OFF
*SM913	Получены данные через COM2 по подключению 18	○	OFF	–	–	N	R	OFF
*SM914	Получены данные через COM2 по подключению 19	○	OFF	–	–	N	R	OFF
*SM915	Получены данные через COM2 по подключению 20	○	OFF	–	–	N	R	OFF
*SM916	Получены данные через COM2 по подключению 21	○	OFF	–	–	N	R	OFF
*SM917	Получены данные через COM2 по подключению 22	○	OFF	–	–	N	R	OFF
*SM918	Получены данные через COM2 по подключению 23	○	OFF	–	–	N	R	OFF
*SM919	Получены данные через COM2 по подключению 24	○	OFF	–	–	N	R	OFF
*SM920	Получены данные через COM2 по подключению 25	○	OFF	–	–	N	R	OFF
*SM921	Получены данные через COM2 по подключению 26	○	OFF	–	–	N	R	OFF
*SM922	Получены данные через COM2 по подключению 27	○	OFF	–	–	N	R	OFF
*SM923	Получены данные через COM2 по подключению 28	○	OFF	–	–	N	R	OFF
*SM924	Получены данные через COM2 по подключению 29	○	OFF	–	–	N	R	OFF
*SM925	Получены данные через COM2 по подключению 30	○	OFF	–	–	N	R	OFF
*SM926	Получены данные через COM2 по подключению 31	○	OFF	–	–	N	R	OFF
*SM927	Получены данные через COM2 по подключению 32	○	OFF	–	–	N	R	OFF
*SM928	Произошла ошибка в подключении 1 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM929	Произошла ошибка в подключении 2 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM930	Произошла ошибка в подключении 3 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM931	Произошла ошибка в подключении 4 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM932	Произошла ошибка в подключении 5 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM933	Произошла ошибка в подключении 6 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM934	Произошла ошибка в подключении 7 для обмена данными через COM2	○	OFF	–	–	N	R	OFF

2

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM935	Произошла ошибка в подключении 8 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM936	Произошла ошибка в подключении 9 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM937	Произошла ошибка в подключении 10 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM938	Произошла ошибка в подключении 11 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM939	Произошла ошибка в подключении 12 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM940	Произошла ошибка в подключении 13 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM941	Произошла ошибка в подключении 14 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM942	Произошла ошибка в подключении 15 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM943	Произошла ошибка в подключении 16 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM944	Произошла ошибка в подключении 17 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM945	Произошла ошибка в подключении 18 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM946	Произошла ошибка в подключении 19 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM947	Произошла ошибка в подключении 20 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM948	Произошла ошибка в подключении 21 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM949	Произошла ошибка в подключении 22 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM950	Произошла ошибка в подключении 23 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM951	Произошла ошибка в подключении 24 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM952	Произошла ошибка в подключении 25 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM953	Произошла ошибка в подключении 26 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM954	Произошла ошибка в подключении 27 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM955	Произошла ошибка в подключении 28 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM956	Произошла ошибка в подключении 29 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM957	Произошла ошибка в подключении 30 для обмена данными через COM2	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM958	Произошла ошибка в подключении 31 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
*SM959	Произошла ошибка в подключении 32 для обмена данными через COM2	○	OFF	–	–	N	R	OFF
SM1000	Флаг настройки Ethernet. Когда SM1000 включен, данные из SR1000~SR1006 записываются во флеш-память.	○	OFF	–	–	N	R/W	OFF
SM1001	Состояние подключения Ethernet	○	OFF	–	–	N	R	OFF
*SM1090	TCP соединение занято	○	OFF	–	–	N	R	OFF
*SM1091	UDP соединение занято	○	OFF	–	–	N	R	OFF
SM1100	Сетевой кабель не подключен	○	OFF	–	–	N	R	OFF
*SM1106	Базовое управление — ошибка подключение Ethernet	○	OFF	–	–	N	R	OFF
*SM1107	Базовое управление – ошибка настройки Ethernet	○	OFF	–	–	N	R	OFF
*SM1109	Базовое управление сокетом TCP/UDP – локальный порт уже используется	○	OFF	–	–	N	R	OFF
SM1111	Флаг обмена данными EtherNet/IP	○	OFF	–	–	N	R	OFF
*SM1113	Ошибка сообщений электронной почты	○	OFF	–	–	N	R	OFF
*SM1116	Переключатель триггера 1 в электронной почте.	○	OFF	–	–	N	R	OFF
*SM1117	Триггер 1 электронной почты	○	OFF	–	–	N	R	OFF
*SM1119	Когда срабатывает триггер 1 и сообщение успешно отправлено, включается SM1119	○	OFF	–	–	N	R	OFF
*SM1120	Когда запускается триггер 1, но сообщение не может быть отправлено из-за ошибки содержимого электронной почты, включается SM1120	○	OFF	–	–	N	R	OFF
*SM1122	Когда запускается триггер 1 и происходит задержка ответа SMTP-сервера, включается SM1122	○	OFF	–	–	N	R	OFF
*SM1123	Когда запускается триггер 1 и возникает ошибка ответа сервера SMTP, включается SM1123	○	OFF	–	–	N	R	OFF
*SM1124	Когда запускается триггер 1 и размер вложения превышает допустимый диапазон, включается SM1124	○	OFF	–	–	N	R	OFF
*SM1125	Когда триггер 1 запускается и вложение не найдено, включается SM1125	○	OFF	–	–	N	R	OFF
*SM1126	Переключатель триггера 2 в электронной почте.	○	OFF	–	–	N	R	OFF
*SM1127	Триггер 2 электронной почты	○	OFF	–	–	N	R	OFF
*SM1129	Когда срабатывает триггер 2 и сообщение успешно отправлено, включается SM1129	○	OFF	–	–	N	R	OFF
*SM1130	Когда запускается триггер 2, но сообщение не может быть отправлено из-за ошибки содержимого электронной почты, включается SM1130	○	OFF	–	–	N	R	OFF
*SM1132	Когда запускается триггер 2 и происходит задержка ответа SMTP-сервера, включается SM1132	○	OFF	–	–	N	R	OFF
*SM1133	Когда запускается триггер 2 и возникает ошибка ответа сервера SMTP, включается SM1133	○	OFF	–	–	N	R	OFF
*SM1134	Когда запускается триггер 2 и размер вложения превышает допустимый диапазон, включается SM1134	○	OFF	–	–	N	R	OFF
*SM1135	Когда триггер 2 запускается и вложение не найдено, включается SM1135	○	OFF	–	–	N	R	OFF
*SM1136	Переключатель триггера 3 в электронной почте.	○	OFF	–	–	N	R	OFF
*SM1137	Триггер 3 электронной почты	○	OFF	–	–	N	R	OFF



SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM1139	Когда срабатывает триггер 3 и сообщение успешно отправлено, включается SM1139	○	OFF	–	–	N	R	OFF
*SM1140	Когда запускается триггер 3, но сообщение не может быть отправлено из-за ошибки содержимого электронной почты, включается SM1140	○	OFF	–	–	N	R	OFF
*SM1142	Когда запускается триггер 3 и происходит задержка ответа SMTP-сервера, включается SM1142	○	OFF	–	–	N	R	OFF
*SM1143	Когда запускается триггер 3 и возникает ошибка ответа сервера SMTP, включается SM1143	○	OFF	–	–	N	R	OFF
*SM1144	Когда запускается триггер 3 и размер вложения превышает допустимый диапазон, включается SM1144	○	OFF	–	–	N	R	OFF
*SM1145	Когда триггер 3 запускается и вложение не найдено, включается SM1145	○	OFF	–	–	N	R	OFF
*SM1146	Переключатель триггера 4 в электронной почте.	○	OFF	–	–	N	R	OFF
*SM1147	Триггер 4 электронной почты	○	OFF	–	–	N	R	OFF
*SM1149	Когда срабатывает триггер 4 и сообщение успешно отправлено, включается SM1149	○	OFF	–	–	N	R	OFF
*SM1150	Когда запускается триггер 4, но сообщение не может быть отправлено из-за ошибки содержимого электронной почты, включается SM1150	○	OFF	–	–	N	R	OFF
*SM1152	Когда запускается триггер 4 и происходит задержка ответа SMTP-сервера, включается SM1152	○	OFF	–	–	N	R	OFF
*SM1153	Когда запускается триггер 4 и возникает ошибка ответа сервера SMTP, включается SM1153	○	OFF	–	–	N	R	OFF
*SM1154	Когда запускается триггер 4 и размер вложения превышает допустимый диапазон, включается SM1154	○	OFF	–	–	N	R	OFF
*SM1155	Когда триггер 4 запускается и вложение не найдено, включается SM1155	○	OFF	–	–	N	R	OFF
*SM1166	Ошибка инициализации обмена данными через порт Ethernet	○	–	–	–	N	R	OFF
*SM1167	Обмен данными через порт Ethernet запущен с помощью ПО IPSOft.	○	OFF	–	–	H	R/W	OFF
*SM1168	Запущено подключение 1 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1169	Запущено подключение 2 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1170	Запущено подключение 3 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1171	Запущено подключение 4 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1172	Запущено подключение 5 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1173	Запущено подключение 6 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1174	Запущено подключение 7 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1175	Запущено подключение 8 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1176	Запущено подключение 9 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM1177	Запущено подключение 10 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1178	Запущено подключение 11 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1179	Запущено подключение 12 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1180	Запущено подключение 13 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1181	Запущено подключение 14 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1182	Запущено подключение 15 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1183	Запущено подключение 16 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1184	Запущено подключение 17 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1185	Запущено подключение 18 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1186	Запущено подключение 19 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1187	Запущено подключение 20 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1188	Запущено подключение 21 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1189	Запущено подключение 22 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1190	Запущено подключение 23 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1191	Запущено подключение 24 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1192	Запущено подключение 25 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1193	Запущено подключение 26 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1194	Запущено подключение 27 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1195	Запущено подключение 28 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1196	Запущено подключение 29 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1197	Запущено подключение 30 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF
*SM1198	Запущено подключение 31 для обмена данными через Ethernet	○	OFF	–	–	H	R/W	OFF

2

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM1199	Запущено подключение 32 для обмена данными через Ethernet	○	OFF	–	–	H	RW	OFF
*SM1200	Получены данные через Ethernet по подключению 1	○	OFF	–	–	N	R	OFF
*SM1201	Получены данные через Ethernet по подключению 2	○	OFF	–	–	N	R	OFF
*SM1202	Получены данные через Ethernet по подключению 3	○	OFF	–	–	N	R	OFF
*SM1203	Получены данные через Ethernet по подключению 4	○	OFF	–	–	N	R	OFF
*SM1204	Получены данные через Ethernet по подключению 5	○	OFF	–	–	N	R	OFF
*SM1205	Получены данные через Ethernet по подключению 6	○	OFF	–	–	N	R	OFF
*SM1206	Получены данные через Ethernet по подключению 7	○	OFF	–	–	N	R	OFF
*SM1207	Получены данные через Ethernet по подключению 8	○	OFF	–	–	N	R	OFF
*SM1208	Получены данные через Ethernet по подключению 9	○	OFF	–	–	N	R	OFF
*SM1209	Получены данные через Ethernet по подключению 10	○	OFF	–	–	N	R	OFF
*SM1210	Получены данные через Ethernet по подключению 11	○	OFF	–	–	N	R	OFF
*SM1211	Получены данные через Ethernet по подключению 12	○	OFF	–	–	N	R	OFF
*SM1212	Получены данные через Ethernet по подключению 13	○	OFF	–	–	N	R	OFF
*SM1213	Получены данные через Ethernet по подключению 14	○	OFF	–	–	N	R	OFF
*SM1214	Получены данные через Ethernet по подключению 15	○	OFF	–	–	N	R	OFF
*SM1215	Получены данные через Ethernet по подключению 16	○	OFF	–	–	N	R	OFF
*SM1216	Получены данные через Ethernet по подключению 17	○	OFF	–	–	N	R	OFF
*SM1217	Получены данные через Ethernet по подключению 18	○	OFF	–	–	N	R	OFF
*SM1218	Получены данные через Ethernet по подключению 19	○	OFF	–	–	N	R	OFF
*SM1219	Получены данные через Ethernet по подключению 20	○	OFF	–	–	N	R	OFF
*SM1220	Получены данные через Ethernet по подключению 21	○	OFF	–	–	N	R	OFF
*SM1221	Получены данные через Ethernet по подключению 22	○	OFF	–	–	N	R	OFF
*SM1222	Получены данные через Ethernet по подключению 23	○	OFF	–	–	N	R	OFF
*SM1223	Получены данные через Ethernet по подключению 24	○	OFF	–	–	N	R	OFF
*SM1224	Получены данные через Ethernet по подключению 25	○	OFF	–	–	N	R	OFF
*SM1225	Получены данные через Ethernet по подключению 26	○	OFF	–	–	N	R	OFF
*SM1226	Получены данные через Ethernet по подключению 27	○	OFF	–	–	N	R	OFF
*SM1227	Получены данные через Ethernet по подключению 28	○	OFF	–	–	N	R	OFF
*SM1228	Получены данные через Ethernet по подключению 29	○	OFF	–	–	N	R	OFF
*SM1229	Получены данные через Ethernet по подключению 30	○	OFF	–	–	N	R	OFF
*SM1230	Получены данные через Ethernet по подключению 31	○	OFF	–	–	N	R	OFF
*SM1231	Получены данные через Ethernet по подключению 32	○	OFF	–	–	N	R	OFF
*SM1232	Произошла ошибка в подключении 1 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1233	Произошла ошибка в подключении 2 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1234	Произошла ошибка в подключении 3 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM1235	Произошла ошибка в подключении 4 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1236	Произошла ошибка в подключении 5 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1237	Произошла ошибка в подключении 6 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1238	Произошла ошибка в подключении 7 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1239	Произошла ошибка в подключении 8 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1240	Произошла ошибка в подключении 9 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1241	Произошла ошибка в подключении 10 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1242	Произошла ошибка в подключении 11 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1243	Произошла ошибка в подключении 12 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1244	Произошла ошибка в подключении 13 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1245	Произошла ошибка в подключении 14 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1246	Произошла ошибка в подключении 15 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1247	Произошла ошибка в подключении 16 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1248	Произошла ошибка в подключении 17 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1249	Произошла ошибка в подключении 18 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1250	Произошла ошибка в подключении 19 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1251	Произошла ошибка в подключении 20 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1252	Произошла ошибка в подключении 21 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1253	Произошла ошибка в подключении 22 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1254	Произошла ошибка в подключении 23 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1255	Произошла ошибка в подключении 24 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1256	Произошла ошибка в подключении 25 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SM1257	Произошла ошибка в подключении 26 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1258	Произошла ошибка в подключении 27 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1259	Произошла ошибка в подключении 28 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1260	Произошла ошибка в подключении 29 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1261	Произошла ошибка в подключении 30 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1262	Произошла ошибка в подключении 31 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
*SM1263	Произошла ошибка в подключении 32 для обмена данными через Ethernet	○	OFF	–	–	N	R	OFF
SM1269	Ошибка конфигурации сокета	○	OFF	–	–	N	R/W	OFF
SM1270	TCP сокет 1—Соединение установлено.	○	OFF	–	–	N	R	OFF
SM1271	TCP сокет 1—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1272	TCP сокет 1—Данные переданы.	○	OFF	–	–	N	R	OFF
SM1273	TCP сокет 1—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1274	TCP сокет 1—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1275	TCP сокет 1—Данные отправляются.	○	OFF	–	–	N	R	OFF
SM1277	TCP сокет 1—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1278	TCP сокет 2—Соединение установлено.	○	OFF	–	–	N	R	OFF
SM1279	TCP сокет 2—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1280	TCP сокет 2—Данные переданы.	○	OFF	–	–	N	R	OFF
SM1281	TCP сокет 2—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1282	TCP сокет 2—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1283	TCP сокет 2—Данные отправляются.	○	OFF	–	–	N	R	OFF
SM1285	TCP сокет 2—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1286	TCP сокет 3—Соединение установлено.	○	OFF	–	–	N	R	OFF
SM1287	TCP сокет 3—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1288	TCP сокет 3—Данные переданы.	○	OFF	–	–	N	R	OFF
SM1289	TCP сокет 3—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1290	TCP сокет 3—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1291	TCP сокет 3—Данные отправляются.	○	OFF	–	–	N	R	OFF
SM1293	TCP сокет 3—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1294	TCP сокет 4—Соединение установлено.	○	OFF	–	–	N	R	OFF
SM1295	TCP сокет 4—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1296	TCP сокет 4—Данные переданы.	○	OFF	–	–	N	R	OFF
SM1297	TCP сокет 4—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1298	TCP сокет 4—Соединение закрыто.	○	ON	–	–	Y	R	ON

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1299	TCP сокет 4—Данные отправляются.	○	OFF	–	–	N	R	OFF
SM1301	TCP сокет 4—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1334	UDP сокет 1—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1335	UDP сокет 1—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1336	UDP сокет 1—Данные отправлены.	○	OFF	–	–	N	R	OFF
SM1337	UDP сокет 1—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1338	UDP сокет 1—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1339	UDP сокет 2—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1340	UDP сокет 2—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1341	UDP сокет 2—Данные отправлены.	○	OFF	–	–	N	R	OFF
SM1342	UDP сокет 2—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1343	UDP сокет 2—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1344	UDP сокет 3—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1345	UDP сокет 3—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1346	UDP сокет 3—Данные отправлены.	○	OFF	–	–	N	R	OFF
SM1347	UDP сокет 3—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1348	UDP сокет 3—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1349	UDP сокет 4—Соединение запущено.	○	OFF	–	–	N	R	OFF
SM1350	UDP сокет 4—Данные приняты.	○	OFF	–	–	N	R	OFF
SM1351	UDP сокет 4—Данные отправлены.	○	OFF	–	–	N	R	OFF
SM1352	UDP сокет 4—Соединение закрыто.	○	ON	–	–	Y	R	ON
SM1353	UDP сокет 4—Флаг ошибки	○	OFF	–	–	N	R	OFF
SM1375	Запущен обмен данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1376	Запущено подключение 1 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1377	Запущено подключение 2 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1378	Запущено подключение 3 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1379	Запущено подключение 4 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1380	Запущено подключение 5 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1381	Запущено подключение 6 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1382	Запущено подключение 7 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1383	Запущено подключение 8 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1384	Запущено подключение 9 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1385	Запущено подключение 10 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1386	Запущено подключение 11 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1387	Запущено подключение 12 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1388	Запущено подключение 13 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1389	Запущено подключение 14 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1390	Запущено подключение 15 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1391	Запущено подключение 16 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1392	Запущено подключение 17 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1393	Запущено подключение 18 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1394	Запущено подключение 19 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1395	Запущено подключение 20 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1396	Запущено подключение 21 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1397	Запущено подключение 22 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1398	Запущено подключение 23 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1399	Запущено подключение 24 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1400	Запущено подключение 25 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1401	Запущено подключение 26 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1402	Запущено подключение 27 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1403	Запущено подключение 28 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1404	Запущено подключение 29 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1405	Запущено подключение 30 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1406	Запущено подключение 31 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1407	Запущено подключение 32 для обмена данными через EtherNet/IP	○	OFF	–	–	H	R/W	OFF
SM1408	Произошла ошибка в подключении 1 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1409	Произошла ошибка в подключении 2 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1410	Произошла ошибка в подключении 3 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1411	Произошла ошибка в подключении 4 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1412	Произошла ошибка в подключении 5 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1413	Произошла ошибка в подключении 6 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1414	Произошла ошибка в подключении 7 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1415	Произошла ошибка в подключении 8 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1416	Произошла ошибка в подключении 9 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1417	Произошла ошибка в подключении 10 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1418	Произошла ошибка в подключении 11 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1419	Произошла ошибка в подключении 12 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1420	Произошла ошибка в подключении 13 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1421	Произошла ошибка в подключении 14 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1422	Произошла ошибка в подключении 15 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1423	Произошла ошибка в подключении 16 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1424	Произошла ошибка в подключении 17 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1425	Произошла ошибка в подключении 18 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1426	Произошла ошибка в подключении 19 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1427	Произошла ошибка в подключении 20 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF
SM1428	Произошла ошибка в подключении 21 для обмена данными через EtherNet/IP	○	OFF	–	–	N	R	OFF



SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1429	Произошла ошибка в подключении 22 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1430	Произошла ошибка в подключении 23 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1431	Произошла ошибка в подключении 24 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1432	Произошла ошибка в подключении 25 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1433	Произошла ошибка в подключении 26 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1434	Произошла ошибка в подключении 27 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1435	Произошла ошибка в подключении 28 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1436	Произошла ошибка в подключении 29 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1437	Произошла ошибка в подключении 30 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1438	Произошла ошибка в подключении 31 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1439	Произошла ошибка в подключении 32 для обмена данными через EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1440	Ошибка подключения 1 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1441	Ошибка подключения 2 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1442	Ошибка подключения 3 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1443	Ошибка подключения 4 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1444	Ошибка подключения 5 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1445	Ошибка подключения 6 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1446	Ошибка подключения 7 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1447	Ошибка подключения 8 для входов/выходов EtherNet/IP	○	OFF	-	-	N	R	OFF
SM1581	Обновление и сброс соединения по CAN управления оси 1 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1582	Обновление и сброс соединения по CAN управления оси 2 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1583	Обновление и сброс соединения по CAN управления оси 3 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1584	Обновление и сброс соединения по CAN управления оси 4 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1585	Обновление и сброс соединения по CAN управления оси 5 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1586	Обновление и сброс соединения по CAN управления оси 6 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1587	Обновление и сброс соединения по CAN управления оси 7 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1588	Обновление и сброс соединения по CAN управления оси 8 сервопривода Delta	○	OFF	OFF	-	N	R/W	OFF
SM1601	Обновление и сброс соединения по CAN управления оси 21 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1602	Обновление и сброс соединения по CAN управления оси 22 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1603	Обновление и сброс соединения по CAN управления оси 23 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1604	Обновление и сброс соединения по CAN управления оси 24 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1605	Обновление и сброс соединения по CAN управления оси 25 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1606	Обновление и сброс соединения по CAN управления оси 26 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1607	Обновление и сброс соединения по CAN управления оси 27 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1608	Обновление и сброс соединения по CAN управления оси 28 ПЧ Delta	○	OFF	OFF	-	N	R/W	OFF
SM1611	Старт соединения по CAN ПЧ Delta ID 21	○	OFF	OFF	-	N	R	OFF
SM1612	Старт соединения по CAN ПЧ Delta ID 22	○	OFF	OFF	-	N	R	OFF
SM1613	Старт соединения по CAN ПЧ Delta ID 23	○	OFF	OFF	-	N	R	OFF
SM1614	Старт соединения по CAN ПЧ Delta ID 24	○	OFF	OFF	-	N	R	OFF
SM1615	Старт соединения по CAN ПЧ Delta ID 25	○	OFF	OFF	-	N	R	OFF
SM1616	Старт соединения по CAN ПЧ Delta ID 26	○	OFF	OFF	-	N	R	OFF
SM1617	Старт соединения по CAN ПЧ Delta ID 27	○	OFF	OFF	-	N	R	OFF
SM1618	Старт соединения по CAN ПЧ Delta ID 28	○	OFF	OFF	-	N	R	OFF
SM1621	Старт соединения по CAN ПЧ Delta ID 21	○	OFF	OFF	-	N	R	OFF
SM1622	Старт соединения по CAN ПЧ Delta ID 22	○	OFF	OFF	-	N	R	OFF
SM1623	Старт соединения по CAN ПЧ Delta ID 23	○	OFF	OFF	-	N	R	OFF
SM1624	Старт соединения по CAN ПЧ Delta ID 24	○	OFF	OFF	-	N	R	OFF
SM1625	Старт соединения по CAN ПЧ Delta ID 25	○	OFF	OFF	-	N	R	OFF
SM1626	Старт соединения по CAN ПЧ Delta ID 26	○	OFF	OFF	-	N	R	OFF
SM1627	Старт соединения по CAN ПЧ Delta ID 27	○	OFF	OFF	-	N	R	OFF
SM1628	Старт соединения по CAN ПЧ Delta ID 28	○	OFF	OFF	-	N	R	OFF
SM1631	Позиционирование для оси 1 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1632	Позиционирование для оси 2 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1633	Позиционирование для оси 3 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1634	Позиционирование для оси 4 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1635	Позиционирование для оси 5 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1636	Позиционирование для оси 6 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1637	Позиционирование для оси 7 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF
SM1638	Позиционирование для оси 8 ASD-A2 CAN выполнено	○	OFF	OFF	-	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1641	Останов коммуникации для оси 1 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1642	Останов коммуникации для оси 2 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1643	Останов коммуникации для оси 3 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1644	Останов коммуникации для оси 4 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1645	Останов коммуникации для оси 5 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1646	Останов коммуникации для оси 6 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1647	Останов коммуникации для оси 7 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1648	Останов коммуникации для оси 8 ASD-A2 CAN	○	OFF	OFF	–	N	R/W	OFF
SM1651	Сервопривод запущен для оси 1 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1652	Сервопривод запущен для оси 2 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1653	Сервопривод запущен для оси 3 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1654	Сервопривод запущен для оси 4 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1655	Сервопривод запущен для оси 5 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1656	Сервопривод запущен для оси 6 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1657	Сервопривод запущен для оси 7 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1658	Сервопривод запущен для оси 8 ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1661	Функция перемещения назад и вперед включена для оси 1 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1662	Функция перемещения назад и вперед включена для оси 2 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1663	Функция перемещения назад и вперед включена для оси 3 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1664	Функция перемещения назад и вперед включена для оси 4 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1665	Функция перемещения назад и вперед включена для оси 5 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1666	Функция перемещения назад и вперед включена для оси 6 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1667	Функция перемещения назад и вперед включена для оси 7 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1668	Функция перемещения назад и вперед включена для оси 8 ASD-A2 CAN.	○	OFF	OFF	–	N	R/W	OFF
SM1671	Флаг индикации направления перемещения для оси 1 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1672	Флаг индикации направления перемещения для оси 2 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1673	Флаг индикации направления перемещения для оси 3 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1674	Флаг индикации направления перемещения для оси 4 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1675	Флаг индикации направления перемещения для оси 5 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1676	Флаг индикации направления перемещения для оси 6 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1677	Флаг индикации направления перемещения для оси 7 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1678	Флаг индикации направления перемещения для оси 8 ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1681	Выполнены инициализация и коммуникация для ASD-A2 CAN.	○	OFF	OFF	–	N	R	OFF
SM1682	Флаг ошибки коммуникации CANopen для ASD-A2 CAN	○	OFF	OFF	–	N	R	OFF
SM1683	Завершение инициализации и коммуникации CANopen для серводвигателя Delta	○	OFF	OFF	–	N	R	OFF
SM1684	Задание включения/отключения коммуникации CANopen при потере соединения: ВЫКЛ: Остановка всех соединений ВКЛ: Остановка только потерянных соединений	○	OFF	OFF	–	N	R/W	OFF
SM1691	Ошибка соединения по CAN с осью 1	○	OFF	OFF	–	N	R	OFF
SM1692	Ошибка соединения по CAN с осью 2	○	OFF	OFF	–	N	R	OFF
SM1693	Ошибка соединения по CAN с осью 3	○	OFF	OFF	–	N	R	OFF
SM1694	Ошибка соединения по CAN с осью 4	○	OFF	OFF	–	N	R	OFF
SM1695	Ошибка соединения по CAN с осью 5	○	OFF	OFF	–	N	R	OFF
SM1696	Ошибка соединения по CAN с осью 6	○	OFF	OFF	–	N	R	OFF
SM1697	Ошибка соединения по CAN с осью 7	○	OFF	OFF	–	N	R	OFF
SM1698	Ошибка соединения по CAN с осью 8	○	OFF	OFF	–	N	R	OFF
SM1709	Ошибка инициализации при обмене данными через плату расширения 1	○	OFF	–	–	N	R/W	OFF
SM1710	Обмен данными через плату расширения 1 активирован ПО ISPSoft	○	OFF	–	OFF	N	R/W	OFF
SM1712	Старт соединения 1 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1713	Старт соединения 2 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1714	Старт соединения 3 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1715	Старт соединения 4 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1716	Старт соединения 5 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1717	Старт соединения 6 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1718	Старт соединения 7 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1719	Старт соединения 8 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1720	Старт соединения 9 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF

2

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1721	Старт соединения 10 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1722	Старт соединения 11 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1723	Старт соединения 12 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1724	Старт соединения 13 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1725	Старт соединения 14 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1726	Старт соединения 15 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1727	Старт соединения 16 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1728	Старт соединения 17 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1729	Старт соединения 18 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1730	Старт соединения 19 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1731	Старт соединения 20 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1732	Старт соединения 21 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1733	Старт соединения 22 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1734	Старт соединения 23 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1735	Старт соединения 24 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1736	Старт соединения 25 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1737	Старт соединения 26 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1738	Старт соединения 27 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1739	Старт соединения 28 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1740	Старт соединения 29 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1741	Старт соединения 30 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF
SM1742	Старт соединения 31 на плате расширения 1 для обмена данными	○	OFF	-	-	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1743	Старт соединения 32 на плате расширения 1 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1744	Успешный обмен данными по соединению 1 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1745	Успешный обмен данными по соединению 2 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1746	Успешный обмен данными по соединению 3 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1747	Успешный обмен данными по соединению 4 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1748	Успешный обмен данными по соединению 5 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1749	Успешный обмен данными по соединению 6 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1750	Успешный обмен данными по соединению 7 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1751	Успешный обмен данными по соединению 8 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1752	Успешный обмен данными по соединению 9 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1753	Успешный обмен данными по соединению 10 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1754	Успешный обмен данными по соединению 11 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1755	Успешный обмен данными по соединению 12 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1756	Успешный обмен данными по соединению 13 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1757	Успешный обмен данными по соединению 14 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1758	Успешный обмен данными по соединению 15 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1759	Успешный обмен данными по соединению 16 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1760	Успешный обмен данными по соединению 17 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1761	Успешный обмен данными по соединению 18 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1762	Успешный обмен данными по соединению 19 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1763	Успешный обмен данными по соединению 20 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1764	Успешный обмен данными по соединению 21 на плате расширения 1	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1765	Успешный обмен данными по соединению 22 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1766	Успешный обмен данными по соединению 23 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1767	Успешный обмен данными по соединению 24 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1768	Успешный обмен данными по соединению 25 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1769	Успешный обмен данными по соединению 26 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1770	Успешный обмен данными по соединению 27 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1771	Успешный обмен данными по соединению 28 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1772	Успешный обмен данными по соединению 29 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1773	Успешный обмен данными по соединению 30 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1774	Успешный обмен данными по соединению 31 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1775	Успешный обмен данными по соединению 32 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1776	Ошибка обмена данными по соединению 1 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1777	Ошибка обмена данными по соединению 2 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1778	Ошибка обмена данными по соединению 3 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1779	Ошибка обмена данными по соединению 4 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1780	Ошибка обмена данными по соединению 5 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1781	Ошибка обмена данными по соединению 6 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1782	Ошибка обмена данными по соединению 7 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1783	Ошибка обмена данными по соединению 8 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1784	Ошибка обмена данными по соединению 9 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1785	Ошибка обмена данными по соединению 10 на плате расширения 1	○	OFF	-	-	N	R	OFF
SM1786	Ошибка обмена данными по соединению 11 на плате расширения 1	○	OFF	-	-	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1787	Ошибка обмена данными по соединению 12 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1788	Ошибка обмена данными по соединению 13 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1789	Ошибка обмена данными по соединению 14 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1790	Ошибка обмена данными по соединению 15 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1791	Ошибка обмена данными по соединению 16 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1792	Ошибка обмена данными по соединению 17 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1793	Ошибка обмена данными по соединению 18 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1794	Ошибка обмена данными по соединению 19 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1795	Ошибка обмена данными по соединению 20 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1796	Ошибка обмена данными по соединению 21 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1797	Ошибка обмена данными по соединению 22 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1798	Ошибка обмена данными по соединению 23 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1799	Ошибка обмена данными по соединению 24 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1800	Ошибка обмена данными по соединению 25 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1801	Ошибка обмена данными по соединению 26 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1802	Ошибка обмена данными по соединению 27 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1803	Ошибка обмена данными по соединению 28 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1804	Ошибка обмена данными по соединению 29 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1805	Ошибка обмена данными по соединению 30 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1806	Ошибка обмена данными по соединению 31 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1807	Ошибка обмена данными по соединению 32 на плате расширения 1	○	OFF	–	–	N	R	OFF
SM1821	Ошибка инициализации при обмене данными через плату расширения 2	○	OFF	–	–	N	R/W	OFF



SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1822	Обмен данными через плату расширения 2 активирован ПО ISPSOft	○	OFF	–	OFF	N	R/W	OFF
SM1824	Старт соединения 1 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1825	Старт соединения 2 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1826	Старт соединения 3 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1827	Старт соединения 4 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1828	Старт соединения 5 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1829	Старт соединения 6 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1830	Старт соединения 7 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1831	Старт соединения 8 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1832	Старт соединения 9 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1833	Старт соединения 10 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1834	Старт соединения 11 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1835	Старт соединения 12 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1836	Старт соединения 13 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1837	Старт соединения 14 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1838	Старт соединения 15 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1839	Старт соединения 16 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1840	Старт соединения 17 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1841	Старт соединения 18 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1842	Старт соединения 19 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1843	Старт соединения 20 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1844	Старт соединения 21 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1845	Старт соединения 22 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1846	Старт соединения 23 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1847	Старт соединения 24 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1848	Старт соединения 25 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1849	Старт соединения 26 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1850	Старт соединения 27 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1851	Старт соединения 28 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1852	Старт соединения 29 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1853	Старт соединения 30 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1854	Старт соединения 31 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1855	Старт соединения 32 на плате расширения 2 для обмена данными	○	OFF	–	–	N	R/W	OFF
SM1856	Успешный обмен данными по соединению 1 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1857	Успешный обмен данными по соединению 2 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1858	Успешный обмен данными по соединению 3 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1859	Успешный обмен данными по соединению 4 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1860	Успешный обмен данными по соединению 5 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1861	Успешный обмен данными по соединению 6 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1862	Успешный обмен данными по соединению 7 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1863	Успешный обмен данными по соединению 8 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1864	Успешный обмен данными по соединению 9 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1865	Успешный обмен данными по соединению 10 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1866	Успешный обмен данными по соединению 11 на плате расширения 2	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1867	Успешный обмен данными по соединению 12 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1868	Успешный обмен данными по соединению 13 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1869	Успешный обмен данными по соединению 14 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1870	Успешный обмен данными по соединению 15 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1871	Успешный обмен данными по соединению 16 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1872	Успешный обмен данными по соединению 17 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1873	Успешный обмен данными по соединению 18 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1874	Успешный обмен данными по соединению 19 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1875	Успешный обмен данными по соединению 20 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1876	Успешный обмен данными по соединению 21 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1877	Успешный обмен данными по соединению 22 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1878	Успешный обмен данными по соединению 23 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1879	Успешный обмен данными по соединению 24 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1880	Успешный обмен данными по соединению 25 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1881	Успешный обмен данными по соединению 26 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1882	Успешный обмен данными по соединению 27 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1883	Успешный обмен данными по соединению 28 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1884	Успешный обмен данными по соединению 29 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1885	Успешный обмен данными по соединению 30 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1886	Успешный обмен данными по соединению 31 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1887	Успешный обмен данными по соединению 32 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1888	Ошибка обмена данными по соединению 1 на плате расширения 2	○	OFF	–	–	N	R	OFF

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1889	Ошибка обмена данными по соединению 2 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1890	Ошибка обмена данными по соединению 3 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1891	Ошибка обмена данными по соединению 4 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1892	Ошибка обмена данными по соединению 5 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1893	Ошибка обмена данными по соединению 6 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM18934	Ошибка обмена данными по соединению 7 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1895	Ошибка обмена данными по соединению 8 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1896	Ошибка обмена данными по соединению 9 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1897	Ошибка обмена данными по соединению 10 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1898	Ошибка обмена данными по соединению 11 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1899	Ошибка обмена данными по соединению 12 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1900	Ошибка обмена данными по соединению 13 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1901	Ошибка обмена данными по соединению 14 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1902	Ошибка обмена данными по соединению 15 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1903	Ошибка обмена данными по соединению 16 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1904	Ошибка обмена данными по соединению 17 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1905	Ошибка обмена данными по соединению 18 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1906	Ошибка обмена данными по соединению 19 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1907	Ошибка обмена данными по соединению 20 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1908	Ошибка обмена данными по соединению 21 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1909	Ошибка обмена данными по соединению 22 на плате расширения 2	○	OFF	–	–	N	R	OFF
SM1910	Ошибка обмена данными по соединению 23 на плате расширения 2	○	OFF	–	–	N	R	OFF

2

SM	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SM1911	Ошибка обмена данными по соединению 24 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1912	Ошибка обмена данными по соединению 25 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1913	Ошибка обмена данными по соединению 26 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1914	Ошибка обмена данными по соединению 27 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1915	Ошибка обмена данными по соединению 28 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1916	Ошибка обмена данными по соединению 29 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1917	Ошибка обмена данными по соединению 30 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1918	Ошибка обмена данными по соединению 31 на плате расширения 2	○	OFF	-	-	N	R	OFF
SM1919	Ошибка обмена данными по соединению 32 на плате расширения 2	○	OFF	-	-	N	R	OFF

\* 1: для элементов с отметкой \* дополнительную информацию см. В Дополнительном примечании по специальным вспомогательным реле и специальным регистрам данных.

\* 2: выполняется в соответствии с параметрами, установленными в HWCONFIG, когда состояние SM изменяется от ВЫКЛ к ВКЛ, состояние -, и энергонезависимая область отмечена как N.

\* 3: коммуникационные платы здесь: AS-F232, AS-F422 и AS-F485.

### 2.2.8 Условия обновления состояния специальных вспомогательных реле

Специальное вспомогательное реле	Условия обновления состояния
SM0~SM1	Система автоматически устанавливает флаг в положение ВКЛ и сбрасывает его на ВЫКЛ. Флаг автоматически устанавливается в положение ВКЛ при возникновении ошибки операции
SM5	Система автоматически устанавливает флаг в положение ВКЛ и сбрасывает его на ВЫКЛ. Флаг автоматически устанавливается в ВКЛ, когда ошибка, возникшая во время работы программы, записывается в память ПЛК
SM6	При включении система проверяет потерю данных в энергонезависимой области памяти, если данные потеряны, флаг включается. Пользователь может сбросить его в состояние ВЫКЛ вручную
SM7	Когда напряжение питания (номинально 24 В) слишком низкое, флаг включается. Пользователь может сбросить его в состояние ВЫКЛ вручную
SM8	Система автоматически устанавливает SM8 в положение ВКЛ и сбрасывает его на ВЫКЛ. SM8 автоматически устанавливается в ВКЛ при возникновении ошибки сторожевого таймера
SM9	Система автоматически устанавливает SM9 в положение ВКЛ и сбрасывает его на ВЫКЛ. SM9 автоматически устанавливается в ВКЛ при возникновении системной ошибки

Специальное вспомогательное реле	Условия обновления состояния
SM10	Система автоматически устанавливает SM10 в положение ВКЛ и сбрасывает его на ВЫКЛ. SM10 автоматически устанавливается в ВКЛ при возникновении ошибки на входах/выходах
SM22~SM24	Пользователь может установить флаг в положение ВКЛ, а система автоматически сбрасывает его на ВЫКЛ. Журнал очищается, когда данный флаг включен
SM25~SM26	Когда пользователь редактирует программу через ISPSoft, флаг включен; при выходе из ISPSoft, флаг автоматически отключается
SM28	При проверке системой наличия неисправностей флаг включается при нахождении неисправности. Пользователь может сбросить флаг вручную
SM30	Флаг автоматически включается при возникновении ошибки удаленного модуля. Система автоматически сбрасывает флаг в состояние ВЫКЛ
SM34	Флаг включается при вводе неверного пароля. Система автоматически сбрасывает флаг в состояние ВЫКЛ
SM36	Пользователь может установить флаг в положение ВКЛ, и система сохранит данные на карте памяти. По завершении сохранения система автоматически сбрасывает флаг в положение ВЫКЛ
SM76~SM77	Пользователь может установить флаг в положение ВКЛ, и система организует коммуникацию. По завершении сохранения система автоматически сбрасывает флаг в положение ВЫКЛ
SM78~SM79	Флаг включен, пока выполняется процесс коммуникации. По завершении система автоматически сбрасывает его в положение ВЫКЛ
SM80~SM81	Флаг включен после завершения приема данных. Пользователь может сбросить его в состояние ВЫКЛ вручную
SM82~SM83	Флаг включен при возникновении ошибки в ответе. Пользователь может сбросить его в состояние ВЫКЛ вручную
SM84~SM85	Флаг включен при превышении времени ожидания при коммуникации. Пользователь может сбросить его в состояние ВЫКЛ вручную
SM86~SM87	Пользователь может включать и отключать флаг вручную. ВКЛ: 8-битовый режим ВЫКЛ: 16-битовый режим
SM90~SM91	Пользователь может задать флагу состояние ВКЛ. После изменения протокола связи система сбрасывает его на ВЫКЛ
SM94~SM95	После подачи питания флаг включается / выключается в соответствии с настройками в HWCONFIG; пользователь может изменить этот параметр
SM96~SM97	Пользователь может задать флагу состояние ВКЛ. После приема данных система сбрасывает его на ВЫКЛ
SM98~SM99	Флаг включен в процессе коммуникации. После завершения коммуникации флаг автоматически сбрасывается
SM100~SM101	Система автоматически включает флаг, пользователь может выключить его вручную Флаг автоматически включается при получении команды
SM102~SM103	Система автоматически включает флаг, пользователь может выключить его вручную Флаг автоматически включается при ошибочной полученной команде
SM104~SM105	Система автоматически включает флаг, пользователь может выключить его вручную Флаг автоматически включается при превышении времени ожидания на приеме
SM106~SM107	Пользователь может вручную включить/выключить флаг ВКЛ: 8-битовый режим ВЫКЛ: 16-битовый режим
SM166~SM167	Пользователь может вручную включить/выключить флаг
SM168~SM171	Система автоматически включает/выключает флаг

Специальное вспомогательное реле	Условия обновления состояния
SM204~SM205	Пользователь включает флаг вручную, система автоматически отключает флаг ВКЛ: Очистка энергозависимой/энергонезависимой области памяти
SM206	Пользователь может вручную включить/выключить флаг ВКЛ: Запрет выхода
SM209	Пользователь включает флаг вручную, система автоматически отключает флаг ВКЛ: Изменение протокола коммуникации на COM1
SM210	Пользователь может включить/выключить флаг для COM1 ВКЛ: Режим RTU ВЫКЛ: Режим ASCII
SM211	Пользователь включает флаг вручную, система автоматически отключает флаг ВКЛ: Изменение протокола коммуникации на COM2
SM212	Пользователь может включить/выключить флаг для COM2 ВКЛ: Режим RTU ВЫКЛ: Режим ASCII
SM215	Пользователь может вручную включить/выключить флаг ВКЛ: ПЛК работает ВЫКЛ: ПЛК остановлен
SM218	Система проверяет часы реального времени при включении питания. ON: часы реального времени неисправны Пользователь может вручную выключить флаг
SM219	Система контролирует мощность батареи часов реального времени. ВКЛ: мощность низкая Система отключает флаг автоматически
SM220	Пользователь может вручную включить/выключить флаг ВКЛ: Калибровка часов реального времени в пределах $\pm 30$ секунд
SM221	Состояние флага обновляется в соответствии с настройками в HWCONFIG или при выполнении инструкции API1607 DST. ВКЛ: Инструкция DST выполняется
SM270 ~ SM275	Флаг включен при выполнении инструкции CSFO. ВКЛ: Включение функции реверсирования направления ввода ВЫКЛ: Отключение функции реверсирования направления ввода
SM281~SM288	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM291~SM296	Пользователь может вручную включить/выключить флаг ВКЛ: Разрешение функции очистки входов ВЫКЛ: Запрет функции очистки выходов
SM300	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM301~SM303	Система автоматически включает/выключает флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM304	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх

Специальное вспомогательное реле	Условия обновления состояния
SM305~SM307	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM308	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM309~SM311	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM312	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM313~SM315	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM316	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM317~SM319	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM320	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM321~SM323	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM332	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM333~SM335	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM336	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM337~SM339	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM340	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM341	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх



Специальное вспомогательное реле	Условия обновления состояния
SM342	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM343	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM344	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM345	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM346	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM347	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM348	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM349	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM350	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM351	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM352~SM353	Пользователь может вручную включить/выключить флаг ВКЛ: Подсчет сверху вниз ВЫКЛ: Подсчет снизу вверх
SM400~SM403	Система автоматически включает/выключает флаг Состояние флага обновляется каждый цикл сканирования
SM404	Система автоматически включает/выключает флаг SM404 обновляется каждые 5 мс
SM405	Система автоматически включает/выключает флаг SM405 обновляется каждые 50 мс
SM406	Система автоматически включает/выключает флаг SM406 обновляется каждые 100 мс
SM407	Система автоматически включает/выключает флаг SM407 обновляется каждые 500 мс
SM450	Система автоматически включает/выключает флаг ВКЛ: Карта памяти помещена в ПЛК ВЫКЛ: Карта памяти извлечена из ПЛК
SM452~SM453	Система автоматически включает/выключает флаг



Специальное вспомогательное реле	Условия обновления состояния
SM540	Система автоматически включает/выключает флаг
SM541	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM542~SM544	Пользователь может вручную включить/выключить флаг
SM545	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM546	Пользователь может вручную включить/выключить флаг
SM547	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM548~SM551	Пользователь может вручную включить/выключить флаг
SM552	Система автоматически включает/выключает флаг
SM553	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM554~SM557	Пользователь может вручную включить/выключить флаг
SM560	Система автоматически включает/выключает флаг
SM561	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM562~SM564	Пользователь может вручную включить/выключить флаг
SM565	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM566	Пользователь может вручную включить/выключить флаг
SM567	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM568~SM569	Пользователь может вручную включить/выключить флаг
SM572	Система автоматически включает/выключает флаг
SM573	Система автоматически включает флаг, пользователь вручную может выключить флаг
SM574	Пользователь может вручную включить/выключить флаг
SM580	Пользователь может вручную включить флаг, а система отключает флаг автоматически ВКЛ: Запрет функции высокоскоростного выхода
SM600~SM602	Система автоматически включает/выключает флаг Флаг обновляет состояние при выполнении инструкции
SM604	Пользователь может вручную включить/выключить флаг ВКЛ: Сортировка по убыванию ВЫКЛ: Сортировка по возрастанию
SM605	Пользователь может вручную включить/выключить флаг
SM606	Пользователь может вручную включить/выключить флаг ВКЛ: 8-битовый режим ВЫКЛ: 16-битовый режим
SM607	Пользователь может вручную включить/выключить флаг
SM608	Флаг обновляет состояние при выполнении инструкции
SM609	Пользователь может вручную включить/выключить флаг
SM610~SM611	Флаг обновляет состояние при выполнении инструкции
SM612~SM613	Пользователь может вручную включить/выключить флаг
SM614	Флаг обновляет состояние при выполнении инструкции
SM615~SM617	Пользователь может вручную включить/выключить флаг
SM618	Флаг обновляет состояние при выполнении инструкции
SM619	Флаг обновляет состояние при выполнении инструкции EI или DI
SM620	Флаг обновляет состояние при выполнении инструкции CMPT
SM621~SM686	Пользователь может вручную включить/выключить флаг
SM687	Флаг обновляет состояние при выполнении инструкции RAMP
SM688	Флаг обновляет состояние при выполнении инструкции INCD
SM690~SM691	Пользователь может вручную включить/выключить флаг

Специальное вспомогательное реле	Условия обновления состояния
SM692	Флаг обновляет состояние при выполнении инструкции НКУ
SM693	Флаг обновляет состояние при выполнении инструкции SEGL
SM694	Флаг обновляет состояние при выполнении инструкции DSW
SM695	Пользователь может вручную включить/выключить флаг
SM749	Каждый раз после загрузки параметров обмена данными система обновляется при включении питания
SM750~SM783	После загрузки параметров обмена данными пользователь может установить флаг в состояние ВКЛ или ВЫКЛ
SM784~SM847	Флаг включается при обновлении системы
SM861	Каждый раз после загрузки параметров обмена данными система обновляется при включении питания
SM862~SM895	После загрузки параметров обмена данными пользователь может установить флаг в состояние ВКЛ или ВЫКЛ
SM896~SM959	Флаг включен, когда система обновляется автоматически
SM976~SM990	Флаг включен, когда система обновляется автоматически
SM1000	Пользователь может задать флаг в состояние ВКЛ; после сохранения система сбрасывает флаг в состояние ВЫКЛ
SM1001	Флаг включен, если подключение Ethernet активно Флаг выключен, если подключение Ethernet неактивно
SM1090	SM1090 включен, когда соединение TCP занято
SM1091	SM1091 включен, когда соединение UDP занято
SM1100	Флаг обновляет свое состояние при выполнении инструкций API2200-API2210 или после повторного подключения сетевого кабеля
SM1106	SM1106 включен, когда инициализация PHY завершается с ошибкой
SM1107	SM1107 включен, когда IP-адрес, адрес сетевой маски и адрес шлюза установлены некорректно
SM1109	SM1109 включен, когда функция сокета включена и используется один и тот же порт
SM1111	Пользователь может вручную включить/выключить флаг
SM1113	Флаг включен при возникновении ошибки на сервере
SM1116	SM1116 включен, когда активирован триггер параметра ПЛК
SM1117	SM1117 включен, когда активирован триггер параметра ПЛК
SM1119	SM1119 включен, когда триггер активирован и последнее сообщение отправлено успешно
SM1120	SM1120 включен, когда триггер активирован и последнее сообщение отправлено с ошибкой
SM1122~SM1123	Флаг включен, когда триггер активирован и превышено время ответа SMTP-сервера
SM1124	SM1124 включен, когда триггер активирован и размер вложения превышает допустимый предел
SM1125	SM1125 включен, когда триггер активирован и вложение не найдено
SM1126~SM1127	Флаг включен, когда активирован триггер параметра ПЛК
SM1129	SM1129 включен, когда триггер активирован и последнее сообщение отправлено успешно
SM1130	SM1130 включен, когда триггер активирован и последнее сообщение отправлено с ошибкой
SM1132	SM1132 включен, когда триггер активирован и превышено время ответа SMTP-сервера
SM1133	SM1133 включен, когда триггер активирован и возникла ошибка ответа SMTP-сервера
SM1134	SM1134 включен, когда триггер активирован и размер вложения превышает допустимый предел

Специальное вспомогательное реле	Условия обновления состояния
SM1135	SM1135 включен, когда триггер активирован и вложение не найдено
SM1136	SM1136 включен, когда активирован триггер ПЛК
SM1137	SM1137 включен, когда запускается триггер ПЛК.
SM1139	SM1139 включен, когда триггер активирован и последнее сообщение отправлено успешно
SM1140	SM1140 включен, когда триггер активирован и последнее сообщение отправлено с ошибкой
SM1142	SM1142 включен, когда триггер активирован и превышено время ответа SMTP-сервера
SM1143	SM1143 включен, когда триггер активирован и возникла ошибка ответа SMTP-сервера
SM1144	SM1144 включен, когда триггер активирован и размер вложения превышает допустимый предел
SM1145	SM1145 включен, когда триггер активирован и вложение не найдено
SM1146	SM1146 включен, когда активирован триггер ПЛК
SM1147	SM1147 включен, когда запускается триггер ПЛК.
SM1149	SM1149 включен, когда триггер активирован и последнее сообщение отправлено успешно
SM1150	SM1150 включен, когда триггер активирован и последнее сообщение отправлено с ошибкой
SM1152	SM1152 включен, когда триггер активирован и превышено время ответа SMTP-сервера
SM1153	SM1153 включен, когда триггер активирован и возникла ошибка ответа SMTP-сервера
SM1154	SM1154 включен, когда триггер активирован и размер вложения превышает допустимый предел
SM1155	SM1155 включен, когда триггер активирован и вложение не найдено
SM1166	После загрузки параметров обмена данными система обновляется
SM1167~SM1199	После загрузки параметров обмена данными пользователь может установить флаг в состояние ВКЛ или ВЫКЛ
SM1200~SM1263	Флаг включен, когда система обновляется
SM1269	Флаг включен, когда возникает ошибка конфигурации сокета.
SM1270~SM1353	Флаг обновляет свое состояние при выполнении функции сокета
SM1375~SM1407	После загрузки параметров обмена данными через EtherNet / IP пользователь может установить флаг в состояние ВКЛ или ВЫКЛ
SM1408~SM1439	Флаг включен при возникновении ошибки при обмене данными через EtherNet / IP
SM1440~SM1447	Флаг включен, когда превышено время ожидания в Slave устройстве ввода / вывода через EtherNet / IP
SM1581~SM1588	Пользователь может вручную включить/выключить флаг
SM1601~SM1608	Пользователь может вручную включить/выключить флаг
SM1611~SM1618	Система автоматически включает/выключает флаг
SM1621~SM1628	Система автоматически включает/выключает флаг
SM1631~SM1638	Система автоматически включает, а пользователь может вручную выключить флаг
SM1641~SM1648	Пользователь может вручную включить/выключить флаг
SM1651~SM1658	Система автоматически включает/выключает флаг
SM1661~SM1668	Пользователь может вручную включить/выключить флаг
SM1671~SM1682	Система автоматически включает/выключает флаг
SM1683	Система автоматически включает/выключает флаг
SM1684	Пользователь может вручную включить/выключить флаг
SM1691~SM1698	Система автоматически включает/выключает флаг
SM1709~SM1710	Пользователь может вручную включить/выключить флаг

Специальное вспомогательное реле	Условия обновления состояния
SM1712~SM1743	Пользователь может вручную включить/выключить флаг
SM1744~SM1807	Система автоматически включает/выключает флаг
SM1821~SM1822	Пользователь может вручную включить/выключить флаг
SM1824~SM1855	Пользователь может вручную включить/выключить флаг
SM1856~SM1919	Система автоматически включает/выключает флаг

## 2.2.9 Шаговые реле (S)

Функции шаговых реле:

Шаговое реле используется в промышленной автоматизации для задания процедур. Это основной объект в программировании на языке программирования SFC. Более подробную информацию о данном языке программирования см. Руководство пользователя ISPSoft.

В ПЛК реализовано 2048 шаговых реле: S0 ~ S2047. Каждое шаговое реле похоже на выходное реле, поскольку оно имеет выходную катушку, контакт А и контакт В. Его можно использовать несколько раз в программе, но оно не может напрямую управлять внешней нагрузкой. Кроме того, шаговое реле может использоваться как общее вспомогательное реле, когда оно не используется в программировании на языке SFC.

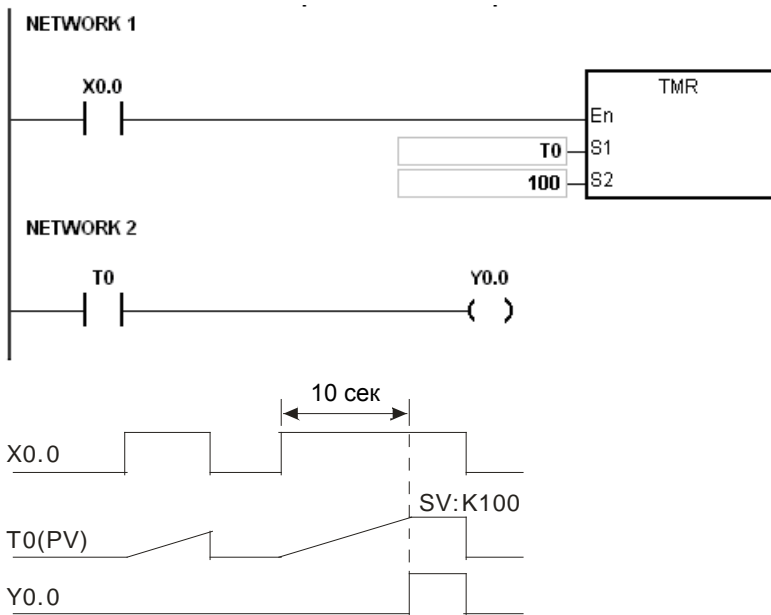
### 2.2.10 Таймеры (Т)

1. Таймер 100 мс: Таймер, применяемый в инструкции TMR, имеет единицу синхронизации 100 мс.
2. Таймер 1 мс: Таймер, применяемый в инструкции TMRH, имеет единицу синхронизации 1 мс.
3. Аккумулятивные таймеры ST0~ST511. Таймеры T0~T511 можно использовать для функции мониторинга устройств.
4. Если один и тот же таймер используется повторно в программе, в том числе в инструкциях TMR и TMRH, значение настройки таймера соответствует первому заданному значению.
5. Если один и тот же таймер используется повторно в программе, он отключается, если отключается один из условных контактов.
6. Если один и тот же таймер используется повторно в программе для исключительного использования в подпрограмме и аккумулятивного таймера в программе, он отключается, если отключается один из условных контактов.
7. Когда таймер выключается и условный контакт включен, таймер сбрасывается и начинает новый подсчет.
8. После выполнения инструкции TMR указанный таймер включен и начинает подсчет. Когда подсчитанное значение таймера будет соответствовать заданному значению, катушка контакта включается.

#### А. Общий таймер

При выполнении инструкции TMR, общий таймер начинает подсчет. Когда подсчитанное значение таймера будет соответствовать заданному значению, катушка выходного контакта включается.

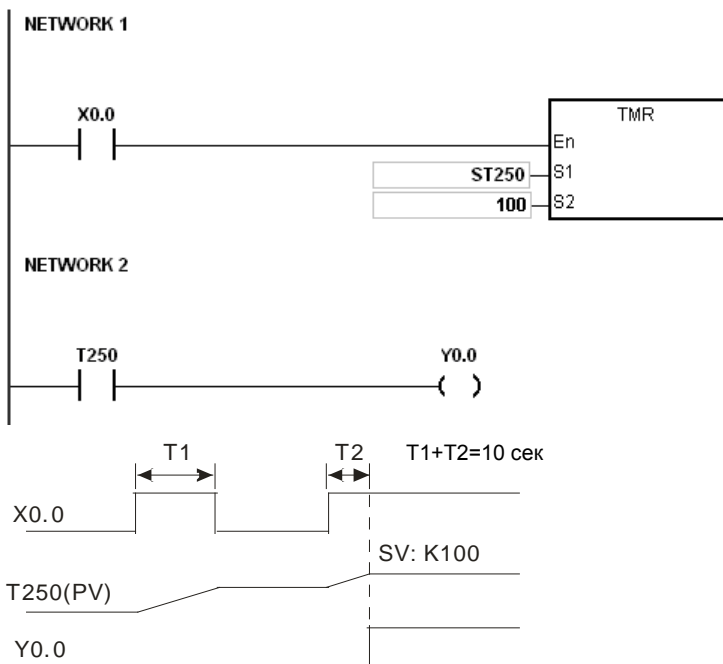
1. Когда контакт X0.0 включен, а таймер имеет единицу синхронизации 100 мс, выходная катушка T0 будет включена при значении таймера 100.
2. Когда контакт X0.0 выключен или питание отключено, значение таймера будет равно 0 и входная катушка T0 будет отключена.



**В. Аккумулятивный таймер**

При выполнении инструкции TMR, аккумулятивный таймер начинает подсчет. Когда значение таймера соответствует заданному значению, выходная катушка включается. Таймер становится аккумулятивным при добавлении пользователем буквы S перед буквой T в обозначении таймера. Когда условный контакт выключен, значение аккумулятивного таймера не сбрасывается. Когда условный контакт включен, таймер будет подсчитывать, начиная с текущего значения.

1. Когда контакт X0.0 включен, а таймер T250 имеет единицу синхронизации 100 мс, выходная катушка T250 включается при значении подсчета таймера T250 равно заданному значению 100.
2. Когда контакт X0.0 выключен или питание отключено, таймер T250 прекращает подсчет и его значение остается неизменным. Когда контакт X0.0 включается снова, значение таймера будет подсчитываться далее, а когда аккумулятивное значение станет равно значению 100, выходная катушка T250 будет включена.



### С. Использование таймера в функциональном блоке

T412~T511 – это таймеры, которые можно использовать в функциональном блоке или в прерывании.

Когда выполняются инструкции TMR или END, таймер начинает подсчет, используемый в функциональном блоке. Когда значение таймера соответствует заданному значению, выходная катушка включается.

Если таймер общего назначения используется в функциональном блоке или прерывании, а функционал не выполняется, таймер не может вести подсчет правильно.

2

#### 2.2.11 Счетчики

- Характеристики 16-битного счетчика

Пункт	16-битный счетчик
Тип	Основной
Нумерация	C0~C511
Направление счета	Вверх
Задаваемое значение	0~32,767
Характеристики заданного значения	Значение настройки может быть либо константой, либо значением в регистре данных
Изменение текущего значения	Счетчик прекращает отсчет, когда его значение достигает заданного значения
Выходной контакт	Контакт включается, когда значение счетчика достигает заданного значения
Сброс значения	При выполнении инструкции RST текущее значение счетчика сбрасывается до нуля, а контакт выключается
Активация контакта	Контакт активируется после выполнения сканирования

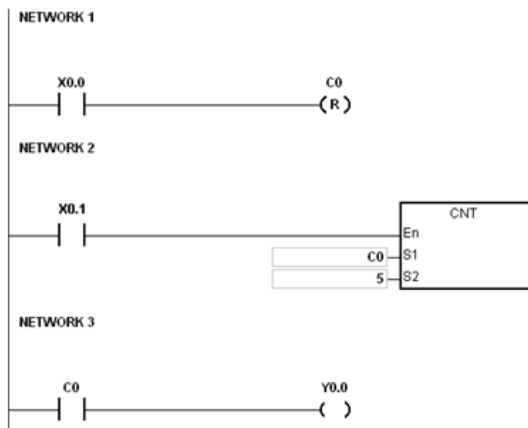
- Функция счетчика

Каждый раз, когда состояние входа переключается с ВЫКЛ на ВКЛ, значение счетчика соответствует состоянию выходной катушки. Пользователь может использовать либо десятичное значение константы, либо значение в регистре данных в качестве заданного значения.

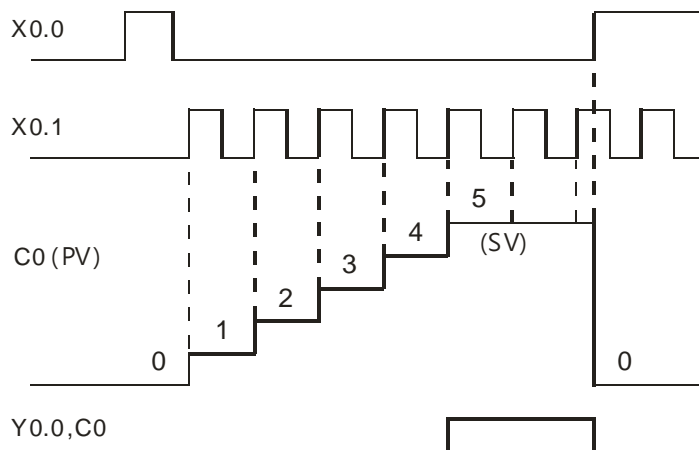
16-битный счетчик:

1. Диапазон задания: 0~32,767 (Значения заданий 0 и 1 означают, что выходной контакт включен, когда счетчик начинает подсчет в первый раз)
2. Для общего счетчика текущее значение счетчика очищается при отключении питания. Если счетчик является фиксированным, текущее значение счетчика и состояние контакта до выключения питания будут сохранены. Зафиксированный счетчик рассчитывается с текущего значения при восстановлении источника питания.
3. Если пользователи используют команду MOV или ISPSoft для передачи значения, большего чем заданное значение, в регистр текущих значений C0, контакт счетчика C0 будет включен, и текущее значение станет заданным в следующий раз, когда X0.1 включается.
4. Пользователи могут использовать либо константу, либо значение в регистре данных в качестве значения задания счетчика.
5. Значение задания счетчика может быть положительным или отрицательным. Если счетчик досчитывает до верхнего предела 32 767, следующее текущее значение становится равным 0.





1. Когда контакт X0.0 включен, инструкция RST будет выполнена и текущее значение счетчика C0 будет сброшено на ноль, а выходной контакт счетчика C0 будет отключен.
2. Когда контакт X0.1 включается, значение счетчика увеличивается на единицу.
3. Когда значение на счетчике C0 достигнет заданного значения 5, контакт счетчика C0 будет включен (текущее значение на C0 = заданное значение = 5). После этого включения контакта X0.1 не будут считываться счетчиком C0 и текущее значение C0 останется на значении 5.



### 2.2.12 32-битные счетчики (HC)

- Характеристики 32-битных счетчиков

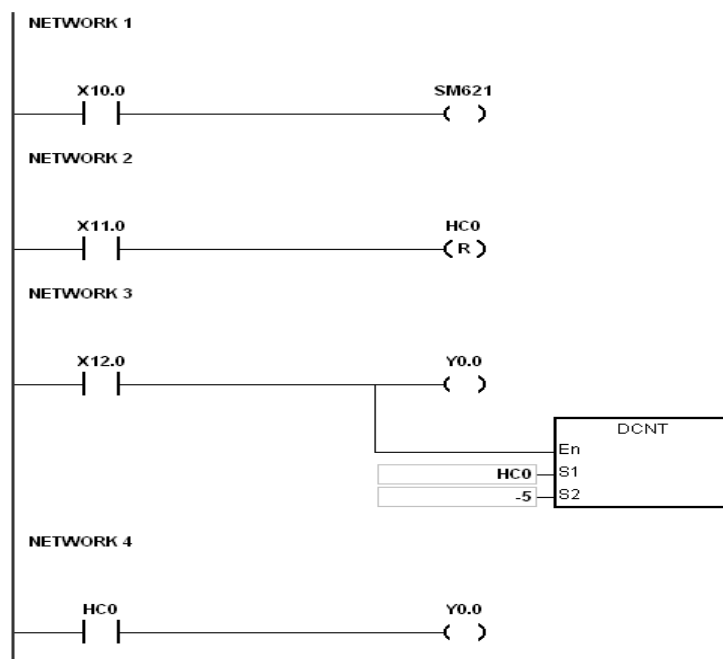
Пункт	32-битный счетчик		
	Счетчик вверх/вниз	Счетчик вверх	Высокоскоростной счетчик
Нумерация	HC0 ~ HC63	HC64 ~ HC199	HC200 ~ HC255
Направление счета	Вверх/вниз	Вверх	Вверх/вниз
Задаваемое значение	-2,147,483,648~+2,147,483,647		
Характеристики заданного значения	Значение настройки может быть либо константой, либо значением, занимающим два (32 бит) регистра данных		
Изменение текущего значения	Счетчик продолжает подсчет даже после того, как значение счетчика достигает заданного значения		

Пункт	32-битный счетчик
Выходной контакт	Контакт включен, когда текущее значение счетчика при подсчете вверх достигает заданного значения Контакт выключен, когда текущее значение счетчика при подсчете вниз снижается меньше заданного значения
Сброс значения	При выполнении инструкции RST текущее значение счетчика сбрасывается до нуля, а контакт выключается
Активация контакта	После выполнения сканирования инструкции DCNT контакт активируется

- 32-битный основной счетчик сложения/вычитания (счетчик вверх/вниз)
  1. Диапазон задания: -2,147,483,648~2,147,483,647
  2. Переключение между 32-битными счетчиками сложения и 32-битными счетчиками вычитания зависит от состояний специальных вспомогательных реле SM621 ~ SM684. Например, счетчик HC0 является счетчиком сложения, когда SM621 выключено, и HC0 является счетчиком вычитания, когда SM621 включено.
  3. Пользователь может использовать константу или значение в регистрах данных как заданное значение счетчика, и оно может быть положительным или отрицательным. При использовании значения в регистрах данных в качестве заданного, значение занимает два последовательных регистра.
  4. Для общего счетчика текущее значение счетчика очищается при отключении питания. Если счетчик является фиксированным, текущее значение счетчика и состояние контакта до выключения питания будут сохранены. Зафиксированный счетчик рассчитывается с текущего значения при восстановлении источника питания.
  5. Если значение счетчика превышает 2,147,483,647, следующее текущее значение составляет -2,147,483,648. Если значение счетчика становится меньше -2,147,483,648, следующее текущее значение составляет 2,147,483,647.
- 32-битный высокоскоростной счетчик вверх/вниз

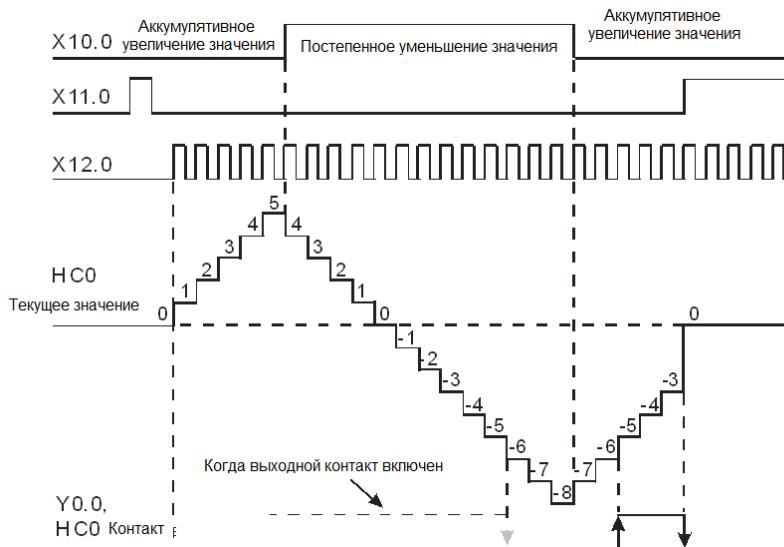
См. описание инструкции API1004 DCNT в настоящем Руководстве по программированию.

Пример:



1. Контакт X10.0 управляет флагом SM621 для задания направления счета (вверх/вниз) счетчика HC0.

2. При включении контакта X11.0, инструкция RST будет выполнена, текущее значение счетчика HC0 обнулится, а его контакт отключится .
3. При включении контакта X12.0, текущее значение счетчика HC0 подсчитывается вверх (+ 1) или вниз (- 1).
4. Когда значение HC0 меняется с -6 на -5, контакт HC0 включается. Когда значение HC0 меняется с -5 на -6, контакт HC0 выключается.
5. Если пользователи используют команду MOV или ISPSofT для передачи значения, большего чем заданное значение, в регистр текущих значений HC0, контакт счетчика HC0 будет включен, и текущее значение станет заданным в следующий раз, когда X12.0 включается.



### 2.2.13 Регистры данных (D)

Регистр данных хранит 16-битные данные. Самый старший бит представляет либо положительный знак, либо отрицательный знак, а значения, которые могут храниться в регистрах данных, варьируются от -32,768 до +32,767. Два 16-разрядных регистра могут быть объединены в 32-битный регистр, то есть (D + 1, D), в котором регистр, число которого меньше, сохраняет младшие 16 бит. Самый старший бит представляет либо положительный знак, либо отрицательный знак, а значения, которые могут храниться в регистрах данных, варьируются от -2,147,483,648 до +2,147,483,647. Кроме того, четыре 16-битных регистра могут быть объединены в 64-битный регистр, то есть (D + 3, D + 2, D + 1, D), в котором регистр, число которого меньше, представляет собой младшие 16 бит. Самый старший бит представляет либо положительный знак, либо отрицательный знак, а значения, которые могут храниться в регистрах данных, варьируются от -9,223,372,036,854,776 до +9,223,372,036,854,775,807. Регистры данных также могут использоваться для обновления значений в регистрах управления в модулях, отличных от модулей дискретного ввода/вывода. Дополнительную информацию об обновлении значений в регистрах управления см. Руководство пользователя ISPSofT..

Регистры могут быть разделены на два типа в зависимости от их свойств.

1. **Общий регистр:** Когда ПЛК начинает работать или отключается, значение в регистре будет очищено до нуля. Если необходимо сохранить данные при запуске ПЛК, см. Руководство пользователя ISPSofT. Обратите внимание, что значение будет в любом случае сбрасываться до нуля при отключении ПЛК.
2. **Энергонезависимый регистр:** Если ПЛК отключается, данные в энергонезависимом регистре не будут очищены. Если необходимо очистить данные в энергонезависимой области, надо применять инструкции RST или ZRST.

### 2.2.14 Специальные регистры данных (SR)

Каждый специальный регистр данных имеет свое определение и специфическую функцию. Системные состояния и сообщения об ошибках сохраняются в специальных регистрах данных. Кроме того, специальные регистры данных могут использоваться для контроля состояния системы. Специальные регистры данных и их функции перечислены ниже. Для SR с пометкой «\*», пользователи могут ознакомиться с дополнительной информацией по специальному вспомогательному реле / специальным регистрам данных. «R» в столбце атрибута указывает, что из специального регистра данных можно только считывать данные, тогда как «R / W» в столбце атрибута указывает, что можно считывать и записывать данные. Указатель “\_” означает, что состояние реле не изменяется, указатель “#” показывает, что установка реле зависит от состояния ПЛК, но можно считывать их значения.

«Y» в столбце означает, что специальный регистр энергонезависимый, «N» - энергозависимы; атрибут «H», означает, что регистр определяется настройками в HWCONFIG.

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR0	Код ошибки выполнения операции ПЛК / операнда	○	0	0	-	N	R	0
SR1	Адрес ошибки выполнения операции (32 бит)	○	0	0	-	N	R	0
SR2								
SR4	Код синтаксической ошибки	○	0	0	-	N	R	0
SR5	Адрес синтаксической ошибки (32 бит)	○	0	0	-	N	R	0
SR6								
*SR8	Адрес шага, на котором включен сторожевой таймер (32 бит)	○	0	-	-	N	R	0
SR9								
SR23	Количество сформированных MAC адресов	○	-	-	-	N	R	-
SR28	Последний номер выхода, на котором наиболее часто используются высокоскоростные инструкции	○	-1	-1	-1	N	R	-1
SR32	Адрес последней инструкции, превысившей допустимый диапазон	○	-1	-1	-	N	R	-1
SR33								
*SR36	Система сохраняет данные на карту памяти. Используется вместе с SM36.	○	0	-	-	N	R/W	0
*SR40	Число журналов ошибок	○	-	-	-	Y	R	0
*SR41	Указатель журнала ошибок	○	-	-	-	Y	R	0
*SR43	Журнал ошибок 1: Указатель модуля	○	-	-	-	Y	R	0
*SR44	Журнал ошибок 1: Код ошибки	○	-	-	-	Y	R	0
*SR45	Журнал ошибок 1: Год и месяц	○	-	-	-	Y	R	0
*SR46	Журнал ошибок 1: День и час	○	-	-	-	Y	R	0
*SR47	Журнал ошибок 1: Минута и секунда	○	-	-	-	Y	R	0
*SR49	Журнал ошибок 2: Указатель модуля	○	-	-	-	Y	R	0
*SR50	Журнал ошибок 2: Код ошибки	○	-	-	-	Y	R	0
*SR51	Журнал ошибок 2: Год и месяц	○	-	-	-	Y	R	0
*SR52	Журнал ошибок 2: День и час	○	-	-	-	Y	R	0
*SR53	Журнал ошибок 2: Минута и секунда	○	-	-	-	Y	R	0
*SR55	Журнал ошибок 3: Указатель модуля	○	-	-	-	Y	R	0
*SR56	Журнал ошибок 3: Код ошибки	○	-	-	-	Y	R	0
*SR57	Журнал ошибок 3: Год и месяц	○	-	-	-	Y	R	0
*SR58	Журнал ошибок 3: День и час	○	-	-	-	Y	R	0
*SR59	Журнал ошибок 3: Минута и секунда	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR61	Журнал ошибок 4: Указатель модуля	○	-	-	-	Y	R	0
*SR62	Журнал ошибок 4: Код ошибки	○	-	-	-	Y	R	0
*SR63	Журнал ошибок 4: Год и месяц	○	-	-	-	Y	R	0
*SR64	Журнал ошибок 4: День и час	○	-	-	-	Y	R	0
*SR65	Журнал ошибок 4: Минута и секунда	○	-	-	-	Y	R	0
*SR67	Журнал ошибок 5: Указатель модуля	○	-	-	-	Y	R	0
*SR68	Журнал ошибок 5: Код ошибки	○	-	-	-	Y	R	0
*SR69	Журнал ошибок 5: Год и месяц	○	-	-	-	Y	R	0
*SR70	Журнал ошибок 5: День и час	○	-	-	-	Y	R	0
*SR71	Журнал ошибок 5: Минута и секунда	○	-	-	-	Y	R	0
*SR73	Журнал ошибок 6: Указатель модуля	○	-	-	-	Y	R	0
*SR74	Журнал ошибок 6: Код ошибки	○	-	-	-	Y	R	0
*SR75	Журнал ошибок 6: Год и месяц	○	-	-	-	Y	R	0
*SR76	Журнал ошибок 6: День и час	○	-	-	-	Y	R	0
*SR77	Журнал ошибок 6: Минута и секунда	○	-	-	-	Y	R	0
*SR79	Журнал ошибок 7: Указатель модуля	○	-	-	-	Y	R	0
*SR80	Журнал ошибок 7: Код ошибки	○	-	-	-	Y	R	0
*SR81	Журнал ошибок 7: Год и месяц	○	-	-	-	Y	R	0
*SR82	Журнал ошибок 7: День и час	○	-	-	-	Y	R	0
*SR83	Журнал ошибок 7: Минута и секунда	○	-	-	-	Y	R	0
*SR85	Журнал ошибок 8: Указатель модуля	○	-	-	-	Y	R	0
*SR86	Журнал ошибок 8: Код ошибки	○	-	-	-	Y	R	0
*SR87	Журнал ошибок 8: Год и месяц	○	-	-	-	Y	R	0
*SR88	Журнал ошибок 8: День и час	○	-	-	-	Y	R	0
*SR89	Журнал ошибок 8: Минута и секунда	○	-	-	-	Y	R	0
*SR91	Журнал ошибок 9: Указатель модуля	○	-	-	-	Y	R	0
*SR92	Журнал ошибок 9: Код ошибки	○	-	-	-	Y	R	0
*SR93	Журнал ошибок 9: Год и месяц	○	-	-	-	Y	R	0
*SR94	Журнал ошибок 9: День и час	○	-	-	-	Y	R	0
*SR95	Журнал ошибок 9: Минута и секунда	○	-	-	-	Y	R	0
*SR97	Журнал ошибок 10: Указатель модуля	○	-	-	-	Y	R	0
*SR98	Журнал ошибок 10: Код ошибки	○	-	-	-	Y	R	0
*SR99	Журнал ошибок 10: Год и месяц	○	-	-	-	Y	R	0
*SR100	Журнал ошибок 10: День и час	○	-	-	-	Y	R	0
*SR101	Журнал ошибок 10: Минута и секунда	○	-	-	-	Y	R	0
*SR103	Журнал ошибок 11: Указатель модуля	○	-	-	-	Y	R	0
*SR104	Журнал ошибок 11: Код ошибки	○	-	-	-	Y	R	0
*SR105	Журнал ошибок 11: Год и месяц	○	-	-	-	Y	R	0
*SR106	Журнал ошибок 11: День и час	○	-	-	-	Y	R	0
*SR107	Журнал ошибок 11: Минута и секунда	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR109	Журнал ошибок 12: Указатель модуля	○	-	-	-	Y	R	0
*SR110	Журнал ошибок 12: Код ошибки	○	-	-	-	Y	R	0
*SR111	Журнал ошибок 12: Год и месяц	○	-	-	-	Y	R	0
*SR112	Журнал ошибок 12: День и час	○	-	-	-	Y	R	0
*SR113	Журнал ошибок 12: Минута и секунда	○	-	-	-	Y	R	0
*SR115	Журнал ошибок 13: Указатель модуля	○	-	-	-	Y	R	0
*SR116	Журнал ошибок 13: Код ошибки	○	-	-	-	Y	R	0
*SR117	Журнал ошибок 13: Год и месяц	○	-	-	-	Y	R	0
*SR118	Журнал ошибок 13: День и час	○	-	-	-	Y	R	0
*SR119	Журнал ошибок 13: Минута и секунда	○	-	-	-	Y	R	0
*SR121	Журнал ошибок 14: Указатель модуля	○	-	-	-	Y	R	0
*SR122	Журнал ошибок 14: Код ошибки	○	-	-	-	Y	R	0
*SR123	Журнал ошибок 14: Год и месяц	○	-	-	-	Y	R	0
*SR124	Журнал ошибок 14: День и час	○	-	-	-	Y	R	0
*SR125	Журнал ошибок 14: Минута и секунда	○	-	-	-	Y	R	0
*SR127	Журнал ошибок 15: Указатель модуля	○	-	-	-	Y	R	0
*SR128	Журнал ошибок 15: Код ошибки	○	-	-	-	Y	R	0
*SR129	Журнал ошибок 15: Год и месяц	○	-	-	-	Y	R	0
*SR130	Журнал ошибок 15: День и час	○	-	-	-	Y	R	0
*SR131	Журнал ошибок 15: Минута и секунда	○	-	-	-	Y	R	0
*SR133	Журнал ошибок 16: Указатель модуля	○	-	-	-	Y	R	0
*SR134	Журнал ошибок 16: Код ошибки	○	-	-	-	Y	R	0
*SR135	Журнал ошибок 16: Год и месяц	○	-	-	-	Y	R	0
*SR136	Журнал ошибок 16: День и час	○	-	-	-	Y	R	0
*SR137	Журнал ошибок 16: Минута и секунда	○	-	-	-	Y	R	0
*SR139	Журнал ошибок 17: Указатель модуля	○	-	-	-	Y	R	0
*SR140	Журнал ошибок 17: Код ошибки	○	-	-	-	Y	R	0
*SR141	Журнал ошибок 17: Год и месяц	○	-	-	-	Y	R	0
SR142	Журнал ошибок 17: День и час	○	-	-	-	Y	R	0
*SR143	Журнал ошибок 17: Минута и секунда	○	-	-	-	Y	R	0
*SR145	Журнал ошибок 18: Указатель модуля	○	-	-	-	Y	R	0
*SR146	Журнал ошибок 18: Код ошибки	○	-	-	-	Y	R	0
*SR147	Журнал ошибок 18: Год и месяц	○	-	-	-	Y	R	0
*SR148	Журнал ошибок 18: День и час	○	-	-	-	Y	R	0
*SR149	Журнал ошибок 18: Минута и секунда	○	-	-	-	Y	R	0
*SR151	Журнал ошибок 19: Указатель модуля	○	-	-	-	Y	R	0
*SR152	Журнал ошибок 19: Код ошибки	○	-	-	-	Y	R	0
*SR153	Журнал ошибок 19: Год и месяц	○	-	-	-	Y	R	0
*SR154	Журнал ошибок 19: День и час	○	-	-	-	Y	R	0
*SR155	Журнал ошибок 19: Минута и секунда	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR157	Журнал ошибок 20: Указатель модуля	○	-	-	-	Y	R	0
*SR158	Журнал ошибок 20: Код ошибки	○	-	-	-	Y	R	0
*SR159	Журнал ошибок 20: Год и месяц	○	-	-	-	Y	R	0
*SR160	Журнал ошибок 20: День и час	○	-	-	-	Y	R	0
*SR161	Журнал ошибок 20: Минута и секунда	○	-	-	-	Y	R	0
SR162	Продолжительность подачи питания на ПЛК (минуты) (32 бит)	○	-	-	-	Y	R	-
SR163								
SR166	Значение VR0 (используется с SM166)	○	0	-	-	N	R	0
SR167	Значение VR1 (используется с SM167)	○	0	-	-	N	R	0
SR168	Значение для канала 1 функциональной платы 1 F2AD	○	0	-	-	N	R	0
SR169	Значение для канала 2 функциональной платы 1 F2AD	○	0	-	-	N	R	0
SR170	Значение для канала 1 функциональной платы 2 F2AD	○	0	-	-	N	R	0
SR171	Значение для канала 2 функциональной платы 2 F2AD	○	0	-	-	N	R	0
SR172	Значение для канала 1 функциональной платы 1 F2DA	○	0	-	0	N	R/W	0
SR173	Значение для канала 2 функциональной платы 1 F2DA	○	0	-	0	N	R/W	0
SR174	Значение для канала 1 функциональной платы 2 F2DA	○	0	-	0	N	R/W	0
SR175	Значение для канала 2 функциональной платы 2 F2DA	○	0	-	0	N	R/W	0
SR176	ID коммуникации для функциональной платы 1 (COM11)	○	-	-	-	N	R	1
SR177	Код протокола для функциональной платы 1 (COM11)	○	-	-	-	N	R	0x24
SR178	ID коммуникации для функциональной платы 2 (COM12)	○	-	-	-	N	R	1
SR179	Код протокола для функциональной платы 2 (COM12)	○	-	-	-	N	R	0x24
SR180	Код последней ошибки, при которой произошло предупреждение	○	0	-	-	N	R	0
SR182	Задержка для коммуникационной платы 1 (COM11) (шаг: мс), если значение 0, задержка отсутствует	○	-	-	-	N	R/W	200
SR183	Задержка для коммуникационной платы 2 (COM12) (шаг: мс), если значение 0, задержка отсутствует	○	-	-	-	N	R/W	200
SR185	Время цикла связи всех удаленных модулей (минуты)	○	0	-	-	N	R	0
SR187	Скорость обмена данными через коммуникационную плату 1 (COM11) (шаг: 100 бит/с)	○	96	-	-	N	R/W	96
SR188	Скорость обмена данными через коммуникационную плату 2 (COM12) (шаг: 100 бит/с)	○	96	-	-	N	R/W	96

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR190	Умножение частоты высокоскоростного счетчика 1 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR191	Умножение частоты высокоскоростного счетчика 2 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR192	Умножение частоты высокоскоростного счетчика 3 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR193	Умножение частоты высокоскоростного счетчика 4 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR194	Умножение частоты высокоскоростного счетчика 5 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR195	Умножение частоты высокоскоростного счетчика 6 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR196	Умножение частоты высокоскоростного счетчика 7 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR197	Умножение частоты высокоскоростного счетчика 8 (по умолчанию: 1-кратное)	○	1	-	-	N	R/W	1
SR198	Pi (π), число с плавающей точкой (32 бит)	○	16#0F DB	16#0FD B	16#0F DB	N	R	16#0F DB
SR199			16#40 49	16#404 9	16#40 49	N		16#40 49
*SR201	Коммуникационный адрес COM1	○	-	-	-	H	R/W	1
*SR202	Коммуникационный адрес COM2	○	-	-	-	H	R/W	1
*SR209	Коммуникационный протокол COM1	○	-	-	-	H	R/W	16#00 24
*SR210	Задержка коммуникации COM1 (мс), 0: задержки нет	○	-	-	-	H	R/W	0
*SR212	Коммуникационный протокол COM2	○	-	-	-	H	R/W	16#00 24
*SR213	Задержка коммуникации COM2 (мс), 0: задержки нет	○	-	-	-	H	R/W	0
*SR215	Наименование функциональной платы 1	○	-	-	-	N	R	0
*SR216	Наименование функциональной платы 2	○	-	-	-	N	R	0
SR217	Скорость передачи данных для COM1 (шаг:100 бит/с)	○	96	-	-	H	R/W	96
SR218	Скорость передачи данных для COM2 (шаг:100 бит/с)	○	96	-	-	H	R/W	96
*SR220	Значение года часов реального времени (RTC): 00~99	○	-	-	-	Y	R	0
*SR221	Значение месяца часов реального времени (RTC): 01~12	○	-	-	-	Y	R	1
*SR222	Значение дня часов реального времени (RTC): 1~31	○	-	-	-	Y	R	1
*SR223	Значение часа часов реального времени (RTC): 00~23	○	-	-	-	Y	R	0
*SR224	Значение минуты часов реального времени (RTC): 00~59	○	-	-	-	Y	R	0
*SR225	Значение секунды часов реального времени (RTC): 00~59	○	-	-	-	Y	R	0
*SR226	Значение дня недели часов реального времени (RTC): 1~7	○	-	-	-	Y	R	1
*SR227	Номер загружаемого журнала (максимум 20)	○	-	-	-	Y	R	0



SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR228	Указатель загружаемого журнала	○	-	-	-	Y	R	0
*SR229	Загружаемый журнал 1: Номер действия	○	-	-	-	Y	R	0
*SR230	Загружаемый журнал 1: Год и месяц	○	-	-	-	Y	R	0
*SR231	Загружаемый журнал 1: День и час	○	-	-	-	Y	R	0
*SR232	Загружаемый журнал 1: Минута и секунда	○	-	-	-	Y	R	0
*SR233	Загружаемый журнал 2: Номер действия	○	-	-	-	Y	R	0
*SR234	Загружаемый журнал 2: Год и месяц	○	-	-	-	Y	R	0
*SR235	Загружаемый журнал 2: День и час	○	-	-	-	Y	R	0
*SR236	Загружаемый журнал 2: Минута и секунда	○	-	-	-	Y	R	0
*SR237	Загружаемый журнал 3: Номер действия	○	-	-	-	Y	R	0
*SR238	Загружаемый журнал 3: Год и месяц	○	-	-	-	Y	R	0
*SR239	Загружаемый журнал 3: День и час	○	-	-	-	Y	R	0
*SR240	Загружаемый журнал 3: Минута и секунда	○	-	-	-	Y	R	0
*SR241	Загружаемый журнал 4: Номер действия	○	-	-	-	Y	R	0
*SR242	Загружаемый журнал 4: Год и месяц	○	-	-	-	Y	R	0
*SR243	Загружаемый журнал 4: День и час	○	-	-	-	Y	R	0
*SR244	Загружаемый журнал 4: Минута и секунда	○	-	-	-	Y	R	0
*SR245	Загружаемый журнал 5: Номер действия	○	-	-	-	Y	R	0
*SR246	Загружаемый журнал 5: Год и месяц	○	-	-	-	Y	R	0
*SR247	Загружаемый журнал 5: День и час	○	-	-	-	Y	R	0
*SR248	Загружаемый журнал 5: Минута и секунда	○	-	-	-	Y	R	0
*SR249	Загружаемый журнал 6: Номер действия	○	-	-	-	Y	R	0
*SR250	Загружаемый журнал 6: Год и месяц	○	-	-	-	Y	R	0
*SR251	Загружаемый журнал 6: День и час	○	-	-	-	Y	R	0
*SR252	Загружаемый журнал 6: Минута и секунда	○	-	-	-	Y	R	0
*SR253	Загружаемый журнал 7: Номер действия	○	-	-	-	Y	R	0
*SR254	Загружаемый журнал 7: Год и месяц	○	-	-	-	Y	R	0
*SR255	Загружаемый журнал 7: День и час	○	-	-	-	Y	R	0
*SR256	Загружаемый журнал 7: Минута и секунда	○	-	-	-	Y	R	0
*SR257	Загружаемый журнал 8: Номер действия	○	-	-	-	Y	R	0
*SR258	Загружаемый журнал 8: Год и месяц	○	-	-	-	Y	R	0
*SR259	Загружаемый журнал 8: День и час	○	-	-	-	Y	R	0
*SR260	Загружаемый журнал 8: Минута и секунда	○	-	-	-	Y	R	0
*SR261	Загружаемый журнал 9: Номер действия	○	-	-	-	Y	R	0
*SR262	Загружаемый журнал 9: Год и месяц	○	-	-	-	Y	R	0
*SR263	Загружаемый журнал 9: День и час	○	-	-	-	Y	R	0
*SR264	Загружаемый журнал 9: Минута и секунда	○	-	-	-	Y	R	0
*SR265	Загружаемый журнал 10: Номер действия	○	-	-	-	Y	R	0
*SR266	Загружаемый журнал 10: Год и месяц	○	-	-	-	Y	R	0
*SR267	Загружаемый журнал 10: День и час	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR268	Загружаемый журнал 10: Минута и секунда	○	-	-	-	Y	R	0
*SR269	Загружаемый журнал 11: Номер действия	○	-	-	-	Y	R	0
*SR270	Загружаемый журнал 11: Год и месяц	○	-	-	-	Y	R	0
*SR271	Загружаемый журнал 11: День и час	○	-	-	-	Y	R	0
*SR272	Загружаемый журнал 11: Минута и секунда	○	-	-	-	Y	R	0
*SR273	Загружаемый журнал 12: Номер действия	○	-	-	-	Y	R	0
*SR274	Загружаемый журнал 12: Год и месяц	○	-	-	-	Y	R	0
*SR275	Загружаемый журнал 12: День и час	○	-	-	-	Y	R	0
*SR276	Загружаемый журнал 12: Минута и секунда	○	-	-	-	Y	R	0
*SR277	Загружаемый журнал 13: Номер действия	○	-	-	-	Y	R	0
*SR278	Загружаемый журнал 13: Год и месяц	○	-	-	-	Y	R	0
*SR279	Загружаемый журнал 13: День и час	○	-	-	-	Y	R	0
*SR280	Загружаемый журнал 13: Минута и секунда	○	-	-	-	Y	R	0
*SR281	Загружаемый журнал 14: Номер действия	○	-	-	-	Y	R	0
*SR282	Загружаемый журнал 14: Год и месяц	○	-	-	-	Y	R	0
*SR283	Загружаемый журнал 14: День и час	○	-	-	-	Y	R	0
*SR284	Загружаемый журнал 14: Минута и секунда	○	-	-	-	Y	R	0
*SR285	Загружаемый журнал 15: Номер действия	○	-	-	-	Y	R	0
*SR286	Загружаемый журнал 15: Год и месяц	○	-	-	-	Y	R	0
*SR287	Загружаемый журнал 15: День и час	○	-	-	-	Y	R	0
*SR288	Загружаемый журнал 15: Минута и секунда	○	-	-	-	Y	R	0
*SR289	Загружаемый журнал 16: Номер действия	○	-	-	-	Y	R	0
*SR290	Загружаемый журнал 16: Год и месяц	○	-	-	-	Y	R	0
*SR291	Загружаемый журнал 16: День и час	○	-	-	-	Y	R	0
*SR292	Загружаемый журнал 16: Минута и секунда	○	-	-	-	Y	R	0
*SR293	Загружаемый журнал 17: Номер действия	○	-	-	-	Y	R	0
*SR294	Загружаемый журнал 17: Год и месяц	○	-	-	-	Y	R	0
*SR295	Загружаемый журнал 17: День и час	○	-	-	-	Y	R	0
*SR296	Загружаемый журнал 17: Минута и секунда	○	-	-	-	Y	R	0
*SR297	Загружаемый журнал 18: Номер действия	○	-	-	-	Y	R	0
*SR298	Загружаемый журнал 18: Год и месяц	○	-	-	-	Y	R	0
*SR299	Загружаемый журнал 18: День и час	○	-	-	-	Y	R	0
*SR300	Загружаемый журнал 18: Минута и секунда	○	-	-	-	Y	R	0
*SR301	Загружаемый журнал 19: Номер действия	○	-	-	-	Y	R	0
*SR302	Загружаемый журнал 19: Год и месяц	○	-	-	-	Y	R	0
*SR303	Загружаемый журнал 19: День и час	○	-	-	-	Y	R	0
*SR304	Загружаемый журнал 19: Минута и секунда	○	-	-	-	Y	R	0
*SR305	Загружаемый журнал 20: Номер действия	○	-	-	-	Y	R	0
*SR306	Загружаемый журнал 20: Год и месяц	○	-	-	-	Y	R	0
*SR307	Загружаемый журнал 20: День и час	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR308	Загружаемый журнал 20: Минута и секунда	○	-	-	-	Y	R	0
*SR309	Число журналов состояния ПЛК (максимум 20)	○	-	-	-	Y	R	0
*SR310	Указатель журнала состояния ПЛК	○	-	-	-	Y	R	0
*SR311	Журнал состояния ПЛК 1: Номер действия	○	-	-	-	Y	R	0
*SR312	Журнал состояния ПЛК 1: Год и месяц	○	-	-	-	Y	R	0
*SR313	Журнал состояния ПЛК 1: День и час	○	-	-	-	Y	R	0
*SR314	Журнал состояния ПЛК 1: Минута и секунда	○	-	-	-	Y	R	0
*SR315	Журнал состояния ПЛК 2: Номер действия	○	-	-	-	Y	R	0
*SR316	Журнал состояния ПЛК 2: Год и месяц	○	-	-	-	Y	R	0
*SR317	Журнал состояния ПЛК 2: День и час	○	-	-	-	Y	R	0
*SR318	Журнал состояния ПЛК 2: Минута и секунда	○	-	-	-	Y	R	0
*SR319	Журнал состояния ПЛК 3: Номер действия	○	-	-	-	Y	R	0
*SR320	Журнал состояния ПЛК 3: Год и месяц	○	-	-	-	Y	R	0
*SR321	Журнал состояния ПЛК 3: День и час	○	-	-	-	Y	R	0
*SR322	Журнал состояния ПЛК 3: Минута и секунда	○	-	-	-	Y	R	0
*SR323	Журнал состояния ПЛК 4: Номер действия	○	-	-	-	Y	R	0
*SR324	Журнал состояния ПЛК 4: Год и месяц	○	-	-	-	Y	R	0
*SR325	Журнал состояния ПЛК 4: День и час	○	-	-	-	Y	R	0
*SR326	Журнал состояния ПЛК 4: Минута и секунда	○	-	-	-	Y	R	0
*SR327	Журнал состояния ПЛК 5: Номер действия	○	-	-	-	Y	R	0
*SR328	Журнал состояния ПЛК 5: Год и месяц	○	-	-	-	Y	R	0
*SR329	Журнал состояния ПЛК 5: День и час	○	-	-	-	Y	R	0
*SR330	Журнал состояния ПЛК 5: Минута и секунда	○	-	-	-	Y	R	0
*SR331	Журнал состояния ПЛК 6: Номер действия	○	-	-	-	Y	R	0
*SR332	Журнал состояния ПЛК 6: Год и месяц	○	-	-	-	Y	R	0
*SR333	Журнал состояния ПЛК 6: День и час	○	-	-	-	Y	R	0
*SR334	Журнал состояния ПЛК 6: Минута и секунда	○	-	-	-	Y	R	0
*SR335	Журнал состояния ПЛК 7: Номер действия	○	-	-	-	Y	R	0
*SR336	Журнал состояния ПЛК 7: Год и месяц	○	-	-	-	Y	R	0
*SR337	Журнал состояния ПЛК 7: День и час	○	-	-	-	Y	R	0
*SR338	Журнал состояния ПЛК 7: Минута и секунда	○	-	-	-	Y	R	0
*SR339	Журнал состояния ПЛК 8: Номер действия	○	-	-	-	Y	R	0
*SR340	Журнал состояния ПЛК 8: Год и месяц	○	-	-	-	Y	R	0
*SR341	Журнал состояния ПЛК 8: День и час	○	-	-	-	Y	R	0
*SR342	Журнал состояния ПЛК 8: Минута и секунда	○	-	-	-	Y	R	0
*SR343	Журнал состояния ПЛК 9: Номер действия	○	-	-	-	Y	R	0
*SR344	Журнал состояния ПЛК 9: Год и месяц	○	-	-	-	Y	R	0
*SR345	Журнал состояния ПЛК 9: День и час	○	-	-	-	Y	R	0
*SR346	Журнал состояния ПЛК 9: Минута и секунда	○	-	-	-	Y	R	0
*SR347	Журнал состояния ПЛК 10: Номер действия	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR348	Журнал состояния ПЛК 10: Год и месяц	○	-	-	-	Y	R	0
*SR349	Журнал состояния ПЛК 10: День и час	○	-	-	-	Y	R	0
*SR350	Журнал состояния ПЛК 10: Минута и секунда	○	-	-	-	Y	R	0
*SR351	Журнал состояния ПЛК 11: Номер действия	○	-	-	-	Y	R	0
*SR352	Журнал состояния ПЛК 11: Год и месяц	○	-	-	-	Y	R	0
*SR353	Журнал состояния ПЛК 11: День и час	○	-	-	-	Y	R	0
*SR354	Журнал состояния ПЛК 11: Минута и секунда	○	-	-	-	Y	R	0
*SR355	Журнал состояния ПЛК 12: Номер действия	○	-	-	-	Y	R	0
*SR356	Журнал состояния ПЛК 12: Год и месяц	○	-	-	-	Y	R	0
*SR357	Журнал состояния ПЛК 12: День и час	○	-	-	-	Y	R	0
*SR358	Журнал состояния ПЛК 12: Минута и секунда	○	-	-	-	Y	R	0
*SR359	Журнал состояния ПЛК 13: Номер действия	○	-	-	-	Y	R	0
*SR360	Журнал состояния ПЛК 13: Год и месяц	○	-	-	-	Y	R	0
*SR361	Журнал состояния ПЛК 13: День и час	○	-	-	-	Y	R	0
*SR362	Журнал состояния ПЛК 13: Минута и секунда	○	-	-	-	Y	R	0
*SR363	Журнал состояния ПЛК 14: Номер действия	○	-	-	-	Y	R	0
*SR364	Журнал состояния ПЛК 14: Год и месяц	○	-	-	-	Y	R	0
*SR365	Журнал состояния ПЛК 14: День и час	○	-	-	-	Y	R	0
*SR366	Журнал состояния ПЛК 14: Минута и секунда	○	-	-	-	Y	R	0
*SR367	Журнал состояния ПЛК 15: Номер действия	○	-	-	-	Y	R	0
*SR368	Журнал состояния ПЛК 15: Год и месяц	○	-	-	-	Y	R	0
*SR369	Журнал состояния ПЛК 15: День и час	○	-	-	-	Y	R	0
*SR370	Журнал состояния ПЛК 15: Минута и секунда	○	-	-	-	Y	R	0
*SR371	Журнал состояния ПЛК 16: Номер действия	○	-	-	-	Y	R	0
*SR372	Журнал состояния ПЛК 16: Год и месяц	○	-	-	-	Y	R	0
*SR373	Журнал состояния ПЛК 16: День и час	○	-	-	-	Y	R	0
*SR374	Журнал состояния ПЛК 16: Минута и секунда	○	-	-	-	Y	R	0
*SR375	Журнал состояния ПЛК 17: Номер действия	○	-	-	-	Y	R	0
*SR376	Журнал состояния ПЛК 17: Год и месяц	○	-	-	-	Y	R	0
*SR377	Журнал состояния ПЛК 17: День и час	○	-	-	-	Y	R	0
*SR378	Журнал состояния ПЛК 17: Минута и секунда	○	-	-	-	Y	R	0
*SR379	Журнал состояния ПЛК 18: Номер действия	○	-	-	-	Y	R	0
*SR380	Журнал состояния ПЛК 18: Год и месяц	○	-	-	-	Y	R	0
*SR381	Журнал состояния ПЛК 18: День и час	○	-	-	-	Y	R	0
*SR382	Журнал состояния ПЛК 18: Минута и секунда	○	-	-	-	Y	R	0
*SR383	Журнал состояния ПЛК 19: Номер действия	○	-	-	-	Y	R	0
*SR384	Журнал состояния ПЛК 19: Год и месяц	○	-	-	-	Y	R	0
*SR385	Журнал состояния ПЛК 19: День и час	○	-	-	-	Y	R	0
*SR386	Журнал состояния ПЛК 19: Минута и секунда	○	-	-	-	Y	R	0
*SR387	Журнал состояния ПЛК 20: Номер действия	○	-	-	-	Y	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR388	Журнал состояния ПЛК 20: Год и месяц	○	-	-	-	Y	R	0
*SR389	Журнал состояния ПЛК 20: День и час	○	-	-	-	Y	R	0
*SR390	Журнал состояния ПЛК 20: Минута и секунда	○	-	-	-	Y	R	0
*SR391	Значение года часов реального времени (RTC): 00~99	○	-	-	-	Y	R	0
*SR392	Значение месяца часов реального времени (RTC): 01~12	○	-	-	-	Y	R	1
*SR393	Значение дня часов реального времени (RTC): 1~31	○	-	-	-	Y	R	1
*SR394	Значение часа часов реального времени (RTC): 00~23	○	-	-	-	Y	R	0
*SR395	Значение минуты часов реального времени (RTC): 00~59	○	-	-	-	Y	R	0
*SR396	Значение секунды часов реального времени (RTC): 00~59	○	-	-	-	Y	R	0
*SR397	Значение дня недели часов реального времени (RTC): 1~7	○	-	-	-	Y	R	1
SR407	При работе ПЛК значение SR407 увеличивается на 1 каждую секунду. SR407 считает от 0 до 32767 и затем от -32768 до 0.	○	0	0	-	N	R/W	0
SR408	При работе ПЛК значение SR408 увеличивается на 1 каждый цикл сканирования. SR408 SR407 считает от 0 до 32767 и затем от -32768 до 0.	○	0	0	-	N	R/W	0
SR411	Текущее время скана сохраняется в SR411 и SR412, единица измерения - 100 мкс. Значение мс сохраняется в SR411 (Диапазон 0~65535). Значение мкс сохраняется в SR421 (Диапазон 0~900). Например, если 12 сохраняется в SR411 и 300 сохраняется в SR412, текущее время скана составляет 12,3 мс.	○	0	-	-	N	R	0
SR412	Текущее время скана сохраняется в SR411 и SR412, единица измерения - 100 мкс. Значение мс сохраняется в SR411 (Диапазон 0~65535). Значение мкс сохраняется в SR421 (Диапазон 0~900). Например, если 12 сохраняется в SR411 и 300 сохраняется в SR412, текущее время скана составляет 12,3 мс.	○	0	-	-	N	R	0
SR413	Минимальное время скана сохраняется в SR413 и SR414, единица измерения - 100 мкс. Значение мс сохраняется в SR413.	○	0	-	-	N	R	0
SR414	Минимальное время скана сохраняется в SR413 и SR414, единица измерения - 100 мкс. Значение мс сохраняется в SR413.	○	0	-	-	N	R	0
SR415	Максимальное время скана сохраняется в SR415 и SR416, единица измерения - 100 мкс. Значение мс сохраняется в SR415.	○	0	-	-	N	R	0
SR416	Максимальное время скана сохраняется в SR415 и SR416, единица измерения - 100 мкс. Значение мс сохраняется в SR415.	○	0	-	-	N	R	0
SR421	Время прерывания I601 (шаг: мс); по умолчанию 0, т.е. система следует настройкам в файле HWCONFIG	○	0	0	-	N	R/W	0
SR422	Время прерывания I602 (шаг: мс); по умолчанию 0, т.е. т.е. система следует настройкам в файле HWCONFIG.	○	0	0	-	N	R/W	0
SR423	Время прерывания I603 (шаг: мс); по умолчанию 0, т.е. система следует настройкам в файле HWCONFIG.	○	0	0	-	N	R/W	0
SR424	Время прерывания I604 (шаг: 0,1 мс); по умолчанию 0, т.е. система следует настройкам в файле HWCONFIG.	○	0	0	-	N	R/W	0
SR440	MAC адрес	○	-	-	-	Y	R	-
SR441	(пример: 12:34:56:78:9A:BC => SR440=16#1234, SR441=16#5678, SR442=16#9ABC)	○	-	-	-	Y	R	-

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR442		○	-	-	-	Y	R	-
SR443	Серия ПЛК	○	-	-	-	Y	R	-
SR444	Пример : AS324MTAW15500012	○	-	-	-	Y	R	-
SR445	AS → SR443 = 16#5341	○	-	-	-	Y	R	-
SR446	32 → SR444 = 16#3233	○	-	-	-	Y	R	-
SR447	4M → SR445 = 16#4D34	○	-	-	-	Y	R	-
SR448	TA → SR446 = 16#4154	○	-	-	-	Y	R	-
SR449	W1 → SR447 = 16#3157	○	-	-	-	Y	R	-
SR450	55 → SR448 = 16#3535	○	-	-	-	Y	R	-
SR451	00 → SR449 = 16#3030	○	-	-	-	Y	R	-
SR451	01 → SR450 = 16#3130	○	-	-	-	Y	R	-
SR451	2 → SR451 = 16#0032	○	-	-	-	Y	R	-
*SR453	Сохраняет код ошибки, возникшей при работе с картой памяти	○	-	-	-	Y	R	0
SR460	Количество выданных импульсов для Y0.0 ось 1 (Y0.0/Y0.1)	○	-	-	-	Y	R/W	0
SR461								
SR462	Выходной режим для оси 1 (Y0.0/Y0.1)	○	-	-	-	Y	R/W	0
SR463	Стартовая/конечная частота для оси 1 (Y0.0/Y0.1)	○	-	-	-	Y	R/W	200
SR464	Время разгона для оси 1 (Y0.0/Y0.1)	○	-	-	-	Y	R/W	200
SR465	Время торможения для оси 1 (Y0.0/Y0.1)	○	-	-	-	Y	R/W	200
SR466	Частота JOG для оси 1 (Y0.0/Y0.1)	○	-	-	-	Y	R/W	200
SR467	Текущий номер оси 1 (Y0.0/Y0.1) в таблице позиционирования	○	0	0	-	N	R	0
SR468	Числитель задания в линейных единицах, передаваемого на ось 1	○	-	-	-	H	R/W	0
SR469	Знаменатель задания в линейных единицах, передаваемого на ось 1	○	-	-	-	H	R/W	0
SR470	Позиция оси 1 (одинарное значение с плавающей точкой)	○	-	-	-	Y	R	0
SR471								
SR472	Заданная частота фиксированного наклона оси 1	○	-	-	-	Y	R	0
SR473								
SR474	Позиция Y0.1 (число импульсов)	○	-	-	-	Y	R/W	0
SR475								
SR476	Стартовая/конечная частота для Y0.1	○	-	-	-	Y	R/W	200
SR477	Время разгона/замедления для Y0.1.	○	-	-	-	Y	R/W	200
SR478	Импульс компенсации люфта оси 1 (Y0.0)	○	-	-	-	Y	R/W	0
SR479	Импульс компенсации люфта Y0.1	○	-	-	-	Y	R/W	0
SR480	Количество выданных импульсов для Y0.2 ось 2 (Y0.2/Y0.3)	○	-	-	-	Y	R/W	0
SR481								
SR482	Выходной режим для оси 2 (Y0.2/Y0.3)	○	-	-	-	Y	R/W	0
SR483	Стартовая/конечная частота для оси 2 (Y0.2/Y0.3)	○	-	-	-	Y	R/W	200

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR484	Время разгона для оси 2 (Y0.2/Y0.3)	○	-	-	-	Y	R/W	200
SR485	Время торможения для оси 2 (Y0.2/Y0.3)	○	-	-	-	Y	R/W	200
SR486	Частота JOG для оси 2 (Y0.2/Y0.3)	○	-	-	-	Y	R/W	200
SR487	Текущий номер оси 2 (Y0.2/Y0.3) в таблице позиционирования	○	0	0	-	N	R	0
SR488	Числитель задания в линейных единицах, передаваемого на ось 2	○	-	-	-	H	R/W	0
SR489	Знаменатель задания в линейных единицах, передаваемого на ось 2	○	-	-	-	H	R/W	0
SR490	Позиция оси 2 (одинарное значение с плавающей точкой)	○	-	-	-	Y	R	0
SR491								
SR492	Заданная частота фиксированного наклона оси 2	○	-	-	-	Y	R	0
SR493								
SR494	Позиция Y0.3 (число импульсов)	○	-	-	-	Y	R/W	0
SR495								
SR496	Стартовая/конечная частота для Y0.3	○	-	-	-	Y	R/W	200
SR497	Время разгона/замедления для Y0.3.	○	-	-	-	Y	R/W	200
SR498	Импульс компенсации люфта оси 2 (Y0.2)	○	-	-	-	Y	R/W	0
SR499	Импульс компенсации люфта Y0.3	○	-	-	-	Y	R/W	0
SR500	Количество выданных импульсов для Y0.4 ось 3 (Y0.4/Y0.5)	○	-	-	-	Y	R/W	0
SR501								
SR502	Выходной режим для оси 3 (Y0.4/Y0.5)	○	-	-	-	Y	R/W	0
SR503	Стартовая/конечная частота для оси 3 (Y0.4/Y0.5)	○	-	-	-	Y	R/W	200
SR504	Время разгона для оси 3 (Y0.4/Y0.5)	○	-	-	-	Y	R/W	200
SR505	Время торможения для оси 3 (Y0.4/Y0.5)	○	-	-	-	Y	R/W	200
SR506	Частота JOG для оси 3 (Y0.4/Y0.5)	○	-	-	-	Y	R/W	200
SR507	Текущий номер оси 3 (Y0.4/Y0.5) в таблице позиционирования	○	0	0	-	N	R	0
SR508	Числитель задания в линейных единицах, передаваемого на ось 3	○	-	-	-	H	R/W	0
SR509	Знаменатель задания в линейных единицах, передаваемого на ось 3	○	-	-	-	H	R/W	0
SR510	Позиция оси 3 (одинарное значение с плавающей точкой)	○	-	-	-	Y	R	0
SR511								
SR512	Заданная частота фиксированного наклона оси 3	○	-	-	-	Y	R	0
SR513								
SR514	Позиция Y0.5 (число импульсов)	○	-	-	-	Y	R/W	0
SR515								
SR516	Стартовая/конечная частота для Y0.5	○	-	-	-	Y	R/W	200
SR517	Время разгона/замедления для Y0.5	○	-	-	-	Y	R/W	200
SR518	Импульс компенсации люфта оси 3 (Y0.4)	○	-	-	-	Y	R/W	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR519	Импульс компенсации люфта Y0.5	○	-	-	-	Y	R/W	0
SR520	Количество выданных импульсов для Y0.6 ось 4 (Y0.6/Y0.7)	○	-	-	-	Y	R/W	0
SR521								
SR522	Выходной режим для оси 4 (Y0.6/Y0.7)	○	-	-	-	Y	R/W	0
SR523	Стартовая/конечная частота для оси 4 (Y0.6/Y0.7)	○	-	-	-	Y	R/W	200
SR524	Время разгона для оси 4 (Y0.6/Y0.7)	○	-	-	-	Y	R/W	200
SR525	Время торможения для оси 4 (Y0.6/Y0.7)	○	-	-	-	Y	R/W	200
SR526	Частота JOG для оси 4 (Y0.6/Y0.7)	○	-	-	-	Y	R/W	200
SR527	Текущий номер оси 4 (Y0.6/Y0.7) в таблице позиционирования	○	0	0	-	N	R	0
SR528	Числитель задания в линейных единицах, передаваемого на ось 4	○	-	-	-	H	R/W	0
SR529	Знаменатель задания в линейных единицах, передаваемого на ось 4	○	-	-	-	H	R/W	0
SR530	Позиция оси 4 (одинарное значение с плавающей точкой)	○	-	-	-	Y	R	0
SR531								
SR532	Заданная частота фиксированного наклона оси 4	○	-	-	-	Y	R	0
SR533								
SR534	Позиция Y0.7 (число импульсов)	○	-	-	-	Y	R/W	0
SR535								
SR536	Стартовая/конечная частота для Y0.7	○	-	-	-	Y	R/W	200
SR537	Время разгона/замедления для Y0.7	○	-	-	-	Y	R/W	200
SR538	Импульс компенсации люфта оси 4 (Y0.6)	○	-	-	-	Y	R/W	0
SR539	Импульс компенсации люфта Y0.7	○	-	-	-	Y	R/W	0
SR540	Количество выданных импульсов для Y0.8 ось 5 (Y0.8/Y0.9)	○	-	-	-	Y	R/W	0
SR541								
SR542	Выходной режим для оси 5 (Y0.8/Y0.9)	○	-	-	-	Y	R/W	0
SR543	Стартовая/конечная частота для оси 5 (Y0.8/Y0.9)	○	-	-	-	Y	R/W	200
SR544	Время разгона для оси 5 (Y0.8/Y0.9)	○	-	-	-	Y	R/W	200
SR545	Время торможения для оси 5 (Y0.8/Y0.9)	○	-	-	-	Y	R/W	200
SR546	Частота JOG для оси 5 (Y0.8/Y0.9)	○	-	-	-	Y	R/W	200
SR547	Текущий номер оси 5 (Y0.8/Y0.9) в таблице позиционирования	○	0	0	-	N	R	0
SR548	Числитель задания в линейных единицах, передаваемого на ось 5	○	-	-	-	H	R/W	0
SR549	Знаменатель задания в линейных единицах, передаваемого на ось 5	○	-	-	-	H	R/W	0
SR550	Позиция оси 5 (одинарное значение с плавающей точкой)	○	-	-	-	Y	R	0
SR551								
SR552	Заданная частота фиксированного наклона оси 5	○	-	-	-	Y	R	0
SR553								



SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR554	Позиция Y0.9	○	-	-	-	Y	R/W	0
SR555	(число импульсов)							
SR556	Стартовая/конечная частота для Y0.9	○	-	-	-	Y	R/W	200
SR557	Время разгона/замедления для Y0.9	○	-	-	-	Y	R/W	200
SR558	Импульс компенсации люфта оси 5 (Y0.8)	○	-	-	-	Y	R/W	0
SR559	Импульс компенсации люфта Y0.9	○	-	-	-	Y	R/W	0
SR560	Количество выданных импульсов для Y0.10 ось 6 (Y0.10/Y0.11)	○	-	-	-	Y	R/W	0
SR561								
SR562	Выходной режим для оси 6 (Y0.10/Y0.11)	○	-	-	-	Y	R/W	0
SR563	Стартовая/конечная частота для оси 6 (Y0.10/Y0.11)	○	-	-	-	Y	R/W	200
SR564	Время разгона для оси 6 (Y0.10/Y0.11)	○	-	-	-	Y	R/W	200
SR565	Время торможения для оси 6 (Y0.10/Y0.11)	○	-	-	-	Y	R/W	200
SR566	Частота JOG для оси 6 (Y0.10/Y0.11)	○	-	-	-	Y	R/W	200
SR567	Текущий номер оси 6 (Y0.10/Y0.11) в таблице позиционирования	○	0	0	-	N	R	0
SR568	Числитель задания в линейных единицах, передаваемого на ось 6	○	-	-	-	H	R/W	0
SR569	Знаменатель задания в линейных единицах, передаваемого на ось 6	○	-	-	-	H	R/W	0
SR570	Позиция оси 6 (одинарное значение с плавающей точкой)	○	-	-	-	Y	R	0
SR571								
SR572	Заданная частота фиксированного наклона оси 6	○	-	-	-	Y	R	0
SR573								
SR574	Позиция Y0.11	○	-	-	-	Y	R/W	0
SR575	(число импульсов)							
SR576	Стартовая/конечная частота для Y0.11	○	-	-	-	Y	R/W	200
SR577	Время разгона/замедления для Y0.11	○	-	-	-	Y	R/W	200
SR578	Импульс компенсации люфта оси 6 (Y0.10)	○	-	-	-	Y	R/W	0
SR579	Импульс компенсации люфта Y0.11	○	-	-	-	Y	R/W	0
*SR580	Верхний предел позиционирования для оси 1 (Y0.0/Y0.1) в ISPSOft (число импульсов)	○	-	-	-	H	R/W	0
*SR581								
*SR582	Нижний предел позиционирования для оси 1 (Y0.0/Y0.1) в ISPSOft (число импульсов)	○	-	-	-	H	R/W	0
*SR583								
*SR584	Верхний предел позиционирования для оси 2 (Y0.2/Y0.3) в ISPSOft (число импульсов)	○	-	-	-	H	R/W	0
*SR585								
*SR586	Нижний предел позиционирования для оси 2 (Y0.2/Y0.3) в ISPSOft (число импульсов)	○	-	-	-	H	R/W	0
*SR587								
*SR588	Верхний предел позиционирования для оси 3 (Y0.4/Y0.5) в ISPSOft (число импульсов)	○	-	-	-	H	R/W	0
*SR589								
*SR590	Нижний предел позиционирования для оси 3	○	-	-	-	H	R/W	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR591	(Y0.4/Y0.5) в ISPSOft (число импульсов)							
*SR592	Верхний предел позиционирования для оси 4	○	-	-	-	H	R/W	0
*SR593	(Y0.6/Y0.7) в ISPSOft (число импульсов)							
*SR594	Нижний предел позиционирования для оси 4	○	-	-	-	H	R/W	0
*SR595	(Y0.6/Y0.7) в ISPSOft (число импульсов)							
*SR596	Верхний предел позиционирования для оси 5	○	-	-	-	H	R/W	0
*SR597	(Y0.8/Y0.9) в ISPSOft (число импульсов)							
*SR598	Нижний предел позиционирования для оси 5	○	-	-	-	H	R/W	0
*SR599	(Y0.8/Y0.9) в ISPSOft (число импульсов)							
*SR600	Верхний предел позиционирования для оси 6	○	-	-	-	H	R/W	0
*SR601	(Y0.10/Y0.11) в ISPSOft (число импульсов)							
*SR602	Нижний предел позиционирования для оси 6	○	-	-	-	H	R/W	0
*SR603	(Y0.10/Y0.11) в ISPSOft (число импульсов)							
*SR604	S-образная кривая для оси 1 (Y0.0/Y0.1)	○	0	-	-	N	R/W	0
*SR605	S-образная кривая для оси 2 (Y0.2/Y0.3)	○	0	-	-	N	R/W	0
*SR606	S-образная кривая для оси 3 (Y0.4/Y0.5)	○	0	-	-	N	R/W	0
*SR607	S-образная кривая для оси 4 (Y0.6/Y0.7)	○	0	-	-	N	R/W	0
*SR608	S-образная кривая для оси 5 (Y0.8/Y0.9)	○	0	-	-	N	R/W	0
*SR609	S-образная кривая для оси 6 (Y0.10/Y0.11)	○	0	-	-	N	R/W	0
SR610	Текущая выходная скорость для оси 1 (Y0.0/Y0.1) (Гц)	○	0	0	0	N	R	0
SR611								
SR612	Текущая выходная скорость для оси 2 (Y0.2/Y0.3) (Гц)	○	0	0	0	N	R	0
SR613								
SR614	Текущая выходная скорость для оси 3 (Y0.4/Y0.5) (Гц)	○	0	0	0	N	R	0
SR615								
SR616	Текущая выходная скорость для оси 4 (Y0.6/Y0.7) (Гц)	○	0	0	0	N	R	0
SR617								
SR618	Текущая выходная скорость для оси 5 (Y0.8/Y0.9) (Гц)	○	0	0	0	N	R	0
SR619								
SR620	Текущая выходная скорость для оси 6 (Y0.10/Y0.11) (Гц)	○	0	0	0	N	R	0
SR621								
SR623	Условие внешнего прерывания: входы X0.0~X0.15, срабатывание по заднему фронту	○	FFFF	FFFF	-	N	R	FFFF
SR624	Условие внешнего прерывания входы X0.0~X0.15, срабатывание по переднему фронту	○	FFFF	FFFF	-	N	R	FFFF
SR625	Условие прерывания высокоскоростным компаратором I200~I233	○	FFFF	FFFF	-	N	R	FFFF
SR626	Условие прерывания высокоскоростным компаратором I240~I253	○	FFFF	FFFF	-	N	R	FFFF

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR627	Условие прерывания высокоскоростным компаратором I260~I267	○	FFFF	FFFF	-	N	R	FFFF
SR628	Условие прерывания по коммуникации I300~I307	○	FFFF	FFFF	-	N	R	FFFF
SR629	Условие прерывания по коммуникации I500~I505	○	FFFF	FFFF	-	N	R	FFFF
SR630	Условие прерывания по выходу I510~I519	○	FFFF	FFFF	-	N	R	FFFF
SR632	Условие прерывания по таймеру I601~I604	○	FFFF	FFFF	-	N	R	FFFF
SR633	Условие прерывания по внешнему модулю I400~I415	○	FFFF	FFFF	-	N	R	FFFF
SR634	Условие прерывания по внешнему модулю I416~I431	○	FFFF	FFFF	-	N	R	FFFF
SR640	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.0	○	0	-	-	N	R/W	0
SR641	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.1	○	0	-	-	N	R/W	0
SR642	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.2	○	0	-	-	N	R/W	0
SR643	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.3	○	0	-	-	N	R/W	0
SR644	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.4	○	0	-	-	N	R/W	0
SR645	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.5	○	0	-	-	N	R/W	0
SR646	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.6	○	0	-	-	N	R/W	0
SR647	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.7	○	0	-	-	N	R/W	0
SR648	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.8	○	0	-	-	N	R/W	0
SR649	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.9	○	0	-	-	N	R/W	0
SR650	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.10	○	0	-	-	N	R/W	0
SR651	Задание времени задержки вывода (0-20 мс, шаг: 1 мс) для работы с импульсным выходом Y0.11	○	0	-	-	N	R/W	0
SR658	Номер сервопривода Delta, где возникла ошибка коммуникации CANopen	○	0	-	-	N	R	0
SR659	Ошибка коммуникации Delta CANopen	○	0	-	-	N	R	0
SR661	Команда PR на сервопривод Delta 1 по коммуникации CANopen	○	0	-	-	N	R	0
SR662	Команда PR на сервопривод Delta 2 по коммуникации CANopen	○	0	-	-	N	R	0
SR663	Команда PR на сервопривод Delta 3 по коммуникации CANopen	○	0	-	-	N	R	0
SR664	Команда PR на сервопривод Delta 4 по коммуникации CANopen	○	0	-	-	N	R	0
SR665	Команда PR на сервопривод Delta 5 по коммуникации CANopen	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR666	Команда PR на сервопривод Delta 6 по коммуникации CANopen	○	0	-	-	N	R	0
SR667	Команда PR на сервопривод Delta 7 по коммуникации CANopen	○	0	-	-	N	R	0
SR668	Команда PR на сервопривод Delta 8 по коммуникации CANopen	○	0	-	-	N	R	0
SR671	Код ошибки коммуникации CANopen с сервоприводом Delta 1	○	0	-	-	N	R	0
SR672	Код ошибки коммуникации CANopen с сервоприводом Delta 2	○	0	-	-	N	R	0
SR673	Код ошибки коммуникации CANopen с сервоприводом Delta 3	○	0	-	-	N	R	0
SR674	Код ошибки коммуникации CANopen с сервоприводом Delta 4	○	0	-	-	N	R	0
SR675	Код ошибки коммуникации CANopen с сервоприводом Delta 5	○	0	-	-	N	R	0
SR676	Код ошибки коммуникации CANopen с сервоприводом Delta 6	○	0	-	-	N	R	0
SR677	Код ошибки коммуникации CANopen с сервоприводом Delta 7	○	0	-	-	N	R	0
SR678	Код ошибки коммуникации CANopen с сервоприводом Delta 8	○	0	-	-	N	R	0
SR681	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 1	○	0	-	-	N	R	0
SR682	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 2	○	0	-	-	N	R	0
SR683	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 3	○	0	-	-	N	R	0
SR684	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 4	○	0	-	-	N	R	0
SR685	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 5	○	0	-	-	N	R	0
SR686	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 6	○	0	-	-	N	R	0
SR687	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 7	○	0	-	-	N	R	0
SR688	Состояние дискретного выхода при коммуникации CANopen с сервоприводом Delta 8	○	0	-	-	N	R	0
SR691	Текущая позиция сервопривода Delta 1 по	○	0	-	-	N	R	0
SR692	коммуникации CANopen (32 бит)							
SR693	Текущая позиция сервопривода Delta 2 по	○	0	-	-	N	R	0
SR694	коммуникации CANopen (32 бит)							
SR695	Текущая позиция сервопривода Delta 3 по	○	0	-	-	N	R	0
SR696	коммуникации CANopen (32 бит)							
SR697	Текущая позиция сервопривода Delta 4 по	○	0	-	-	N	R	0
SR698	коммуникации CANopen (32 бит)							
SR699	Текущая позиция сервопривода Delta 5 по	○	0	-	-	N	R	0
SR700	коммуникации CANopen (32 бит)							
SR701	Текущая позиция сервопривода Delta 6 по	○	0	-	-	N	R	0
SR702	коммуникации CANopen (32 бит)							
SR703	Текущая позиция сервопривода Delta 7 по	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR704	коммуникации CANopen (32 бит)							
SR705	Текущая позиция сервопривода Delta 8 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR706								
SR711	Заданная позиция сервопривода Delta 1 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR712								
SR713	Заданная позиция сервопривода Delta 2 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR714								
SR715	Заданная позиция сервопривода Delta 3 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR716								
SR717	Заданная позиция сервопривода Delta 4 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR718								
SR719	Заданная позиция сервопривода Delta 5 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR720								
SR721	Заданная позиция сервопривода Delta 6 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR722								
SR723	Заданная позиция сервопривода Delta 7 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR724								
SR725	Заданная позиция сервопривода Delta 8 по коммуникации CANopen (32 бит)	○	0	-	-	N	R	0
SR726								
SR731	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID1	○	0	-	-	N	R	0
SR732	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID2	○	0	-	-	N	R	0
SR733	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID3	○	0	-	-	N	R	0
SR734	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID4	○	0	-	-	N	R	0
SR735	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID5	○	0	-	-	N	R	0
SR736	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID6	○	0	-	-	N	R	0
SR737	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID7	○	0	-	-	N	R	0
SR738	Текущее состояние дискретного входа (DI) при связи по CANopen с сервоприводом Delta ID8	○	0	-	-	N	R	0
SR741	Текущий момент при связи по CANopen с сервоприводом Delta ID1	○	0	-	-	N	R	0
SR742	Текущий момент при связи по CANopen с сервоприводом Delta ID2	○	0	-	-	N	R	0
SR743	Текущий момент при связи по по CANopen с сервоприводом Delta ID3	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR744	Текущий момент при связи по по CANopen с сервоприводом Delta ID4	○	0	-	-	N	R	0
SR745	Текущий момент при связи по по CANopen с сервоприводом Delta ID5	○	0	-	-	N	R	0
SR746	Текущий момент при связи по по CANopen с сервоприводом Delta ID6	○	0	-	-	N	R	0
SR747	Текущий момент при связи по по CANopen с сервоприводом Delta ID7	○	0	-	-	N	R	0
SR748	Текущий момент при связи по по CANopen с сервоприводом Delta ID8	○	0	-	-	N	R	0
SR751	Текущее состояние связи по CANopen с ПЧ Delta ID21	○	0	-	-	N	R	0
SR752	Текущее состояние связи по CANopen с ПЧ Delta ID22	○	0	-	-	N	R	0
SR753	Текущее состояние связи по CANopen с ПЧ Delta ID23	○	0	-	-	N	R	0
SR754	Текущее состояние связи по CANopen с ПЧ Delta ID24	○	0	-	-	N	R	0
SR755	Текущее состояние связи по CANopen с ПЧ Delta ID25	○	0	-	-	N	R	0
SR756	Текущее состояние связи по CANopen с ПЧ Delta ID26	○	0	-	-	N	R	0
SR757	Текущее состояние связи по CANopen с ПЧ Delta ID27	○	0	-	-	N	R	0
SR758	Текущее состояние связи по CANopen с ПЧ Delta ID28	○	0	-	-	N	R	0
SR761	Текущая скорость ПЧ Slave ID21 при связи по CANopen	○	0	-	-	N	R	0
SR762	Текущая скорость ПЧ Slave ID22 при связи по CANopen	○	0	-	-	N	R	0
SR763	Текущая скорость ПЧ Slave ID23 при связи по CANopen	○	0	-	-	N	R	0
SR764	Текущая скорость ПЧ Slave ID24 при связи по CANopen	○	0	-	-	N	R	0
SR765	Текущая скорость ПЧ Slave ID25 при связи по CANopen	○	0	-	-	N	R	0
SR766	Текущая скорость ПЧ Slave ID26 при связи по CANopen	○	0	-	-	N	R	0
SR767	Текущая скорость ПЧ Slave ID27 при связи по CANopen	○	0	-	-	N	R	0
SR768	Текущая скорость ПЧ Slave ID28 при связи по CANopen	○	0	-	-	N	R	0
SR771	Текущий момент ПЧ Delta ID21 при связи по CANopen	○	0	-	-	N	R	0
SR772	Текущий момент ПЧ Delta ID22 при связи по CANopen	○	0	-	-	N	R	0
SR773	Текущий момент ПЧ Delta ID23 при связи по CANopen	○	0	-	-	N	R	0
SR774	Текущий момент ПЧ Delta ID24 при связи по CANopen	○	0	-	-	N	R	0
SR775	Текущий момент ПЧ Delta ID25 при связи по CANopen	○	0	-	-	N	R	0
SR776	Текущий момент ПЧ Delta ID26 при связи по CANopen	○	0	-	-	N	R	0
SR777	Текущий момент ПЧ Delta ID27 при связи по CANopen	○	0	-	-	N	R	0
SR778	Текущий момент ПЧ Delta ID28 при связи по CANopen	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR781	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID21	○	0	-	-	N	R	0
SR782	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID22	○	0	-	-	N	R	0
SR783	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID23	○	0	-	-	N	R	0
SR784	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID24	○	0	-	-	N	R	0
SR785	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID25	○	0	-	-	N	R	0
SR786	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID26	○	0	-	-	N	R	0
SR787	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID27	○	0	-	-	N	R	0
SR788	Текущее состояние дискретного входа (DI) при связи по CANopen с ПЧ Delta ID28	○	0	-	-	N	R	0
SR820	Код состояния master/slave CANopen DS301	○	-	-	-	Y	R	0
SR821	Код версии CANopen DS301	○	-	-	-	Y	R	-
SR822	Скорость передачи данных CANopen (шаг: 1 кбит/с)	○	-	-	-	H	R	125
SR825	Код состояния master для CANopen DS301	○	-1	-	-	N	R	-1
SR826	Состояние slave ID 1~16 для CANopen DS301	○	-1	-	-	N	R	-1
SR827	Состояние slave ID 17~32 для CANopen DS301	○	-1	-	-	N	R	-1
SR828	Состояние slave ID 33~48 для CANopen DS301	○	-1	-	-	N	R	-1
SR829	Состояние slave ID 49~64 для CANopen DS301	○	-1	-	-	N	R	-1
SR830	Состояние slave ID 1 для CANopen DS301	○	-1	-	-	N	R	-1
SR831	Состояние slave ID 2 для CANopen DS301	○	-1	-	-	N	R	-1
SR832	Состояние slave ID 3 для CANopen DS301	○	-1	-	-	N	R	-1
SR833	Состояние slave ID 4 для CANopen DS301	○	-1	-	-	N	R	-1
SR834	Состояние slave ID 5 для CANopen DS301	○	-1	-	-	N	R	-1
SR835	Состояние slave ID 6 для CANopen DS301	○	-1	-	-	N	R	-1
SR836	Состояние slave ID 7 для CANopen DS301	○	-1	-	-	N	R	-1
SR837	Состояние slave ID 8 для CANopen DS301	○	-1	-	-	N	R	-1
SR838	Состояние slave ID 9 для CANopen DS301	○	-1	-	-	N	R	-1
SR839	Состояние slave ID 10 для CANopen DS301	○	-1	-	-	N	R	-1
SR840	Состояние slave ID 11 для CANopen DS301	○	-1	-	-	N	R	-1
SR841	Состояние slave ID 12 для CANopen DS301	○	-1	-	-	N	R	-1
SR842	Состояние slave ID 13 для CANopen DS301	○	-1	-	-	N	R	-1
SR843	Состояние slave ID 14 для CANopen DS301	○	-1	-	-	N	R	-1
SR844	Состояние slave ID 15 для CANopen DS301	○	-1	-	-	N	R	-1
SR845	Состояние slave ID 16 для CANopen DS301	○	-1	-	-	N	R	-1
SR846	Состояние slave ID 17 для CANopen DS301	○	-1	-	-	N	R	-1
SR847	Состояние slave ID 18 для CANopen DS301	○	-1	-	-	N	R	-1

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR848	Состояние slave ID 19 для CANopen DS301	○	-1	-	-	N	R	-1
SR849	Состояние slave ID 20 для CANopen DS301	○	-1	-	-	N	R	-1
SR850	Состояние slave ID 21 для CANopen DS301	○	-1	-	-	N	R	-1
SR851	Состояние slave ID 22 для CANopen DS301	○	-1	-	-	N	R	-1
SR852	Состояние slave ID 23 для CANopen DS301	○	-1	-	-	N	R	-1
SR853	Состояние slave ID 24 для CANopen DS301	○	-1	-	-	N	R	-1
SR854	Состояние slave ID 25 для CANopen DS301	○	-1	-	-	N	R	-1
SR855	Состояние slave ID 26 для CANopen DS301	○	-1	-	-	N	R	-1
SR856	Состояние slave ID 27 для CANopen DS301	○	-1	-	-	N	R	-1
SR857	Состояние slave ID 28 для CANopen DS301	○	-1	-	-	N	R	-1
SR858	Состояние slave ID 29 для CANopen DS301	○	-1	-	-	N	R	-1
SR859	Состояние slave ID 30 для CANopen DS301	○	-1	-	-	N	R	-1
SR860	Состояние slave ID 31 для CANopen DS301	○	-1	-	-	N	R	-1
SR861	Состояние slave ID 32 для CANopen DS301	○	-1	-	-	N	R	-1
SR862	Состояние slave ID 33 для CANopen DS301	○	-1	-	-	N	R	-1
SR863	Состояние slave ID 34 для CANopen DS301	○	-1	-	-	N	R	-1
SR864	Состояние slave ID 35 для CANopen DS301	○	-1	-	-	N	R	-1
SR865	Состояние slave ID 36 для CANopen DS301	○	-1	-	-	N	R	-1
SR866	Состояние slave ID 37 для CANopen DS301	○	-1	-	-	N	R	-1
SR867	Состояние slave ID 38 для CANopen DS301	○	-1	-	-	N	R	-1
SR868	Состояние slave ID 39 для CANopen DS301	○	-1	-	-	N	R	-1
SR869	Состояние slave ID 40 для CANopen DS301	○	-1	-	-	N	R	-1
SR870	Состояние slave ID 41 для CANopen DS301	○	-1	-	-	N	R	-1
SR871	Состояние slave ID 42 для CANopen DS301	○	-1	-	-	N	R	-1
SR872	Состояние slave ID 43 для CANopen DS301	○	-1	-	-	N	R	-1
SR873	Состояние slave ID 44 для CANopen DS301	○	-1	-	-	N	R	-1
SR874	Состояние slave ID 45 для CANopen DS301	○	-1	-	-	N	R	-1
SR875	Состояние slave ID 46 для CANopen DS301	○	-1	-	-	N	R	-1
SR876	Состояние slave ID 47 для CANopen DS301	○	-1	-	-	N	R	-1
SR877	Состояние slave ID 48 для CANopen DS301	○	-1	-	-	N	R	-1
SR878	Состояние slave ID 49 для CANopen DS301	○	-1	-	-	N	R	-1
SR879	Состояние slave ID 50 для CANopen DS301	○	-1	-	-	N	R	-1
SR880	Состояние slave ID 51 для CANopen DS301	○	-1	-	-	N	R	-1
SR881	Состояние slave ID 52 для CANopen DS301	○	-1	-	-	N	R	-1
SR882	Состояние slave ID 53 для CANopen DS301	○	-1	-	-	N	R	-1
SR883	Состояние slave ID 54 для CANopen DS301	○	-1	-	-	N	R	-1
SR884	Состояние slave ID 55 для CANopen DS301	○	-1	-	-	N	R	-1
SR885	Состояние slave ID 56 для CANopen DS301	○	-1	-	-	N	R	-1
SR886	Состояние slave ID 57 для CANopen DS301	○	-1	-	-	N	R	-1
SR887	Состояние slave ID 58 для CANopen DS301	○	-1	-	-	N	R	-1



SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR888	Состояние slave ID 59 для CANopen DS301	○	-1	-	-	N	R	-1
SR889	Состояние slave ID 60 для CANopen DS301	○	-1	-	-	N	R	-1
SR890	Состояние slave ID 61 для CANopen DS301	○	-1	-	-	N	R	-1
SR891	Состояние slave ID 62 для CANopen DS301	○	-1	-	-	N	R	-1
SR892	Состояние slave ID 63 для CANopen DS301	○	-1	-	-	N	R	-1
SR893	Состояние slave ID 64 для CANopen DS301	○	-1	-	-	N	R	-1
SR900	Число выборок в регистраторе данных (32-bit)	○	0	-	-	N	R	0
SR901								
SR902	Код работы регистратора данных и карты памяти (совместно с SM456), например, H5AA5: запись выборки данных из регистратора на карту памяти	○	0	-	-	N	R/W	0
SR1000	IP адрес Ethernet (32 бит)	○	-	-	-	H	R/W	0
SR1001								
SR1002	Маска подсети Ethernet (32 бит)	○	-	-	-	H	R/W	0
SR1003								
SR1004	Адрес шлюза Ethernet (32 бит)	○	-	-	-	H	R/W	0
SR1005								
SR1006	Время, в течение которого TCP-соединение было постоянным	○	-	-	-	H	R/W	30
SR1007	Скорость передачи данных Ethernet	○	0	-	-	N	R	0
SR1009	Количество соединений TCP	○	0	-	-	N	R	0
SR1010	Заданное время для повторной отправки через TCP-соединение	○	-	-	-	N	R/W	20
SR1011	Номер подключения MODBUS/TCP Server	○	0	-	-	N	R	0
SR1012	Номер подключения MODBUS/TCP Client	○	0	-	-	N	R	0
SR1013	Номер подключения EtherNet/IP Adapter	○	0	-	-	N	R	0
SR1014	Номер подключения EtherNet/IP Scanner	○	0	-	-	N	R	0
SR1020	Состояние соединения EtherNet/IP 1	○	0	-	-	N	R	0
SR1021	Состояние соединения EtherNet/IP 2	○	0	-	-	N	R	0
SR1022	Состояние соединения EtherNet/IP 3	○	0	-	-	N	R	0
SR1023	Состояние соединения EtherNet/IP 4	○	0	-	-	N	R	0
SR1024	Состояние соединения EtherNet/IP 5	○	0	-	-	N	R	0
SR1025	Состояние соединения EtherNet/IP 6	○	0	-	-	N	R	0
SR1026	Состояние соединения EtherNet/IP 7	○	0	-	-	N	R	0
SR1027	Состояние соединения EtherNet/IP 8	○	0	-	-	N	R	0
SR1028	Состояние соединения EtherNet/IP 9	○	0	-	-	N	R	0
SR1029	Состояние соединения EtherNet/IP 10	○	0	-	-	N	R	0
SR1030	Состояние соединения EtherNet/IP 11	○	0	-	-	N	R	0
SR1031	Состояние соединения EtherNet/IP 12	○	0	-	-	N	R	0
SR1032	Состояние соединения EtherNet/IP 13	○	0	-	-	N	R	0
SR1033	Состояние соединения EtherNet/IP 14	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR1034	Состояние соединения EtherNet/IP 15	○	0	-	-	N	R	0
SR1035	Состояние соединения EtherNet/IP 16	○	0	-	-	N	R	0
SR1036	Состояние соединения EtherNet/IP 17	○	0	-	-	N	R	0
SR1037	Состояние соединения EtherNet/IP 18	○	0	-	-	N	R	0
SR1038	Состояние соединения EtherNet/IP 19	○	0	-	-	N	R	0
SR1039	Состояние соединения EtherNet/IP 20	○	0	-	-	N	R	0
SR1040	Состояние соединения EtherNet/IP 21	○	0	-	-	N	R	0
SR1041	Состояние соединения EtherNet/IP 22	○	0	-	-	N	R	0
SR1042	Состояние соединения EtherNet/IP 23	○	0	-	-	N	R	0
SR1043	Состояние соединения EtherNet/IP 24	○	0	-	-	N	R	0
SR1044	Состояние соединения EtherNet/IP 25	○	0	-	-	N	R	0
SR1045	Состояние соединения EtherNet/IP 26	○	0	-	-	N	R	0
SR1046	Состояние соединения EtherNet/IP 27	○	0	-	-	N	R	0
SR1047	Состояние соединения EtherNet/IP 28	○	0	-	-	N	R	0
SR1048	Состояние соединения EtherNet/IP 29	○	0	-	-	N	R	0
SR1049	Состояние соединения EtherNet/IP 30	○	0	-	-	N	R	0
SR1050	Состояние соединения EtherNet/IP 31	○	0	-	-	N	R	0
SR1051	Состояние соединения EtherNet/IP 32	○	0	-	-	N	R	0
SR1052	Время обновления для соединения EtherNet/IP 1	○	0	-	-	N	R	0
SR1053	Время обновления для соединения EtherNet/IP 2	○	0	-	-	N	R	0
SR1054	Время обновления для соединения EtherNet/IP 3	○	0	-	-	N	R	0
SR1055	Время обновления для соединения EtherNet/IP 4	○	0	-	-	N	R	0
SR1056	Время обновления для соединения EtherNet/IP 5	○	0	-	-	N	R	0
SR1057	Время обновления для соединения EtherNet/IP 6	○	0	-	-	N	R	0
SR1058	Время обновления для соединения EtherNet/IP 7	○	0	-	-	N	R	0
SR1059	Время обновления для соединения EtherNet/IP 8	○	0	-	-	N	R	0
SR1060	Время обновления для соединения EtherNet/IP 9	○	0	-	-	N	R	0
SR1061	Время обновления для соединения EtherNet/IP 10	○	0	-	-	N	R	0
SR1062	Время обновления для соединения EtherNet/IP 11	○	0	-	-	N	R	0
SR1063	Время обновления для соединения EtherNet/IP 12	○	0	-	-	N	R	0
SR1064	Время обновления для соединения EtherNet/IP с 13	○	0	-	-	N	R	0
SR1065	Время обновления для соединения EtherNet/IP 14	○	0	-	-	N	R	0
SR1066	Время обновления для соединения EtherNet/IP 15	○	0	-	-	N	R	0
SR1067	Время обновления для соединения EtherNet/IP 16	○	0	-	-	N	R	0
SR1068	Время обновления для соединения EtherNet/IP 17	○	0	-	-	N	R	0
SR1069	Время обновления для соединения EtherNet/IP 18	○	0	-	-	N	R	0
SR1070	Время обновления для соединения EtherNet/IP 19	○	0	-	-	N	R	0
SR1071	Время обновления для соединения EtherNet/IP 20	○	0	-	-	N	R	0
SR1072	Время обновления для соединения EtherNet/IP 21	○	0	-	-	N	R	0
SR1073	Время обновления для соединения EtherNet/IP 22	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR1074	Время обновления для соединения EtherNet/IP 23	○	0	-	-	N	R	0
SR1075	Время обновления для соединения EtherNet/IP 24	○	0	-	-	N	R	0
SR1076	Время обновления для соединения EtherNet/IP 25	○	0	-	-	N	R	0
SR1077	Время обновления для соединения EtherNet/IP 26	○	0	-	-	N	R	0
SR1078	Время обновления для соединения EtherNet/IP 27	○	0	-	-	N	R	0
SR1079	Время обновления для соединения EtherNet/IP 28	○	0	-	-	N	R	0
SR1080	Время обновления для соединения EtherNet/IP 29	○	0	-	-	N	R	0
SR1081	Время обновления для соединения EtherNet/IP 30	○	0	-	-	N	R	0
SR1082	Время обновления для соединения EtherNet/IP 31	○	0	-	-	N	R	0
SR1083	Время обновления для соединения EtherNet/IP 32	○	0	-	-	N	R	0
SR1100	Значение счетчика входных пакетов (32 бит)	○	0	-	-	N	R	0
SR1101								
SR1102	Значение счетчика входных байтов (32 бит)	○	0	-	-	N	R	0
SR1103								
SR1104	Значение счетчика выходных пакетов (32 бит)	○	0	-	-	N	R	0
SR1105								
SR1106	Значение счетчика выходных байтов (32 бит)	○	0	-	-	N	R	0
SR1107								
SR1116	Счетчик Email	○	0	-	-	N	R	0
SR1117	Ошибка счетчика Email	○	0	-	-	N	R	0
*SR1120	Текущее время соединения для обмена данными через Ethernet 1	○	0	-	-	N	R	0
*SR1121	Текущее время соединения для обмена данными через Ethernet 2	○	0	-	-	N	R	0
*SR1122	Текущее время соединения для обмена данными через Ethernet 3	○	0	-	-	N	R	0
*SR1123	Текущее время соединения для обмена данными через Ethernet 4	○	0	-	-	N	R	0
*SR1124	Текущее время соединения для обмена данными через Ethernet 5	○	0	-	-	N	R	0
*SR1125	Текущее время соединения для обмена данными через Ethernet 6	○	0	-	-	N	R	0
*SR1126	Текущее время соединения для обмена данными через Ethernet 7	○	0	-	-	N	R	0
*SR1127	Текущее время соединения для обмена данными через Ethernet 8	○	0	-	-	N	R	0
*SR1128	Текущее время соединения для обмена данными через Ethernet 9	○	0	-	-	N	R	0
*SR1129	Текущее время соединения для обмена данными через Ethernet 10	○	0	-	-	N	R	0
*SR1130	Текущее время соединения для обмена данными через Ethernet 11	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR1131	Текущее время соединения для обмена данными через Ethernet 12	○	0	-	-	N	R	0
*SR1132	Текущее время соединения для обмена данными через Ethernet 13	○	0	-	-	N	R	0
*SR1133	Текущее время соединения для обмена данными через Ethernet 14	○	0	-	-	N	R	0
*SR1134	Текущее время соединения для обмена данными через Ethernet 15	○	0	-	-	N	R	0
*SR1135	Текущее время соединения для обмена данными через Ethernet 16	○	0	-	-	N	R	0
*SR1136	Текущее время соединения для обмена данными через Ethernet 17	○	0	-	-	N	R	0
*SR1137	Текущее время соединения для обмена данными через Ethernet 18	○	0	-	-	N	R	0
*SR1138	Текущее время соединения для обмена данными через Ethernet 19	○	0	-	-	N	R	0
*SR1139	Текущее время соединения для обмена данными через Ethernet 20	○	0	-	-	N	R	0
*SR1140	Текущее время соединения для обмена данными через Ethernet 21	○	0	-	-	N	R	0
*SR1141	Текущее время соединения для обмена данными через Ethernet 22	○	0	-	-	N	R	0
*SR1142	Текущее время соединения для обмена данными через Ethernet 23	○	0	-	-	N	R	0
*SR1143	Текущее время соединения для обмена данными через Ethernet 24	○	0	-	-	N	R	0
*SR1144	Текущее время соединения для обмена данными через Ethernet 25	○	0	-	-	N	R	0
*SR1145	Текущее время соединения для обмена данными через Ethernet 26	○	0	-	-	N	R	0
*SR1146	Текущее время соединения для обмена данными через Ethernet 27	○	0	-	-	N	R	0
*SR1147	Текущее время соединения для обмена данными через Ethernet 28	○	0	-	-	N	R	0
*SR1148	Текущее время соединения для обмена данными через Ethernet 29	○	0	-	-	N	R	0
*SR1149	Текущее время соединения для обмена данными через Ethernet 30	○	0	-	-	N	R	0
*SR1150	Текущее время соединения для обмена данными через Ethernet 31	○	0	-	-	N	R	0
*SR1151	Текущее время соединения для обмена данными через Ethernet 32	○	0	-	-	N	R	0
*SR1152	Код ошибки обмена данными через Ethernet соединение 1	○	0	-	-	N	R	0
*SR1153	Код ошибки обмена данными через Ethernet соединение 2	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR1154	Код ошибки обмена данными через Ethernet соединение 3	○	0	-	-	N	R	0
*SR1155	Код ошибки обмена данными через Ethernet соединение 4	○	0	-	-	N	R	0
*SR1156	Код ошибки обмена данными через Ethernet соединение 5	○	0	-	-	N	R	0
*SR1157	Код ошибки обмена данными через Ethernet соединение 6	○	0	-	-	N	R	0
*SR1158	Код ошибки обмена данными через Ethernet соединение 7	○	0	-	-	N	R	0
*SR1159	Код ошибки обмена данными через Ethernet соединение 8	○	0	-	-	N	R	0
*SR1160	Код ошибки обмена данными через Ethernet соединение 9	○	0	-	-	N	R	0
*SR1161	Код ошибки обмена данными через Ethernet соединение 10	○	0	-	-	N	R	0
*SR1162	Код ошибки обмена данными через Ethernet соединение 11	○	0	-	-	N	R	0
*SR1163	Код ошибки обмена данными через Ethernet соединение 12	○	0	-	-	N	R	0
*SR1164	Код ошибки обмена данными через Ethernet соединение 13	○	0	-	-	N	R	0
*SR1165	Код ошибки обмена данными через Ethernet соединение 14	○	0	-	-	N	R	0
*SR1166	Код ошибки обмена данными через Ethernet соединение 15	○	0	-	-	N	R	0
*SR1167	Код ошибки обмена данными через Ethernet соединение 16	○	0	-	-	N	R	0
*SR1168	Код ошибки обмена данными через Ethernet соединение 17	○	0	-	-	N	R	0
*SR1169	Код ошибки обмена данными через Ethernet соединение 18	○	0	-	-	N	R	0
*SR1170	Код ошибки обмена данными через Ethernet соединение 19	○	0	-	-	N	R	0
*SR1171	Код ошибки обмена данными через Ethernet соединение 20	○	0	-	-	N	R	0
*SR1172	Код ошибки обмена данными через Ethernet соединение 21	○	0	-	-	N	R	0
*SR1173	Код ошибки обмена данными через Ethernet соединение 22	○	0	-	-	N	R	0
*SR1174	Код ошибки обмена данными через Ethernet соединение 23	○	0	-	-	N	R	0
*SR1175	Код ошибки обмена данными через Ethernet соединение 24	○	0	-	-	N	R	0
*SR1176	Код ошибки обмена данными через Ethernet соединение 25	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR1177	Код ошибки обмена данными через Ethernet соединение 26	○	0	-	-	N	R	0
*SR1178	Код ошибки обмена данными через Ethernet соединение 27	○	0	-	-	N	R	0
*SR1179	Код ошибки обмена данными через Ethernet соединение 28	○	0	-	-	N	R	0
*SR1180	Код ошибки обмена данными через Ethernet соединение 29	○	0	-	-	N	R	0
*SR1181	Код ошибки обмена данными через Ethernet соединение 30	○	0	-	-	N	R	0
*SR1182	Код ошибки обмена данными через Ethernet соединение 31	○	0	-	-	N	R	0
*SR1183	Код ошибки обмена данными через Ethernet соединение 32	○	0	-	-	N	R	0
SR1318	Счетчик входных сокетов	○	0	-	-	N	R	0
SR1319	Счетчик выходных сокетов	○	0	-	-	N	R	0
SR1320	Ошибка счетчика сокетов	○	0	-	-	N	R	0
*SR1335	Текущее время цикла соединений 1~32 для обмена данными через COM1	○	0	-	-	N	R	0
*SR1336	Номер соединения, которое в настоящее время выполняет циклический обмен данными через COM1	○	0	-	-	N	R	0
*SR1340	Код ошибки обмена данными через соединение COM1 1	○	0	-	-	N	R	0
*SR1341	Код ошибки обмена данными через соединение COM1 2	○	0	-	-	N	R	0
*SR1342	Код ошибки обмена данными через соединение COM1 3	○	0	-	-	N	R	0
*SR1343	Код ошибки обмена данными через соединение COM1 4	○	0	-	-	N	R	0
*SR1344	Код ошибки обмена данными через соединение COM1 5	○	0	-	-	N	R	0
*SR1345	Код ошибки обмена данными через соединение COM1 6	○	0	-	-	N	R	0
*SR1346	Код ошибки обмена данными через соединение COM1 7	○	0	-	-	N	R	0
*SR1347	Код ошибки обмена данными через соединение COM1 8	○	0	-	-	N	R	0
*SR1348	Код ошибки обмена данными через соединение COM1 9	○	0	-	-	N	R	0
*SR1349	Код ошибки обмена данными через соединение COM1 10	○	0	-	-	N	R	0
*SR1350	Код ошибки обмена данными через соединение COM1 11	○	0	-	-	N	R	0
*SR1351	Код ошибки обмена данными через соединение COM1 12	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR1352	Код ошибки обмена данными через соединение COM1 13	○	0	-	-	N	R	0
*SR1353	Код ошибки обмена данными через соединение COM1 14	○	0	-	-	N	R	0
*SR1354	Код ошибки обмена данными через соединение COM1 15	○	0	-	-	N	R	0
*SR1355	Код ошибки обмена данными через соединение COM1 16	○	0	-	-	N	R	0
*SR1356	Код ошибки обмена данными через соединение COM1 17	○	0	-	-	N	R	0
*SR1357	Код ошибки обмена данными через соединение COM1 18	○	0	-	-	N	R	0
*SR1358	Код ошибки обмена данными через соединение COM1 19	○	0	-	-	N	R	0
*SR1359	Код ошибки обмена данными через соединение COM1 20	○	0	-	-	N	R	0
*SR1360	Код ошибки обмена данными через соединение COM1 21	○	0	-	-	N	R	0
*SR1361	Код ошибки обмена данными через соединение COM1 22	○	0	-	-	N	R	0
*SR1362	Код ошибки обмена данными через соединение COM1 23	○	0	-	-	N	R	0
*SR1363	Код ошибки обмена данными через соединение COM1 24	○	0	-	-	N	R	0
*SR1364	Код ошибки обмена данными через соединение COM1 25	○	0	-	-	N	R	0
*SR1365	Код ошибки обмена данными через соединение COM1 26	○	0	-	-	N	R	0
*SR1366	Код ошибки обмена данными через соединение COM1 27	○	0	-	-	N	R	0
*SR1367	Код ошибки обмена данными через соединение COM1 28	○	0	-	-	N	R	0
*SR1368	Код ошибки обмена данными через соединение COM1 29	○	0	-	-	N	R	0
*SR1369	Код ошибки обмена данными через соединение COM1 30	○	0	-	-	N	R	0
*SR1370	Код ошибки обмена данными через соединение COM1 31	○	0	-	-	N	R	0
*SR1371	Код ошибки обмена данными через соединение COM1 32	○	0	-	-	N	R	0
*SR1375	Текущее время цикла соединений 1~32 для обмена данными через COM2	○	0	-	-	N	R	0
*SR1376	Номер соединения, которое в настоящее время выполняет циклический обмен данными через COM2	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
*SR1380	Код ошибки обмена данными через соединение COM2 1	○	0	-	-	N	R	0
SR1381	Код ошибки обмена данными через соединение COM2 2	○	0	-	-	N	R	0
SR1382	Код ошибки обмена данными через соединение COM2 3	○	0	-	-	N	R	0
SR1383	Код ошибки обмена данными через соединение COM2 4	○	0	-	-	N	R	0
SR1384	Код ошибки обмена данными через соединение COM2 5	○	0	-	-	N	R	0
SR1385	Код ошибки обмена данными через соединение COM2 6	○	0	-	-	N	R	0
SR1386	Код ошибки обмена данными через соединение COM2 7	○	0	-	-	N	R	0
SR1387	Код ошибки обмена данными через соединение COM2 8	○	0	-	-	N	R	0
SR1388	Код ошибки обмена данными через соединение COM2 9	○	0	-	-	N	R	0
SR1389	Код ошибки обмена данными через соединение COM2 10	○	0	-	-	N	R	0
SR1390	Код ошибки обмена данными через соединение COM2 11	○	0	-	-	N	R	0
SR1391	Код ошибки обмена данными через соединение COM2 12	○	0	-	-	N	R	0
SR1392	Код ошибки обмена данными через соединение COM2 13	○	0	-	-	N	R	0
SR1393	Код ошибки обмена данными через соединение COM2 14	○	0	-	-	N	R	0
SR1394	Код ошибки обмена данными через соединение COM2 15	○	0	-	-	N	R	0
SR1395	Код ошибки обмена данными через соединение COM2 16	○	0	-	-	N	R	0
SR1396	Код ошибки обмена данными через соединение COM2 17	○	0	-	-	N	R	0
SR1397	Код ошибки обмена данными через соединение COM2 18	○	0	-	-	N	R	0
SR1398	Код ошибки обмена данными через соединение COM2 19	○	0	-	-	N	R	0
SR1399	Код ошибки обмена данными через соединение COM2 20	○	0	-	-	N	R	0
SR1400	Код ошибки обмена данными через соединение COM2 21	○	0	-	-	N	R	0
SR1401	Код ошибки обмена данными через соединение COM2 22	○	0	-	-	N	R	0



SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR1402	Код ошибки обмена данными через соединение COM2 23	○	0	-	-	N	R	0
SR1403	Код ошибки обмена данными через соединение COM2 24	○	0	-	-	N	R	0
SR1404	Код ошибки обмена данными через соединение COM2 25	○	0	-	-	N	R	0
SR1405	Код ошибки обмена данными через соединение COM2 26	○	0	-	-	N	R	0
SR1406	Код ошибки обмена данными через соединение COM2 27	○	0	-	-	N	R	0
SR1407	Код ошибки обмена данными через соединение COM2 28	○	0	-	-	N	R	0
SR1408	Код ошибки обмена данными через соединение COM2 29	○	0	-	-	N	R	0
SR1409	Код ошибки обмена данными через соединение COM2 30	○	0	-	-	N	R	0
SR1410	Код ошибки обмена данными через соединение COM2 31	○	0	-	-	N	R	0
SR1411	Код ошибки обмена данными через соединение COM2 32	○	0	-	-	N	R	0
SR1435	Фактическое время цикла для обмена данными через плату расширения 1	○	0	-	-	N	R	0
SR1436	Число соединений, которые в настоящее время выполняют циклический обмен данными через плату расширения 1	○	0	-	-	N	R	0
SR1440	Ошибка обмена данными соединения 1 через плату расширения 1	○	0	-	-	N	R	0
SR1441	Ошибка обмена данными соединения 2 через плату расширения 1	○	0	-	-	N	R	0
SR1442	Ошибка обмена данными соединения 3 через плату расширения 1	○	0	-	-	N	R	0
SR1443	Ошибка обмена данными соединения 4 через плату расширения 1	○	0	-	-	N	R	0
SR1444	Ошибка обмена данными соединения 5 через плату расширения 1	○	0	-	-	N	R	0
SR1445	Ошибка обмена данными соединения 6 через плату расширения 1	○	0	-	-	N	R	0
SR1446	Ошибка обмена данными соединения 7 через плату расширения 1	○	0	-	-	N	R	0
SR1447	Ошибка обмена данными соединения 8 через плату расширения 1	○	0	-	-	N	R	0
SR1448	Ошибка обмена данными соединения 9 через плату расширения 1	○	0	-	-	N	R	0
SR1449	Ошибка обмена данными соединения 10 через плату расширения 1	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR1450	Ошибка обмена данными соединения 11 через плату расширения 1	○	0	-	-	N	R	0
SR1451	Ошибка обмена данными соединения 12 через плату расширения 1	○	0	-	-	N	R	0
SR1452	Ошибка обмена данными соединения 13 через плату расширения 1	○	0	-	-	N	R	0
SR1453	Ошибка обмена данными соединения 14 через плату расширения 1	○	0	-	-	N	R	0
SR1454	Ошибка обмена данными соединения 15 через плату расширения 1	○	0	-	-	N	R	0
SR1455	Ошибка обмена данными соединения 16 через плату расширения 1	○	0	-	-	N	R	0
SR1456	Ошибка обмена данными соединения 17 через плату расширения 1	○	0	-	-	N	R	0
SR1457	Ошибка обмена данными соединения 18 через плату расширения 1	○	0	-	-	N	R	0
SR1458	Ошибка обмена данными соединения 19 через плату расширения 1	○	0	-	-	N	R	0
SR1459	Ошибка обмена данными соединения 20 через плату расширения 1	○	0	-	-	N	R	0
SR1460	Ошибка обмена данными соединения 21 через плату расширения 1	○	0	-	-	N	R	0
SR1461	Ошибка обмена данными соединения 22 через плату расширения 1	○	0	-	-	N	R	0
SR1462	Ошибка обмена данными соединения 23 через плату расширения 1	○	0	-	-	N	R	0
SR1463	Ошибка обмена данными соединения 24 через плату расширения 1	○	0	-	-	N	R	0
SR1464	Ошибка обмена данными соединения 25 через плату расширения 1	○	0	-	-	N	R	0
SR1465	Ошибка обмена данными соединения 26 через плату расширения 1	○	0	-	-	N	R	0
SR1466	Ошибка обмена данными соединения 27 через плату расширения 1	○	0	-	-	N	R	0
SR1467	Ошибка обмена данными соединения 28 через плату расширения 1	○	0	-	-	N	R	0
SR1468	Ошибка обмена данными соединения 29 через плату расширения 1	○	0	-	-	N	R	0
SR1469	Ошибка обмена данными соединения 30 через плату расширения 1	○	0	-	-	N	R	0
SR1470	Ошибка обмена данными соединения 31 через плату расширения 1	○	0	-	-	N	R	0
SR1471	Ошибка обмена данными соединения 32 через плату расширения 1	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR1475	Фактическое время цикла для обмена данными через плату расширения 2	○	0	-	-	N	R	0
SR1476	Число соединений, которые в настоящее время выполняют циклический обмен данными через плату расширения 2	○	0	-	-	N	R	0
SR1480	Ошибка обмена данными соединения 1 через плату расширения 2	○	0	-	-	N	R	0
SR1481	Ошибка обмена данными соединения 2 через плату расширения 2	○	0	-	-	N	R	0
SR1482	Ошибка обмена данными соединения 3 через плату расширения 2	○	0	-	-	N	R	0
SR1483	Ошибка обмена данными соединения 4 через плату расширения 2	○	0	-	-	N	R	0
SR1484	Ошибка обмена данными соединения 5 через плату расширения 2	○	0	-	-	N	R	0
SR1485	Ошибка обмена данными соединения 6 через плату расширения 2	○	0	-	-	N	R	0
SR1486	Ошибка обмена данными соединения 7 через плату расширения 2	○	0	-	-	N	R	0
SR1487	Ошибка обмена данными соединения 8 через плату расширения 2	○	0	-	-	N	R	0
SR1488	Ошибка обмена данными соединения 9 через плату расширения 2	○	0	-	-	N	R	0
SR1489	Ошибка обмена данными соединения 10 через плату расширения 2	○	0	-	-	N	R	0
SR1490	Ошибка обмена данными соединения 11 через плату расширения 2	○	0	-	-	N	R	0
SR1491	Ошибка обмена данными соединения 12 через плату расширения 2	○	0	-	-	N	R	0
SR1492	Ошибка обмена данными соединения 13 через плату расширения 2	○	0	-	-	N	R	0
SR1493	Ошибка обмена данными соединения 14 через плату расширения 2	○	0	-	-	N	R	0
SR1494	Ошибка обмена данными соединения 15 через плату расширения 2	○	0	-	-	N	R	0
SR1495	Ошибка обмена данными соединения 16 через плату расширения 2	○	0	-	-	N	R	0
SR1496	Ошибка обмена данными соединения 17 через плату расширения 2	○	0	-	-	N	R	0
SR1497	Ошибка обмена данными соединения 18 через плату расширения 2	○	0	-	-	N	R	0
SR1498	Ошибка обмена данными соединения 19 через плату расширения 2	○	0	-	-	N	R	0
SR1499	Ошибка обмена данными соединения 20 через плату расширения 2	○	0	-	-	N	R	0

SR	Функция	AS200 / AS300	OFF ↓ ON	STOP ↓ RUN	RUN ↓ STOP	Э/независимое	Атрибут	По умолчанию
SR1500	Ошибка обмена данными соединения 21 через плату расширения 2	○	0	-	-	N	R	0
SR1501	Ошибка обмена данными соединения 22 через плату расширения 2	○	0	-	-	N	R	0
SR1502	Ошибка обмена данными соединения 23 через плату расширения 2	○	0	-	-	N	R	0
SR1503	Ошибка обмена данными соединения 24 через плату расширения 2	○	0	-	-	N	R	0
SR1504	Ошибка обмена данными соединения 25 через плату расширения 2	○	0	-	-	N	R	0
SR1505	Ошибка обмена данными соединения 26 через плату расширения 2	○	0	-	-	N	R	0
SR1506	Ошибка обмена данными соединения 27 через плату расширения 2	○	0	-	-	N	R	0
SR1507	Ошибка обмена данными соединения 28 через плату расширения 2	○	0	-	-	N	R	0
SR1508	Ошибка обмена данными соединения 29 через плату расширения 2	○	0	-	-	N	R	0
SR1509	Ошибка обмена данными соединения 30 через плату расширения 2	○	0	-	-	N	R	0
SR1510	Ошибка обмена данными соединения 31 через плату расширения 2	○	0	-	-	N	R	0
SR1511	Ошибка обмена данными соединения 32 через плату расширения 2	○	0	-	-	N	R	0

\* 1: обратитесь к дополнительной информации по SM/SR

\* 2: коммуникационные платы здесь: AS-F232, AS-F422 и AS-F485

## 2.2.15 Условия обновления специальных регистров данных

Специальный регистр данных	Условие обновления специальных регистров данных
SR0~SR2	Регистр обновляется, когда программа выполняется с ошибкой
SR4~SR6	Регистр обновляется при ошибке проверки грамматики
SR8~SR9	Регистр обновляется при ошибке сторожевого таймера
SR23	Регистр обновляется при ошибке сторожевого таймера
SR28	Регистр обновляется, когда высокоскоростной выход с последним номером, который используется инструкцией, используется повторно
SR32	Регистр обновляется один раз при возникновении ошибки. -1 означает, что ошибка не произошла
SR36	Регистр обновляется пользователями. Пользователи устанавливают флаг SM36 в положение ON, и система будет сохранять данные на карту памяти. По завершении сохранения система автоматически сбрасывает ее в положение ВЫКЛ
SR40~SR161	Регистр обновляется при возникновении ошибки
SR162~SR163	После завершения работы ПЛК, каждый раз, когда питание ПЛК остается в течение 1 минуты, значение регистра увеличивается на 1
SR166~SR171	Регистр обновляется системой

Специальный регистр данных	Условие обновления специальных регистров данных
SR172~SR175	Регистр обновляется пользователем
SR176~SR179	Регистр обновляется в соответствии с настройками HWCONFIG
SR180	Регистр обновляется при включении и отключении ПЛК
SR182~SR183	Регистр обновляется в соответствии с настройками в HWCONFIG при включении ПЛК. После этого пользователь может редактировать настройки
SR185	Каждый раз, когда удаленный модуль активируется, после сканирования цикла система обновляет время цикла
SR187~SR197	Регистр обновляется пользователем
SR198~SR199	Регистр обновляется после подачи питания на ПЛК
SR201~SR213	Регистр обновляется в соответствии с настройками в HWCONFIG при включении ПЛК. После этого пользователь может редактировать настройки
SR215~SR216	Регистр обновляется системой
SR217~SR218	Регистр обновляется в соответствии с настройками в HWCONFIG при включении ПЛК. После этого пользователь может редактировать настройки
SR220~SR226	Регистр обновляется после каждого цикла сканирования
SR227~SR308	Регистр обновляется при загрузке программы в ПЛК
SR309~SR390	Регистр обновляется при изменении состояния ПЛК
SR391~SR397	Регистр обновляется после каждого цикла сканирования
SR407	Регистр обновляется каждую секунду
SR408~SR416	Регистр обновляется после выполнения инструкции END
SR421~SR424	Регистр обновляется пользователем
SR440~SR443	Регистр обновляется после подачи питания на ПЛК
SR444~SR451	Регистр обновляется после подачи питания на ПЛК
SR453	Регистр обновляется при возникновении ошибки
SR460	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR462~SR466	Регистр обновляется пользователем
SR467	Регистр обновляется системой
SR468~SR469	Когда на ПЛК подается питание, регистр обновляется в соответствии с таблицей планирования позиции. После этого пользователь может редактировать настройки
SR470	Регистр обновляется системой
SR472	Регистр обновляется пользователем
SR474	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR476~SR477	Регистр обновляется пользователем
SR480	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR482~SR486	Регистр обновляется пользователем
SR487	Регистр обновляется системой
SR488~SR489	Когда на ПЛК подается питание, регистр обновляется в соответствии с таблицей планирования позиции. После этого пользователь может редактировать настройки
SR490	Регистр обновляется системой
SR492	Регистр обновляется пользователем

Специальный регистр данных	Условие обновления специальных регистров данных
SR494	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR496~SR497	Регистр обновляется пользователем
SR500	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR502~SR506	Регистр обновляется пользователем
SR507	Регистр обновляется системой
SR508~SR509	Когда на ПЛК подается питание, регистр обновляется в соответствии с таблицей планирования позиции. После этого пользователь может редактировать настройки
SR510	Регистр обновляется системой
SR512	Регистр обновляется пользователем
SR514	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR516~SR517	Регистр обновляется пользователем
SR520	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR522~SR526	Регистр обновляется пользователем
SR527	Регистр обновляется системой
SR528~SR529	Когда на ПЛК подается питание, регистр обновляется в соответствии с таблицей планирования позиции. После этого пользователь может редактировать настройки
SR530	Регистр обновляется системой
SR532	Регистр обновляется пользователем
SR534	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR536~SR537	Регистр обновляется пользователем
SR540	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR542~SR546	Регистр обновляется пользователем
SR547	Регистр обновляется системой
SR548~SR549	Когда на ПЛК подается питание, регистр обновляется в соответствии с таблицей планирования позиции. После этого пользователь может редактировать настройки
SR550	Регистр обновляется системой
SR552	Регистр обновляется пользователем
SR554	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR556~SR557	Регистр обновляется пользователем
SR560	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR562~SR566	Регистр обновляется пользователем
SR567	Регистр обновляется системой

Специальный регистр данных	Условие обновления специальных регистров данных
SR568~SR569	Когда на ПЛК подается питание, регистр обновляется в соответствии с таблицей планирования позиции. После этого пользователь может редактировать настройки
SR570	Регистр обновляется системой
SR572	Регистр обновляется пользователем
SR574	Регистр обновляется системой каждый раз, когда выполняется инструкция для высокоскоростного выхода и программа сканируется. Если инструкция не выполнена, пользователь может редактировать настройки
SR576~SR577	Регистр обновляется пользователем
SR580~SR603	Регистр обновляется в соответствии с настройками в HWCONFIG при включении ПЛК. После этого пользователь может редактировать настройки
SR604~SR609	Регистр обновляется пользователем
SR610~SR621	Регистр обновляется при выполнении инструкции для выхода
SR623~SR634	Регистр обновляется всякий раз, когда выполняется команда EIX или DIX. ВКЛ: прерывание включено ВЫКЛ: прерывание отключено
SR658~SR748	Регистр обновляется системой
SR751~SR768	Регистр обновляется пользователем
SR771~SR788	Регистр обновляется системой
SR820	Регистр обновляется в соответствии с настройками в CANopen Builder
SR821	Регистр обновляется при изменении прошивки
SR822	Регистр обновляется в соответствии с настройками HWCONFIG
SR825~SR893	Регистр обновляется системой
SR902	Регистр обновляется пользователем
SR1000~SR1006	Регистр обновляется пользователем
SR1007	Регистр обновляется системой
SR1009	Регистр обновляется системой
SR1010	Регистр обновляется пользователем
SR1011~SR1014	Регистр обновляется системой
SR1020~SR1107	1. Регистр обновляется при установлении соединения. 2. Регистр обновляется каждый цикл сканирования
SR1116~SR1117	Регистр обновляется при загрузке программы в ПЛК
SR1120~SR1183	Регистр обновляется при выполнении сеанса коммуникации
SR1318~SR1320	Регистр обновляется, когда параметр загружается в ПЛК или когда на ПЛК подается питание
SR1335~SR1336	После включения функции обмена данными регистр обновляется каждый цикл сканирования
SR1340~SR1371	Регистр обновляется при возникновении ошибки
SR1375~SR1376	После включения функции обмена данными регистр обновляется каждый цикл сканирования
SR1380~SR1411	Регистр обновляется при возникновении ошибки
SR1435~SR1436	Регистр обновляется во время коммуникации
SR1440~SR1471	Регистр обновляется при ошибке коммуникации
SR1475~SR1476	Регистр обновляется во время коммуникации
SR1480~SR1511	Регистр обновляется при ошибке коммуникации

## 2.2.16 Дополнительная информация по SM и SR

### 1. Сторожевой таймер

- SM8/SR8

Когда возникает пауза при выполнении программы, все время горит светодиодный индикатор ошибки на ПЛК, а SM8 в это время включен.

Содержание SR8 - это адрес шага, по которому включен сторожевой таймер.

## 2. Сброс сигнала на светодиодном индикаторе ошибок

- SM22

Если SM22 включен, журнал ошибок и сигнал на светодиодный индикатор ошибок будут очищены

## 3. Часы реального времени

- SM220, SR220~SR226 и SR391~SR397

SM220: Калибровка часов реального времени в пределах  $\pm 30$  секунд

Когда SM220 включается, часы реального времени откалиброваны.

Если значение секунд в часах реального времени находится в диапазоне от 0 до 29, значение минут фиксируется, а значение секунд - обнуляется.

Если значение секунд в часах реального времени находится в диапазоне от 30 до 59, значение минут увеличивается на единицу, а значение секунд - обнуляется.

Соответствующие функции и значения SR220 ~ SR226 и SR391 ~ SR397 следующие.

Объект		Функция	Значение
Двоично-десятичная система	Десятичная система		
SR220	SR391	Год	00~99 (A.D.)
SR221	SR392	Месяц	1~12
SR222	SR393	День	1~31
SR223	SR394	Час	0~23
SR224	SR395	Минута	0~59
SR225	SR396	Секунда	0~59
SR226	SR397	День недели	1~7

Регистры SR391 ~ SR397 соответствуют регистрам SR220 ~ SR226. Разница между регистрами SR220 ~ SR226 и SR391 ~ SR397 заключается в том, что первые содержат двоично-десятичное число, а вторые - десятичное число. Например, декабрь представлен как 12 в десятичном формате в SR392, тогда как в SR221 он представлен как 12 в двоично-десятичном формате. Дополнительную информацию, связанную с инструкцией по применению часов реального времени, см. в Главе 6.

## 4. Функции коммуникации

- SM96~SM107, SM209~SM212, SR201~SR202 и SR209~SR216

Регистры SR215 и SR216 используются для записи кода интерфейса коммуникационного порта в ПЛК. Функции, представленные кодами интерфейса, следующие.

Код	0	1	2	3	4	5	6
Функция	Нет	RS232	RS422	RS485	F2AD	F2DA	FCOPM

Функциональная плата FCOPM применяется только как функциональная плата 2.

Когда интерфейсом коммуникационного порта на ПЛК является RS485, RS232 или RS422, в SR209



записывается коммуникационный формат COM1, а в SR212 записывается коммуникационный формат COM2 в ПЛК. Значения заданий протоколов коммуникации показаны в следующей таблице.

Дополнительную информацию, связанную с инструкциями по коммуникации, см. в Главе 6.

<b>b0</b>	Длина данных			7 (значение=0)	8 (значение=1)
<b>b1 b2</b>	Бит четности			00	: Нет
				01	: Нечетный
				10	: Четный
<b>b3</b>	Стоп бит			1 бит (значение=0)	2 бита (значение=1)
<b>b4 b5 b6 b7</b>	0001	( H 1 )	:	4800	
	0010	( H 2 )	:	9600	
	0011	( H 3 )	:	19200	
	0100	( H 4 )	:	38400	
	0101	( H 5 )	:	57600	
	0110	( H 6 )	:	115200	
	0111	( H 7 )	:	230400	
	1000	( H 8 )	:	500000	
	1001	( H 9 )	:	921000	
	1010	( 16#A )	:	Не определен	
	1011	( 16#B )	:	Не определен	
	1100	( 16#C )	:	Не определен	
	1101	( 16#D )	:	Не определен	
1110	( 16#E )	:	Не определен		
1111	( 16#F )	:	Определяется пользователем <sup>*1</sup>		
<b>b8-b15</b>	Не определен (зарезервирован)				

\*1: См. настройки HWCONFIG в ISPSOft.

\*2: См. Раздел 6.19.3 для информации по флагам и регистрам.

## 5. Очистка содержимого объекта

- SM204/SM205

Номер объекта	Очищаемый объект
SM204 Очистка всех энергозависимых областей	Очистка всех входных и выходных реле, шаговых реле, вспомогательных реле, таймеров, счетчиков, 32-битных счетчиков, регистров данных и индексных регистров в энергозависимой области. Сторожевой таймер в это время не работает.
SM205 Очистка всех энергонезависимых областей	Очистка всех входных и выходных реле, шаговых реле, вспомогательных реле, таймеров, счетчиков, 32-битных счетчиков, регистров данных и индексных регистров в энергонезависимой области. Сторожевой таймер в это время не работает.

См. Раздел 2.1.4 для получения дополнительной информации, касающейся энергонезависимых областей в диапазоне объектов.

## 6. Журнал ошибок в ПЛК

- SR40~SR161

SR40: Максимальное количество журналов ошибок, хранящихся в SR40, равно 20. Каждый журнал ошибок занимает 6 регистров.

SR41: Указатель журнала ошибок указывает на последний журнал ошибок. При возникновении ошибки значение указателя журнала ошибок увеличивается на единицу. Диапазон значений указателя равен 0 ~ 19. Например, указатель журнала ошибок указывает на четвертый журнал ошибок, когда значение в SR41 равно 3.

Время, когда возникают ошибки, и позиции, где происходят ошибки, записываются в SR42 ~ SR161.

Соответствующие функции этих регистров данных следующие.

Номер	Слот	Модуль ID	Код ошибки	Время возникновения ошибки					
				Год	Месяц	День	Час	Минута	Секунда
1	SR42 Мл.байт	SR43	SR44	SR45 Ст.байт	SR45 Мл.байт	SR46 Ст.байт	SR46 Мл.байт	SR47 Ст.байт	SR47 Мл.байт
2	SR48 Мл.байт	SR49	SR50	SR51 Ст.байт	SR51 Мл.байт	SR52 Ст.байт	SR52 Мл.байт	SR53 Ст.байт	SR53 Мл.байт
3	SR54 Мл.байт	SR55	SR56	SR57 Ст.байт	SR57 Мл.байт	SR58 Ст.байт	SR58 Мл.байт	SR59 Ст.байт	SR59 Мл.байт
4	SR60 Мл.байт	SR61	SR62	SR63 Ст.байт	SR63 Мл.байт	SR64 Ст.байт	SR64 Мл.байт	SR65 Ст.байт	SR65 Мл.байт
5	SR66 Мл.байт	SR67	SR68	SR69 Ст.байт	SR69 Мл.байт	SR70 Ст.байт	SR70 Мл.байт	SR71 Ст.байт	SR71 Мл.байт
6	SR72 Мл.байт	SR73	SR74	SR75 Ст.байт	SR75 Мл.байт	SR76 Ст.байт	SR76 Мл.байт	SR77 Ст.байт	SR77 Мл.байт
7	SR78 Мл.байт	SR79	SR80	SR81 Ст.байт	SR81 Мл.байт	SR82 Ст.байт	SR82 Мл.байт	SR83 Ст.байт	SR83 Мл.байт
8	SR84 Мл.байт	SR85	SR86	SR87 Ст.байт	SR87 Мл.байт	SR88 Ст.байт	SR88 Мл.байт	SR89 Ст.байт	SR89 Мл.байт
9	SR90 Мл.байт	SR91	SR92	SR93 Ст.байт	SR93 Мл.байт	SR94 Ст.байт	SR94 Мл.байт	SR95 Ст.байт	SR95 Мл.байт
10	SR96 Мл.байт	SR97	SR98	SR99 Ст.байт	SR99 Мл.байт	SR100 Ст.байт	SR100 Мл.байт	SR101 Ст.байт	SR101 Мл.байт
11	SR102 Мл.байт	SR103	SR104	SR105 Ст.байт	SR105 Мл.байт	SR106 Ст.байт	SR106 Мл.байт	SR107 Ст.байт	SR107 Мл.байт
12	SR108 Мл.байт	SR109	SR110	SR111 Ст.байт	SR111 Мл.байт	SR112 Ст.байт	SR112 Мл.байт	SR113 Ст.байт	SR113 Мл.байт
13	SR114 Мл.байт	SR115	SR116	SR117 Ст.байт	SR117 Мл.байт	SR118 Ст.байт	SR118 Мл.байт	SR119 Ст.байт	SR119 Мл.байт
14	SR120 Мл.байт	SR121	SR122	SR123 Ст.байт	SR123 Мл.байт	SR124 Ст.байт	SR124 Мл.байт	SR125 Ст.байт	SR125 Мл.байт
15	SR126 Мл.байт	SR127	SR128	SR129 Ст.байт	SR129 Мл.байт	SR130 Ст.байт	SR130 Мл.байт	SR131 Ст.байт	SR131 Мл.байт
16	SR132 Мл.байт	SR133	SR134	SR135 Ст.байт	SR135 Мл.байт	SR136 Ст.байт	SR136 Мл.байт	SR137 Ст.байт	SR137 Мл.байт
17	SR138 Мл.байт	SR139	SR140	SR141 Ст.байт	SR141 Мл.байт	SR142 Ст.байт	SR142 Мл.байт	SR143 Ст.байт	SR143 Мл.байт
18	SR144 Мл.байт	SR145	SR146	SR147 Ст.байт	SR147 Мл.байт	SR148 Ст.байт	SR148 Мл.байт	SR149 Ст.байт	SR149 Мл.байт

Номер	Слот	Модуль ID	Код ошибки	Время возникновения ошибки					
				Год	Месяц	День	Час	Минута	Секунда
19	SR150 Мл.байт	SR151	SR152	SR153 Ст.байт	SR153 Мл.байт	SR154 Ст.байт	SR154 Мл.байт	SR155 Ст.байт	SR155 Мл.байт
20	SR156 Мл.байт	SR157	SR158	SR159 Ст.байт	SR159 Мл.байт	SR160 Ст.байт	SR160 Мл.байт	SR161 Ст.байт	SR161 Мл.байт

## 7. Журнал загрузок в ПЛК

- SR227~SR308

SR227: Максимальное количество журналов загрузки, хранящихся в SR227, равно 20. Каждый журнал загрузки занимает 4 регистра. Записанные операции загрузки нумеруются, как показано в следующей таблице.

Действие при загрузке	Номер
Загрузка программы	1
Загрузка настроек ПЛК	2
Загрузка таблицы модулей	3

SR228: указатель журнала загрузки указывает на последний журнал загрузки. Когда действие загрузки выполняется, значение указателя журнала загрузки увеличивается на единицу. Диапазон значений указателя равен 0 ~ 19. Например, указатель журнала загрузки указывает на четвертый журнал загрузки, когда значение в SR228 равно 3.

Время выполнения действий по загрузке и номера действий записываются в SR229 ~ SR30.

Соответствующие функции этих регистров данных следующие.

Номер	Номер действия	*Время осуществления загрузки					
		Год	Месяц	День	Час	Минута	Секунда
1	SR229	SR230 Ст.байт	SR230 Мл.байт	SR231 Ст.байт	SR231 Мл.байт	SR232 Ст.байт	SR232 Мл.байт
2	SR233	SR234 Ст.байт	SR234 Мл.байт	SR235 Ст.байт	SR235 Мл.байт	SR236 Ст.байт	SR236 Мл.байт
3	SR237	SR238 Ст.байт	SR238 Мл.байт	SR239 Ст.байт	SR239 Мл.байт	SR240 Ст.байт	SR240 Мл.байт
4	SR241	SR242 Ст.байт	SR242 Мл.байт	SR243 Ст.байт	SR243 Мл.байт	SR244 Ст.байт	SR244 Мл.байт
5	SR245	SR246 Ст.байт	SR246 Мл.байт	SR247 Ст.байт	SR247 Мл.байт	SR248 Ст.байт	SR248 Мл.байт
6	SR249	SR250 Ст.байт	SR250 Мл.байт	SR251 Ст.байт	SR251 Мл.байт	SR252 Ст.байт	SR252 Мл.байт
7	SR253	SR254 Ст.байт	SR254 Мл.байт	SR255 Ст.байт	SR255 Мл.байт	SR256 Ст.байт	SR256 Мл.байт
8	SR257	SR258 Ст.байт	SR258 Мл.байт	SR259 Ст.байт	SR259 Мл.байт	SR260 Ст.байт	SR260 Мл.байт
9	SR261	SR262 Ст.байт	SR262 Мл.байт	SR263 Ст.байт	SR263 Мл.байт	SR264 Ст.байт	SR264 Мл.байт
10	SR265	SR266 Ст.байт	SR266 Мл.байт	SR267 Ст.байт	SR267 Мл.байт	SR268 Ст.байт	SR268 Мл.байт

Номер	Номер действия	*Время осуществления загрузки					
		Год	Месяц	День	Час	Минута	Секунда
11	SR269	SR270 Ст.байт	SR270 Мл.байт	SR271 Ст.байт	SR271 Мл.байт	SR272 Ст.байт	SR272 Мл.байт
12	SR273	SR274 Ст.байт	SR274 Мл.байт	SR275 Ст.байт	SR275 Мл.байт	SR276 Ст.байт	SR276 Мл.байт
13	SR277	SR278 Ст.байт	SR278 Мл.байт	SR279 Ст.байт	SR279 Мл.байт	SR280 Ст.байт	SR280 Мл.байт
14	SR281	SR282 Ст.байт	SR282 Мл.байт	SR283 Ст.байт	SR283 Мл.байт	SR284 Ст.байт	SR284 Мл.байт
15	SR285	SR286 Ст.байт	SR286 Мл.байт	SR287 Ст.байт	SR287 Мл.байт	SR288 Ст.байт	SR288 Мл.байт
16	SR289	SR290 Ст.байт	SR290 Мл.байт	SR291 Ст.байт	SR291 Мл.байт	SR292 Ст.байт	SR292 Мл.байт
17	SR293	SR294 Ст.байт	SR294 Мл.байт	SR295 Ст.байт	SR295 Мл.байт	SR296 Ст.байт	SR296 Мл.байт
18	SR297	SR298 Ст.байт	SR298 Мл.байт	SR299 Ст.байт	SR299 Мл.байт	SR300 Ст.байт	SR300 Мл.байт
19	SR301	SR302 Ст.байт	SR302 Мл.байт	SR303 Ст.байт	SR303 Мл.байт	SR304 Ст.байт	SR304 Мл.байт
20	SR305	SR306 Ст.байт	SR306 Мл.байт	SR307 Ст.байт	SR307 Мл.байт	SR308 Ст.байт	SR308 Мл.байт

\* Время осуществления загрузки: данные сохраняются как значения в двоично-десятичном формате. Диапазон значений следующий:

Функция	Значение
Год	00~99 (A.D.)
Месяц	01~12
День	01~31
Час	00~23
Минута	00~59
Секунда	00~59

## 8. Журнал состояния ПЛК

- SR309~SR390

SR309: Максимальное количество журналов изменений состояния PLC, которые хранятся в SR309, равно 20. Каждый журнал изменений состояния PLC занимает 4 регистра. Записанные действия изменения состояния ПЛК нумеруются, как показано в следующей таблице.

Изменение состояния ПЛК	Номер
На ПЛК подано питание	1
Питание ПЛК отключено	2
ПЛК начинает работу	3
ПЛК останавливает работу	4

Изменение состояния ПЛК	Номер
Настройка ПЛК по умолчанию	5

SR310: указатель журнала изменений состояния ПЛК указывает на последний журнал изменений состояния PLC. Когда состояние ПЛК изменяется один раз, значение указателя журнала изменения состояния ПЛК увеличивается на единицу. Диапазон значений указателя равен 0 ~ 19. Например, указатель журнала изменений состояния ПЛК указывает на четвертый журнал изменений состояния ПЛК, когда значение в SR310 равно 3.

Время, в которое происходят действия изменения состояния ПЛК, записывается в SR311 ~ SR390. Соответствующие функции этих регистров данных следующие.

Номер	Номер действия	*Время осуществления загрузки					
		Год	Год	Год	Год	Год	Год
1	SR311	SR312 Ст.байт	SR312 Мл.байт	SR313 Ст.байт	SR313 Мл.байт	SR314 Ст.байт	SR314 Мл.байт
2	SR315	SR316 Ст.байт	SR316 Мл.байт	SR317 Ст.байт	SR317 Мл.байт	SR318 Ст.байт	SR318 Мл.байт
3	SR319	SR320 Ст.байт	SR320 Мл.байт	SR321 Ст.байт	SR321 Мл.байт	SR322 Ст.байт	SR322 Мл.байт
4	SR323	SR324 Ст.байт	SR324 Мл.байт	SR325 Ст.байт	SR325 Мл.байт	SR326 Ст.байт	SR326 Мл.байт
5	SR327	SR328 Ст.байт	SR328 Мл.байт	SR329 Ст.байт	SR329 Мл.байт	SR330 Ст.байт	SR330 Мл.байт
6	SR331	SR332 Ст.байт	SR332 Мл.байт	SR333 Ст.байт	SR333 Мл.байт	SR334 Ст.байт	SR334 Мл.байт
7	SR335	SR336 Ст.байт	SR336 Мл.байт	SR337 Ст.байт	SR337 Мл.байт	SR338 Ст.байт	SR338 Мл.байт
8	SR339	SR340 Ст.байт	SR340 Мл.байт	SR341 Ст.байт	SR341 Мл.байт	SR342 Ст.байт	SR342 Мл.байт
9	SR343	SR344 Ст.байт	SR344 Мл.байт	SR345 Ст.байт	SR345 Мл.байт	SR346 Ст.байт	SR346 Мл.байт
10	SR347	SR348 Ст.байт	SR348 Мл.байт	SR349 Ст.байт	SR349 Мл.байт	SR350 Ст.байт	SR350 Мл.байт
11	SR351	SR352 Ст.байт	SR352 Мл.байт	SR353 Ст.байт	SR353 Мл.байт	SR354 Ст.байт	SR354 Мл.байт
12	SR355	SR356 Ст.байт	SR356 Мл.байт	SR357 Ст.байт	SR357 Мл.байт	SR358 Ст.байт	SR358 Мл.байт
13	SR359	SR360 Ст.байт	SR360 Мл.байт	SR361 Ст.байт	SR361 Мл.байт	SR362 Ст.байт	SR362 Мл.байт
14	SR363	SR364 Ст.байт	SR364 Мл.байт	SR365 Ст.байт	SR365 Мл.байт	SR366 Ст.байт	SR366 Мл.байт
15	SR367	SR368 Ст.байт	SR368 Мл.байт	SR369 Ст.байт	SR369 Мл.байт	SR370 Ст.байт	SR370 Мл.байт
16	SR371	SR372 Ст.байт	SR372 Мл.байт	SR373 Ст.байт	SR373 Мл.байт	SR374 Ст.байт	SR374 Мл.байт
17	SR375	SR376 Ст.байт	SR376 Мл.байт	SR377 Ст.байт	SR377 Мл.байт	SR378 Ст.байт	SR378 Мл.байт

Номер	Номер действия	*Время осуществления загрузки					
		Год	Год	Год	Год	Год	Год
18	SR379	SR380 Ст.байт	SR380 Мл.байт	SR381 Ст.байт	SR381 Мл.байт	SR382 Ст.байт	SR382 Мл.байт
19	SR383	SR384 Ст.байт	SR384 Мл.байт	SR385 Ст.байт	SR385 Мл.байт	SR386 Ст.байт	SR386 Мл.байт
20	SR387	SR388 Ст.байт	SR388 Мл.байт	SR389 Ст.байт	SR389 Мл.байт	SR390 Ст.байт	SR390 Мл.байт

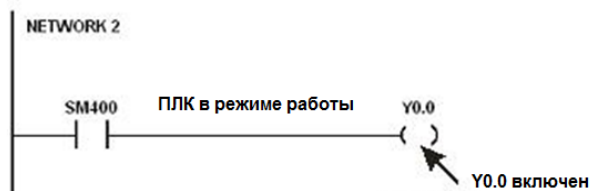
\* Время осуществления загрузки: данные сохраняются как значения в двоично-десятичном формате. Диапазон значений следующий:

Функция	Значение
Год	00~99 (A.D.)
Месяц	01~12
День	01~31
Час	00~23
Минута	00~59
Секунда	00~59

### 9. Флаг работы ПЛК

- SM400~SM403

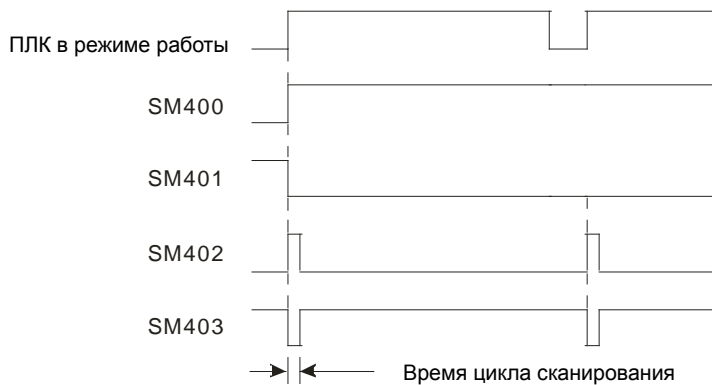
SM400: Н/О контакт



SM401: Н/З контакт

SM402: SM402 остается включенным в течение первого цикла сканирования, а затем выключается. Ширина импульса при этом равна времени одного цикла сканирования. Пользователь может использовать этот контакт для первоначальной настройки.

SM403: SM403 выключен во время первого цикла сканирования, а затем включается. То есть, отрицательный импульс генерируется в тот момент, когда ПЛК запускается.



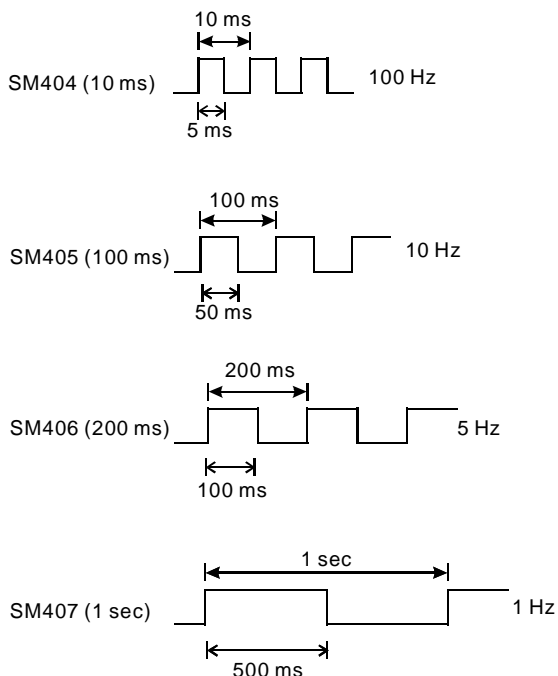
**10. Тактовый импульс**

- SM404 · SM405 · SM406 · SM407

ПЛК обеспечивает семь типов тактовых импульсов. Когда включается питание ПЛК, автоматически задействуются семь типов тактовых импульсов.

Объект	Функция
SM404	10 миллисекундный тактовый импульс, в течение которого импульс включен в течение 5 миллисекунд и выключен на 5 миллисекунд
SM405	100 миллисекундный тактовый импульс, в течение которого импульс включен в течение 50 миллисекунд и выключен на 50 миллисекунд
SM406	200 миллисекундный тактовый импульс, в течение которого импульс включен в течение 100 миллисекунд и выключен на 100 миллисекунд
SM407	1 секундный тактовый импульс, в течение которого импульс включен в течение 500 миллисекунд и выключен на 500 миллисекунд

Тактовые импульсы показаны ниже.



**11. Флаги, относящиеся к карте памяти**

- SM36 · SM450~SM453 · SM456 · SR36 · SR453 · SR902

Карта памяти используется для резервного копирования данных в ПЛК. См. Глава 6 за дополнительными сведениями относительно карты памяти.

Объект	Функция
SM36	Включение сохранения данных на карту памяти. ПЛК будет работать в соответствии со значением в SR36
SM450	Наличие карты памяти ВКЛ: Карта памяти установлена ВЫКЛ: Карта памяти не установлена
SM452	Доступ к данным на карте памяти ВКЛ: Данные на карте памяти доступны ВЫКЛ: Данные на карте памяти недоступны

Объект	Функция
SM453	Ошибка при работе с картой памяти ON: Ошибка возникла
SM456	Взаимодействие регистратора ПЛК и карты памяти (ВКЛ: действия согласно значению в SR902)
SR36	Сохранение данных на карту памяти. Функция работает только вместе с SM36
SR453	При возникновении ошибки при работе с картой памяти в регистр записывается код ошибки
SR902	Код для взаимодействия регистратора ПЛК и карты памяти (работа совместно с SM456), например, код H5AA5: запись данных из регистратора ПЛК на карту памяти

- SR36 сохраняет только 2 журнала :

А. Число 1234 означает, что журнал ошибок (SR40 ~ SR161) ПЛК сохраняется на карте памяти.

В. Число 3456 означает журнал ошибок (SR40 ~ SR161) ПЛК и журнал изменения состояния ПЛК (SR309 ~ SR390) сохраняются на карте памяти.

## 12. Выполнение инструкции для высокоскоростного выхода. Выход немедленно отключается, когда инструкция отключена или остановлено ее выполнение

- SM476, SM477, SM496, SM497, SM516, SM517, SM536, SM537, SM556, SM557, SM576, SM577:

OFF (по умолчанию): останов с замедлением

ON: немедленный стоп

- SM463, SM474, SM483, SM494, SM503, SM514, SM523, SM534, SM543, SM554, SM563, SM574 : флаги используются для приостановки работы выхода.
- Когда флаг включается, работа выхода будет остановлена. Функция должна работать с вышеперечисленными флагами; обратитесь к разделу выше для получения дополнительных действий при останове.
- Когда флаг выключается, это значит, что все выходы отключены

## 13. Выходной предел управления положением в ISPSoft

- SR580~SR603

Положительный выходной предел: установка предела в ISPSoft; когда выходное положение больше положительного предела, работа выхода останавливается немедленно

Отрицательный выходной предел: установка предела в ISPSoft; когда выходное положение меньше отрицательного предела, работа выхода останавливается немедленно.

Когда положительный и отрицательный пределы выхода равны 0, что функция отключена. Эта функция должна работать с инструкцией для работы с выходом. Система проверит только ограничение, установленное в ISPSoft, когда инструкция выполнена. Таким образом, выход не будет немедленно остановлен, даже если положение выходит за пределы выходного сигнала. Если требуется немедленная остановка, рекомендуется использовать внешний вход для проверки предела.

## 14. Режим S-кривой

- SR604~SR609, SM468, SM488, SM508, SM528, SM548, SM568



Есть 3 вида S-кривых: малая, средняя и большая. Диапазон значения находится в пределах от 0 до 2. Когда значение превышает диапазон, система будет обрабатывать значение как минимум 0 или максимум 2.

Режим S-кривой должен работать со следующими флагами: SM468, SM488, SM568. Если флаг включен, параметры S-кривой будут выполняться инструкцией для выхода.

## 15. Функция компенсации люфта

- SR478, SR479, SR498, SR499, SR518, SR519, SR538, SR539, SR558, SR559, SR578, SR579

Для ПЛК серии AS можно установить до 12 высокоскоростных выходов (Y0.0-Y0.11). Каждый выход работает с соответствующим регистром SR, задающим номер выхода для компенсаций люфта. Диапазон настройки: 0-32767. Если значение настройки  $\leq 0$ , эта функция отключена.

- Эта функция доступна для прошивки версии 1.02.30 и более поздних версий. Выходные инструкции, которые поддерживают выходные сигналы с нечетными номерами по умолчанию имеют направленный выход: JOG, DZRN, DPLSV, DDRVI, DDRVA, DPPMR, DPPMA, DCICR, DCICA, DCICCR, DCICCA, DCCMR, DCCMA, DPPGB и TPO. Для четных выходов можно использовать следующие инструкции: JOG, DPLSV, DDRVI и DDRVA.

## 16. Флаги, относящиеся к Ethernet IP

- SM1000 \ SR1000~SR1006

SM/SR	Функция	Действие
SM1000	Флаг настройки Ethernet	ВКЛ: значения в SR1000 ~ SR1006 записываются во флэш-память. После выполнения записи ПЛК отключит флаг. ПРИМЕЧАНИЕ 1: не устанавливайте флаг в положение ВКЛ постоянно, чтобы избежать повреждения флэш-памяти. ПРИМЕЧАНИЕ 2: чтобы записать значения во флэш-память требуется, чтобы ПЛК находился в состоянии останова,
SR1000	IP адрес Ethernet (32-бит)	Например: 192.168.1.5, SR1000 должен иметь значение 16#C0A8 и SR1001 должен иметь значение 0105.
SR1001		
SR1002	Адрес сетевой маски Ethernet (32-бит)	Например: 255.255.255.0, SR1002 должен иметь значение 16#FFFF и SR1003 должен иметь значение FF00
SR1003		
SR1004	Адрес шлюза Ethernet (32-бит)	Например: 192.168.1.1, SR1004 должен иметь значение 16#C0A8 и SR1005 должен иметь значение 0101
SR1005		
SR1006	Время, в течение которого TCP-соединение было постоянным	Ед. изм.: секунды

- SM1090 \ SM1091 \ SM1106~SM1109

SM	Функция	Действие
SM1090	Соединение TCP занято	ВКЛ: превышение времени ожидания соединения TCP
SM1091	Соединение UDP занято	ВКЛ: превышение времени ожидания соединения UDP
SM1106	Ошибка подключения Ethernet	ВЫКЛ: инициализация PNY завершена успешно
SM1107	Ошибка базовой настройки Ethernet	ОН: инициализация PNY завершена с ошибкой
SM1109	TCP / UDP socket - локальный порт уже используется	ВЫКЛ: правильная базовая настройка

Для информации о кодах ошибок, соответствующих сигналах светодиодных индикаторов см. Главу 12 Руководства по эксплуатации ПЛК серии AS.

**17. Настройки E-mail**

- SM1113 · SM1116~SM1155

Если при отправке E-mail произошла ошибка, включается флаг SM1113.

Триггеры отправки E-mail и соответствующие флаги (SM1116 ~ SM1155) перечислены ниже:

Функция \ Пункт	Триггер 1	Триггер 2	Триггер 3	Триггер 4
Служба E-mail	SM1116	SM1126	SM1136	SM1146
	ВКЛ: запущена , ВЫКЛ: отключена			
Отправка E-mail	SM1117	SM1127	SM1137	SM1147
	ВКЛ: E-mail отправляется в настоящее время, ВЫКЛ: E-mail уже отправлен			
Успешная отправка E-mail	SM1119	SM1129	SM1139	SM1149
	ВКЛ: E-mail отправлен удачно			
Ошибка 1 отправки E-mail	SM1120	SM1130	SM1140	SM1150
	Электронная почта не может быть отправлена из-за ошибки содержимого			
Превышение времени ожидания ответа SMTP	SM1122	SM1132	SM1142	SM1152
	После отправки электронной почты превышено время ожидания ответа сервера SMTP			
Ошибка сервера при отклике SMTP	SM1123	SM1133	SM1143	SM1153
	После отправки электронной почты произошла ошибка ответа сервера SMTP			
Ошибка 2 отправки E-mail	SM1124	SM1134	SM1144	SM1154
	Размер вложения в сообщение E-mail превышает допустимый диапазон			
Ошибка 3 отправки E-mail	SM1125	SM1135	SM1145	SM1155
	Вложение в E-mail не найдено, SM1125 включен.			

**18. Флаги и регистры, работающие с обменом данными**

- Флаги для обмена данными через порт COM1

SM	Тип	Функция
SM750	R/W	Обмен данными через COM1 был активирован ISPSoft
SM752 ~ SM783	R/W	Запущено соединение 1 ~ 32 через COM1 для обмена данными
SM784 ~ SM815	R	Данные были получены через соединение 1 ~ 32 порта COM 1
SM816 ~ SM847	R	Ошибка в соединении 1 ~ 32 порта COM1

- Флаги для обмена данными через порт COM2

SM	Тип	Функция
SM862	R/W	Обмен данными через COM2 был включен ISPSoft
SM864 ~ SM895	R/W	Запущено соединение 1 ~ 32 через COM2 для обмена данными
SM896 ~ SM927	R	Данные были получены через соединение 1 ~ 32 порта COM 2
SM928 ~ SM959	R	Ошибка в соединении 1 ~ 32 порта COM2

- Регистры данных для соединения через COM1

SR	Функции
SR1335	Фактическое время цикла соединения 1 ~ 32 для обмена данными через COM1
SR1336	Количество соединений, которое в настоящее время выполняет циклический обмен данными через COM1
SR1340 ~ SR1371	Коды ошибок для обмена данными через соединение 1 ~ 32 порта COM1

- Регистры данных для соединения через COM2

SR1375	Фактическое время цикла соединения 1 ~ 32 для обмена данными через COM2
SR1376	Количество соединений, которое в настоящее время выполняет циклический обмен данными через COM2
SR1380 ~ SR1411	Коды ошибок для обмена данными через соединение 1 ~ 32 порта COM2

Коды ошибок 1 ~ 7 являются стандартными кодами ошибок ответа для протокола MODBUS. Код ошибки 9 означает превышение времени ожидания.

- Флаги для обмена данными через Ethernet

SM	Тип	Функция
SM1167	R/W	ISPSoft поддерживает обмен данными через порт Ethernet
SM1168 ~ SM1199	R/W	Запущено подключение 1 ~ 32 через Ethernet для обмена данными
SM1200 ~ SM1231	R	Данные были получены через Ethernet (подключение 1 ~ 32)
SM1232 ~ SM1263	R	Произошла ошибка в соединении 1 ~ 32 порта Ethernet

- Файловые регистры для обмена данными через Ethernet

SR	Функция
SR1120 ~ SR1151	Фактическое время соединения для обмена данными через Ethernet-соединение 1 ~ 32
SR1152 ~ SR1183	Код ошибки для обмена данными через Ethernet-соединение 1 ~ 32

- Коды ошибок Ethernet-соединения

Код ошибки	Описание
16#00XX	Ошибка ответа удаленного модуля
16#F000	Соединение Ethernet не установлено
16#F001	Превышение времени ожидания ответа удаленного модуля
16#F003	Превышение времени ожидания соединения TCP
16#F007	Ошибка отклика
16#F009	Потеряно соединение с удаленным модулем

- Список состояний SM / SR при подключении к RTU-EN01 через порт Ethernet

SM / SR	Описание	
SM1312 - SM1315	Флаги состояния коммуникации с RTU-EN01 по подключениям ID 1~4	
SM1312 - SM1315	Коды состояния коммуникации с RTU-EN01 по подключениям ID 1~4	
SR	Состояние SM	Описание
0	Выкл	Соединение закрыто
1	Вкл	Соединение установлено
2	Выкл	Пауза связи
3	Выкл	Соединение разорвано принудительно
4	Выкл	Отклик RTU-EN01: ошибка содержимого
5	Выкл	Отклик RTU-EN01: ошибка
6	Выкл	Сеть не подключена или соединение сброшено
Состояние SM		Чтение данных RX и RCR (входная область)
Выкл → Вкл		Запись данных RY и RCR (выходная область)
		Сброс на 0
		Сброс на 0

Вкл → Выкл	Без изменений	Без изменений
------------	---------------	---------------

Примечание: Перед установкой соединения HE рекомендуется использовать данные (RX / RY / RCR Read / RCR Write) в область отображения (входной или выходной).

### 2.2.17 Индексные регистры (E)

Индексный регистр - это 16-разрядный регистр данных. Он похож на общий регистр, в который данные можно записать данные и прочитать данные из него. Однако он в основном используется как индексный регистр.

Диапазон индексных регистров E0 ~ E9. Дополнительную информацию об использовании индексных регистров см. в Разделе 4.4 настоящего Руководства по программированию.

### 2.2.18 Файловые регистры (FR)

- ПЛК серии AS предоставляет пользователю файловые регистры для хранения большего количества параметров.
- Пользователь может редактировать, выгружать, загружать параметры в файловые регистры через ISPSOft.
- Значения в файловых регистрах можно прочитать во время работы ПЛК. Обратитесь к описанию инструкции API2303 MEMW в настоящем Руководстве по программированию для получения дополнительной информации о том, как работать с данными в файловых регистрах.

---

## Глава 3 Сводная таблица инструкций

### Содержание

<b>3.1</b>	<b>Инструкции</b> .....	<b>3-2</b>
3.1.1	Базовые инструкции .....	3-2
3.1.2	Прикладные инструкции .....	3-2
<b>3.2</b>	<b>Сводная таблица инструкций</b> .....	<b>3-3</b>
3.2.1	Базовые инструкции .....	3-3
3.2.2	Прикладные инструкции (Сортировка по номерам) .....	3-4
3.2.3	Прикладные инструкции (Сортировка по алфавиту) .....	3-4
3.2.4	Таблица объектов.....	3-5
<b>3.3</b>	<b>Таблица базовых инструкций</b> .....	<b>3-6</b>
<b>3.4</b>	<b>Таблица прикладных инструкций</b> .....	<b>3-8</b>
3.4.1	Прикладные инструкции (сортировка по номерам) .....	3-8
3.4.2	Прикладные инструкции (сортировка по алфавиту) .....	3-41

## 3.1 Инструкции

Инструкции, используемые в ПЛК серии AS, включают базовые и прикладные инструкции.

### 3.1.1 Базовые инструкции

Классификация	Описание
Инструкции для контактов	Загрузка контактов, последовательное соединение контактов, параллельное соединение контактов и т. д.
Инструкции для выходов	Выход битового регистра; импульсный выход
Инструкции Мастер-контроля	Настройка и сброс Мастер-контроля
Инструкции обнаружения контакта по переднему/заднему фронту	Запуск инструкций, загружающих контакт, осуществляющих последовательное соединение контактов и параллельное соединение контактов
Инструкции выходов по переднему/заднему фронту	Выход битового регистра
Прочие инструкции	Прочие инструкции

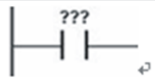

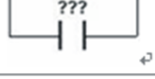
### 3.1.2 Прикладные инструкции

API	Классификация	Описание
0000~0083	Инструкции сравнения	Такие виды сравнений, как =, <>, >, >=, <, <=, и т.п.
0100~0118	Арифметические инструкции	Сложение, вычитание, умножение, деление двоичных и двоично-десятичных чисел.
0200~0217	Инструкции преобразования данных	Преобразование двоично-десятичных значений в двоичные и наоборот.
0300~0310	Инструкции передачи данных	Передача указанных данных.
0400~0402	Инструкции перехода	Программные переходы.
0500~0504	Инструкции по выполнению программы	Отключение или включение прерывания.
0600	Инструкции обновления входов/выходов	Обновление входов/выходов.
0700~0710	Сервисные инструкции	Инструкции, которые применяются к счетчикам, обучающимся таймерам, специальным таймерам и т. д.
0800~0817	Логические инструкции	Логические операции, такие как логическое сложение, логическое умножение и т. д.
0900~0904	Инструкции вращения	Поворот / смещение указанных данных
1000~1011	Инструкции таймеров и счетчиков	Инструкции таймеров и счетчиков
1100~1115	Инструкции сдвига	Сдвиг указанных данных
1200~1225	Инструкции обработки данных	Кодирование и декодирование 16-битных данных
1300~1302	Инструкции по созданию структуры	Вложенные циклы программы
1400~1401	Инструкции для модулей	Чтение данных из специальных модулей и запись данных в специальные модули

API	Классификация	Описание
1500~1517	Инструкции по операциям с числами с плавающей запятой	Операции с числами с плавающей запятой
1600~1608	Инструкции для часов реального времени	Чтение/запись, сложение/вычитание и сравнение значений времени
1700~1704	Инструкции для периферийных устройств	Работа с входами/выходами, подключенными к периферийным устройствам
1806~1817	Инструкции коммуникации	Управление периферийными устройствами посредством коммуникации
1900~1905	Прочие инструкции	Инструкции, отличные от указанных выше
2100~2119	Инструкции по обработке строк	Преобразование двоичных / двоично-десятичных чисел и кодов ASCII; преобразование между двоичными числами и строками; преобразование чисел с плавающей запятой и строк; обработка строк
2200~2210	Инструкции для Ethernet	Управление обменом данными через Ethernet
2300~2303	Инструкции для карты памяти	Чтение данных с карты памяти и запись данных на карту памяти
2400~2401	Инструкции по управлению задачами	Управление задачами в программе
2500~2502	Инструкции SFC	Управление SFC инструкциями
2700~2723	Инструкции высокоскоростных выходов	Инструкции управления высокоскоростными выходами и позиционированием
2800~2807	Инструкции коммуникации CANopen	Инструкции по обмену данными по CANopen, особенно применимы между устройствами Delta

## 3.2 Сводная таблица инструкций

### 3.2.1 Базовые инструкции

Код инструкции	Символ	Функция	Операнды
LD		Вывод контакта / подключение контакта последовательно / подключение контакта параллельно	DX · X · Y · M · S · T · C · HC · D · L · SM · PR
AND			
OR			
↓	↓	↓	↓
①	②	③	④

Описание:

- ①: Наименование инструкции
- ②: Символ на релейно-контактной схеме в ISPSOft
- ③: Функция инструкции
- ④: Операнды, поддерживаемые инструкцией

### 3.2.2 Прикладные инструкции (Сортировка по номерам)

API	Код инструкции		Импульсн. инструкция	Символ		Функция
	16-бит	32-бит				
0000	LD=	DLD=	—			Сравнение контактного типа: ВКЛ: S1 = S2 ВЫКЛ: S1 ≠ S2
0001	LD<>	DLD<>	—			Сравнение контактного типа: ВКЛ: S1 ≠ S2 ВЫКЛ: S1 = S2
0002	LD>	DLD>	—			Сравнение контактного типа: ВКЛ: S1 > S2 ВЫКЛ: S1 ≤ S2

↓
↓
↓
↓
↓
↓

Описание:

①: API номер инструкции

②: Наименование инструкции

③: Если 16-битную инструкцию можно использовать в качестве 32-битной инструкции, перед 16-битной инструкцией добавляется символ D, чтобы сформировать 32-битную.

④: Знак ✓ указывает, что инструкция может использоваться как импульсная, тогда как знак — указывает, что не может.

Если пользователь хочет использовать импульсную версию инструкции, необходимо добавить символ P в конец кода инструкции.

⑤: Символ на релейно-контактной схеме в ISPSOft

⑥: Функция

### 3.2.3 Прикладные инструкции (Сортировка по алфавиту)

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
E	0501	EI	—	—	Включение прерывания
	2208	EIPRW	—	—	Чтение и запись через Ethernet/IP
	0503	EIX	—	—	Отключение специального прерывания
	1203	ENCO	—	✓	Энкодер
	1905	EPOP	—	✓	Чтение данных из индексного регистра
	1904	EPUSH	—	✓	Сохранение в индексные регистры

↓
↓
↓
↓
↓
↓

Описание:



①: Буква алфавита, с которой начинаются коды инструкций

②: API номер инструкции

③ ~ ④ : Наименование инструкции

Если 16-битную инструкцию можно использовать в качестве 32-битной инструкции, перед 16-битной инструкцией добавляется символ D, чтобы сформировать 32-битную.

⑤ : Знак ✓ указывает, что инструкция может использоваться как импульсная, тогда как знак — указывает, что не может.

Если пользователь хочет использовать импульсную версию инструкции, необходимо добавить символ P в конец кода инструкции.

⑥: Функция

### 3.2.4 Таблица объектов

①	②	③	④															
↑	↑	↑	↑															
API	Код инструкции			Операнд											Функция			
0100	D	+	P	S <sub>1</sub> · S <sub>2</sub> · D											Сложение двоичных чисел			

⑤ ←	Объект	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
	S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
	S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		
	D		●			●	●	●	●			○	○				

⑥ ←	Тип данных	BOOL	WORD	DWORD	QWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
	S <sub>1</sub>		●	●		●	●	●				●	●	
	S <sub>2</sub>		●	●		●	●	●				●	●	
	D		●	●		●	●	●				●	●	

Импульсная инстр.	16-бит инструкция	32-бит инструкция
AS	AS	AS

⑧ ←	<table style="border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">+</td> <td style="border: 1px solid black; padding: 2px;">+P</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">En</td> <td style="border: 1px solid black; padding: 2px;">En</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">S1</td> <td style="border: 1px solid black; padding: 2px;">S1</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">S2</td> <td style="border: 1px solid black; padding: 2px;">S2</td> </tr> </table>	+	+P	En	En	S1	S1	S2	S2	S <sub>1</sub> : Слагаемое 1 S <sub>2</sub> : Слагаемое 2 D : Сумма
+	+P									
En	En									
S1	S1									
S2	S2									

→ ⑦

Описание:

①: API номер инструкции

②: Код инструкции

Если 16-битную инструкцию можно использовать в качестве 32-битной инструкции, перед 16-битной инструкцией добавляется символ D, чтобы сформировать 32-битную.

③: Операнды

④: Функция

⑤: Объекты, поддерживаемые операндом

1. Десятичный формат обозначается буквой K, десятичные значения вводятся непосредственно в ISPSoft. Например, десятичное число 30 вводится непосредственно в ISPSoft как 30.
2. Шестнадцатеричный формат обозначается символом 16 #. Например, десятичное число 30 представлено как 16 # 1E в шестнадцатеричном формате.
3. Числа с плавающей запятой обозначаются как F / DF, они представлены десятичными запятыми в ISPSoft. Например, число F500 с плавающей запятой представлено в ISPSoft как 500,0.
4. Строки (строковые значения) обозначаются символом «\$», они представлены в кавычках: «ISPSoft». Например, строка 1234 представлена в ISPSoft как «1234».
5. ○: Пустой круг  
Объект не может быть изменен с помощью индексного регистра.
6. ●: сплошной круг  
Объект не может быть изменен с помощью индексного регистра.

Ⓢ : Объекты операнда

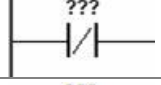
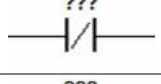
Ⓣ : Формат инструкции

Указывает, может ли инструкция использоваться в качестве импульсной, 16- или 32-битной

Ⓡ : Символ на релейно-контактной диаграмме

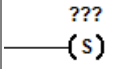

### 3.3 Таблица базовых инструкций

● Инструкции для контактов


Код инструкции	Символ	Функция	Операнды
LD		Вывод контакта A / подключение контакта A последовательно / подключение контакта A параллельно	DX, X, Y, M, SM, S, T, C, HC, D
AND			
OR			
LDI		Вывод контакта B / подключение контакта B последовательно / подключение контакта B параллельно	DX, X, Y, M, SM, S, T, C, HC, D
ANI			
ORI			

● Инструкции для выходов

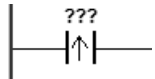

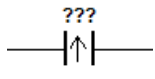
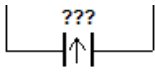
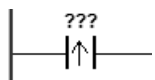

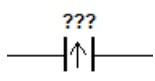
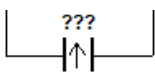
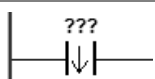

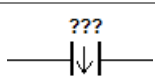
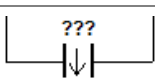
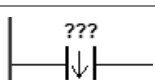

Код инструкции	Символ	Функция	Условие выполнения	Операнды
OUT		Управление катушкой		DY, Y, M, SM, S, T, C, HC, D

Код инструкции	Символ	Функция	Условие выполнения	Операнды
SET		Включение выхода с фиксацией		DY, Y, M, SM, S, T, C, HC, D

- Инструкции Мастер-контроля

Код инструкции	Символ	Функция	Операнды
MC		Настройка Мастер-контроля	N
MCR		Отмена Мастер-контроля	N

- Инструкции обнаружения контакта по переднему/заднему фронту

Код инструкции	Символ	Функция	Условие выполнения	Операнды	
LDP		Запуск обнаружения по переднему фронту / Последовательное подключение обнаружения по переднему фронту / Параллельное подключение обнаружения по переднему фронту		DX, X, Y, M, SM, S, T, C, HC, D	
ANDP					
ORP					
PED					
APED					
OPED					
LDF		Запуск обнаружения по заднему фронту / Последовательное подключение обнаружения по заднему фронту / Параллельное подключение обнаружения по заднему фронту		DX, X, Y, M, SM, S, T, C, HC, D	
ANDF					
ORF					
NED					

Код инструкции	Символ	Функция	Условие выполнения	Операнды
ANED				
ONED				

● Инструкции импульса по переднему/заднему фронту

Код инструкции	Символ	Функция	Условие выполнения	Операнды
PLS		Импульс по переднему фронту		Y, M, SM, S
PLF		Импульс по заднему фронту		Y, M, SM, S

● Прочие инструкции


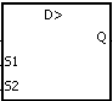
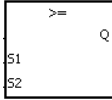
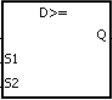
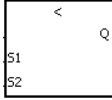
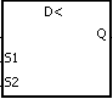
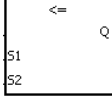
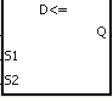
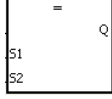
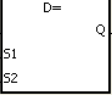
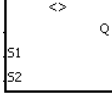
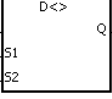

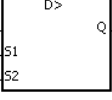
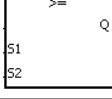
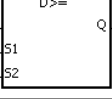
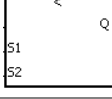
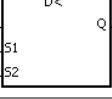
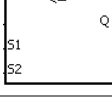
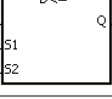
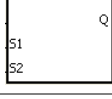
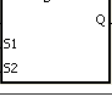
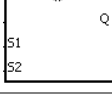
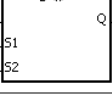

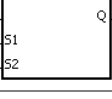

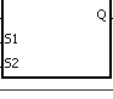
Код инструкции	Символ	Функция	Операнды
INV		Инвертирование результата логической операции	—
NP		Цель переключается по переднему фронту	—
PN		Цель переключается по заднему фронту	—
FB_NP		Цель переключается по переднему фронту	Y, M, S, D
FB_PN		Цель переключается по заднему фронту	Y, M, S, D

### 3.4 Таблица прикладных инструкций

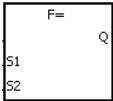
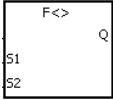
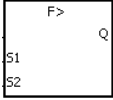
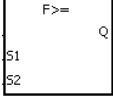
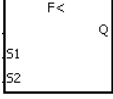
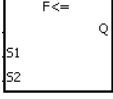
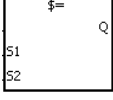
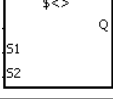
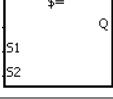
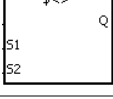
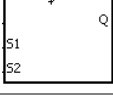
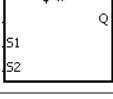
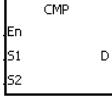
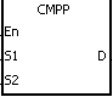
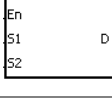
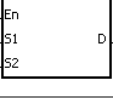
#### 3.4.1 Прикладные инструкции (сортировка по номерам)

● Инструкции сравнения

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0000	LD=	DLD=	—		Сравнение значений ON: <b>S<sub>1</sub> = S<sub>2</sub></b> OFF: <b>S<sub>1</sub> ≠ S<sub>2</sub></b>
0001	LD<>	DLD<>	—		Сравнение значений ON: <b>S<sub>1</sub> ≠ S<sub>2</sub></b> OFF: <b>S<sub>1</sub> = S<sub>2</sub></b>

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0002	LD>	DLD>	—			Сравнение значений ON: $S_1 > S_2$ OFF: $S_1 \leq S_2$
0003	LD>=	DLD>=	—			Сравнение значений ON: $S_1 \geq S_2$ OFF: $S_1 < S_2$
0004	LD<	DLD<	—			Сравнение значений ON: $S_1 < S_2$ OFF: $S_1 \geq S_2$
0005	LD<=	DLD<=	—			Сравнение значений ON: $S_1 \leq S_2$ OFF: $S_1 > S_2$
0006	AND=	DAND=	—			Сравнение значений ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0007	AND<>	DAND<>	—			Сравнение значений ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0008	AND>	DAND>	—			Сравнение значений ON: $S_1 > S_2$ OFF: $S_1 \leq S_2$
0009	AND>=	DAND>=	—			Сравнение значений ON: $S_1 \geq S_2$ OFF: $S_1 < S_2$
0010	AND<	DAND<	—			Сравнение значений ON: $S_1 < S_2$ OFF: $S_1 \geq S_2$
0011	AND<=	DAND<=	—			Сравнение значений ON: $S_1 \leq S_2$ OFF: $S_1 > S_2$
0012	OR=	DOR=	—			Сравнение значений ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0013	OR<>	DOR<>	—			Сравнение значений ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0014	OR>	DOR>	—			Сравнение значений ON: $S_1 > S_2$ OFF: $S_1 \leq S_2$
0015	OR>=	DOR>=	—			Сравнение значений ON: $S_1 \geq S_2$ OFF: $S_1 < S_2$

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0016	OR<	DOR<	—		Сравнение значений ON: $S_1 < S_2$ OFF: $S_1 \geq S_2$
0017	OR<=	DOR<=	—		Сравнение значений ON: $S_1 \leq S_2$ OFF: $S_1 > S_2$
0018	—	FLD=	—		Сравнение чисел с плавающей запятой ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0019	—	FLD<>	—		Сравнение чисел с плавающей запятой ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0020	—	FLD>	—		Сравнение чисел с плавающей запятой ON: $S_1 > S_2$ OFF: $S_1 \leq S_2$
0021	—	FLD>=	—		Сравнение чисел с плавающей запятой ON: $S_1 \geq S_2$ OFF: $S_1 < S_2$
0022	—	FLD<	—		Сравнение чисел с плавающей запятой ON: $S_1 < S_2$ OFF: $S_1 \geq S_2$
0023	—	FLD<=	—		Сравнение чисел с плавающей запятой ON: $S_1 \leq S_2$ OFF: $S_1 > S_2$
0024	—	FAND=	—		Сравнение чисел с плавающей запятой ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0025	—	FAND<>	—		Сравнение чисел с плавающей запятой ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0026	—	FAND>	—		Сравнение чисел с плавающей запятой ON: $S_1 > S_2$ OFF: $S_1 \leq S_2$
0027	—	FAND>=	—		Сравнение чисел с плавающей запятой ON: $S_1 \geq S_2$ OFF: $S_1 < S_2$
0028	—	FAND<	—		Сравнение чисел с плавающей запятой ON: $S_1 < S_2$ OFF: $S_1 \geq S_2$
0029	—	FAND<=	—		Сравнение чисел с плавающей запятой ON: $S_1 \leq S_2$ OFF: $S_1 > S_2$

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0030	—	FOR=	—		Сравнение чисел с плавающей запятой ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0031	—	FOR<>	—		Сравнение чисел с плавающей запятой ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0032	—	FOR>	—		Сравнение чисел с плавающей запятой ON: $S_1 > S_2$ OFF: $S_1 \leq S_2$
0033	—	FOR>=	—		Сравнение чисел с плавающей запятой ON: $S_1 \geq S_2$ OFF: $S_1 < S_2$
0034	—	FOR<	—		Сравнение чисел с плавающей запятой ON: $S_1 < S_2$ OFF: $S_1 \geq S_2$
0035	—	FOR<=	—		Сравнение чисел с плавающей запятой ON: $S_1 \leq S_2$ OFF: $S_1 > S_2$
0036	LD\$=	—	—		Сравнение строковых переменных ON: $S_1 = S_2$ ON: $S_1 \neq S_2$
0037	LD\$<>	—	—		Сравнение строковых переменных ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0042	AND\$=	—	—		Сравнение строковых переменных ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0043	AND\$<>	—	—		Сравнение строковых переменных ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0048	OR\$=	—	—		Сравнение строковых переменных ON: $S_1 = S_2$ OFF: $S_1 \neq S_2$
0049	OR\$<>	—	—		Сравнение строковых переменных ON: $S_1 \neq S_2$ OFF: $S_1 = S_2$
0054	CMP	DCMP	✓	   	Сравнение значений

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0055	ZCP	DZCP	✓			Зонное сравнение
0056	—	FCMP	✓			Сравнение чисел с плавающей запятой
0057	—	FZCP	✓			Зонное сравнение с плавающей запятой
0058	MCMP	—	✓			Матричное сравнение
0059	CMPT=	—	✓			Сравнение таблиц ON: =
0060	CMPT<>	—	✓			Сравнение таблиц ON: ≠
0061	CMPT>	—	✓			Сравнение таблиц ON: >
0062	CMPT>=	—	✓			Сравнение таблиц ON: ≥
0063	CMPT<	—	✓			Сравнение таблиц ON: <
0064	CMPT<=	—	✓			Сравнение таблиц ON: ≤
0065	CHKADR	—	—			Проверка адреса контактного типа регистра указателя
0066	LDZ=	DLDZ=	—			Сравнение абсолютных значений контактного типа ON:   S <sub>1</sub> - S <sub>2</sub>   =   S <sub>3</sub>   OFF:   S <sub>1</sub> - S <sub>2</sub>   ≠   S <sub>3</sub>



API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0067	LDZ<>	DLNZ<>	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \neq  S_3 $ OFF: $ S_1 - S_2  =  S_3 $
0068	LDZ>	DLNZ>	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  >  S_3 $ OFF: $ S_1 - S_2  \leq  S_3 $
0069	LDZ>=	DLNZ>=	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \geq  S_3 $ OFF: $ S_1 - S_2  <  S_3 $
0070	LDZ<	DLNZ<	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  <  S_3 $ OFF: $ S_1 - S_2  \geq  S_3 $
0071	LDZ<=	DLNZ<=	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \leq  S_3 $ OFF: $ S_1 - S_2  >  S_3 $
0072	ANDZ=	DANDZ=	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  =  S_3 $ OFF: $ S_1 - S_2  \neq  S_3 $
0073	ANDZ<>	DANDZ<>	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \neq  S_3 $ OFF: $ S_1 - S_2  =  S_3 $
0074	ANDZ>	DANDZ>	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  >  S_3 $ OFF: $ S_1 - S_2  \leq  S_3 $
0075	ANDZ>=	DANDZ>=	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \geq  S_3 $ OFF: $ S_1 - S_2  <  S_3 $
0076	ANDZ<	DANDZ<	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  <  S_3 $ OFF: $ S_1 - S_2  \geq  S_3 $
0077	ANDZ<=	DANDZ<=	—		Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \leq  S_3 $ OFF: $ S_1 - S_2  >  S_3 $

3

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0078	ORZ=	DORZ=	—			Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  =  S_3 $ OFF: $ S_1 - S_2  \neq  S_3 $
0079	ORZ<>	DORZ<>	—			Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \neq  S_3 $ OFF: $ S_1 - S_2  =  S_3 $
0080	ORZ>	DORZ>	—			Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  >  S_3 $ OFF: $ S_1 - S_2  \leq  S_3 $
0081	ORZ>=	DORZ>=	—			Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \geq  S_3 $ OFF: $ S_1 - S_2  <  S_3 $
0082	ORZ<	DORZ<	—			Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  <  S_3 $ OFF: $ S_1 - S_2  \geq  S_3 $
0083	ORZ<=	DORZ<=	—			Сравнение абсолютных значений контактного типа ON: $ S_1 - S_2  \leq  S_3 $ OFF: $ S_1 - S_2  >  S_3 $

● Арифметические инструкции

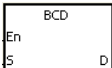

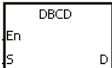



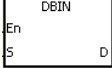





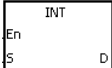
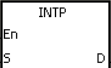
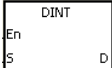
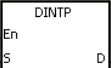
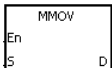
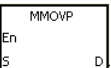
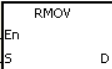



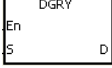





API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0100	+	D+	✓			Сложение двоичных чисел $S_1 + S_2 = D$
0101	-	D-	✓			Вычитание двоичных чисел $S_1 - S_2 = D$
0102	*	D*	✓			Умножение двоичных чисел $S_1 * S_2 = D$

API	Код инструкции		Импульсная инструкция	Символ		Функционал																															
	16 бит	32 бит																																			
0103	/	D/	✓	<table border="1"> <tr><td>En</td><td>/</td><td>En</td><td>/P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>D/</td><td>En</td><td>D/P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table>	En	/	En	/P	S1	D	S1	D	S2		S2		En	D/	En	D/P	S1	D	S1	D	S2		S2		Деление двоичных чисел $S_1/S_2=D$								
En	/	En	/P																																		
S1	D	S1	D																																		
S2		S2																																			
En	D/	En	D/P																																		
S1	D	S1	D																																		
S2		S2																																			
0104	—	F+	✓	<table border="1"> <tr><td>En</td><td>F+</td><td>En</td><td>F+P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table>	En	F+	En	F+P	S1	D	S1	D	S2		S2		Сложение чисел с плавающей запятой $S_1+S_2=D$																				
En	F+	En	F+P																																		
S1	D	S1	D																																		
S2		S2																																			
0105	—	F-	✓	<table border="1"> <tr><td>En</td><td>F-</td><td>En</td><td>F-P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table>	En	F-	En	F-P	S1	D	S1	D	S2		S2		Вычитание чисел с плавающей запятой $S_1-S_2=D$																				
En	F-	En	F-P																																		
S1	D	S1	D																																		
S2		S2																																			
0106	—	F*	✓	<table border="1"> <tr><td>En</td><td>F*</td><td>En</td><td>F*P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table>	En	F*	En	F*P	S1	D	S1	D	S2		S2		Умножение чисел с плавающей запятой $S_1*S_2=D$																				
En	F*	En	F*P																																		
S1	D	S1	D																																		
S2		S2																																			
0107	—	F/	✓	<table border="1"> <tr><td>En</td><td>F/</td><td>En</td><td>F/P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table>	En	F/	En	F/P	S1	D	S1	D	S2		S2		Деление чисел с плавающей запятой $S_1/S_2=D$																				
En	F/	En	F/P																																		
S1	D	S1	D																																		
S2		S2																																			
0112	BK+	DBK+	✓	<table border="1"> <tr><td>En</td><td>BK+</td><td>En</td><td>BK+P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> <tr><td>n</td><td></td><td>n</td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DBK+</td><td>En</td><td>DBK+P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> <tr><td>n</td><td></td><td>n</td><td></td></tr> </table>	En	BK+	En	BK+P	S1	D	S1	D	S2		S2		n		n		En	DBK+	En	DBK+P	S1	D	S1	D	S2		S2		n		n		Сложение двоичных чисел в блоках
En	BK+	En	BK+P																																		
S1	D	S1	D																																		
S2		S2																																			
n		n																																			
En	DBK+	En	DBK+P																																		
S1	D	S1	D																																		
S2		S2																																			
n		n																																			
0113	BK-	DBK-	✓	<table border="1"> <tr><td>En</td><td>BK-</td><td>En</td><td>BK-P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> <tr><td>n</td><td></td><td>n</td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DBK-</td><td>En</td><td>DBK-P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> <tr><td>n</td><td></td><td>n</td><td></td></tr> </table>	En	BK-	En	BK-P	S1	D	S1	D	S2		S2		n		n		En	DBK-	En	DBK-P	S1	D	S1	D	S2		S2		n		n		Вычитание двоичных чисел в блоках
En	BK-	En	BK-P																																		
S1	D	S1	D																																		
S2		S2																																			
n		n																																			
En	DBK-	En	DBK-P																																		
S1	D	S1	D																																		
S2		S2																																			
n		n																																			
0114	\$+	—	✓	<table border="1"> <tr><td>En</td><td>\$+</td><td>En</td><td>\$+P</td></tr> <tr><td>S1</td><td>D</td><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td><td>S2</td><td></td></tr> </table>	En	\$+	En	\$+P	S1	D	S1	D	S2		S2		Связывание строковых переменных																				
En	\$+	En	\$+P																																		
S1	D	S1	D																																		
S2		S2																																			
0115	INC	DINC	✓	<table border="1"> <tr><td>En</td><td>INC</td><td>En</td><td>INCP</td></tr> <tr><td>D</td><td></td><td>D</td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DINC</td><td>En</td><td>DINCP</td></tr> <tr><td>D</td><td></td><td>D</td><td></td></tr> </table>	En	INC	En	INCP	D		D		En	DINC	En	DINCP	D		D		Добавление единицы к двоичному числу																
En	INC	En	INCP																																		
D		D																																			
En	DINC	En	DINCP																																		
D		D																																			
0116	DEC	DDEC	✓	<table border="1"> <tr><td>En</td><td>DEC</td><td>En</td><td>DECP</td></tr> <tr><td>D</td><td></td><td>D</td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DDEC</td><td>En</td><td>DDECP</td></tr> <tr><td>D</td><td></td><td>D</td><td></td></tr> </table>	En	DEC	En	DECP	D		D		En	DDEC	En	DDECP	D		D		Вычитание единицы из двоичного числа																
En	DEC	En	DECP																																		
D		D																																			
En	DDEC	En	DDECP																																		
D		D																																			

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0117	MUL16	MUL32	✓	 	 	Умножение двоичных чисел для 16 бит Умножение двоичных чисел для 32 бит
0118	DIV16	DIV32	✓	 	 	Деление двоичных чисел для 16 бит Деление двоичных чисел для 32 бит

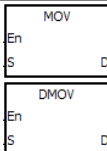
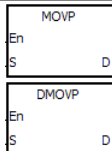
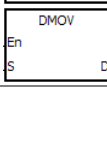
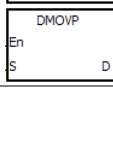
3

● Инструкции преобразования данных

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0200	BCD	DBCD	✓	 	 	Преобразование двоичного числа в двоично-десятичное число
0201	BIN	DBIN	✓	 	 	Преобразование двоично-десятичного числа в двоичное число
0202	FLT	DFLT	✓	 	 	Преобразование двоичного целого числа в двоичное число с плавающей запятой
0204	INT	DINT	✓	 	 	Преобразование 32-битового числа с плавающей запятой в двоичное целое число
0206	MMOV	—	✓	 	Преобразование 16-битового в 32-битовое значение	
0207	RMOV	—	✓	 	Преобразование 32-битового в 16-битовое значение	
0208	GRY	DGRY	✓	 	 	Преобразование двоичного числа в код Грея
0209	GBIN	DGBIN	✓	 	 	Преобразование кода Грея в двоичное число

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0210	NEG	DNEG	✓			Второе дополнение
0211	—	FNEG	✓			Изменение знака 32-битового числа с плавающей запятой
0212	—	FBCD	✓			Преобразование двоичного числа с плавающей запятой в десятичное число с плавающей запятой
0213	—	FBIN	✓			Преобразование десятичного числа с плавающей запятой в двоичное число с плавающей запятой
0214	BKBCD	—	✓			Преобразование двоичных чисел в блоке в десятичные числа в блоке
0215	BKBIN	—	✓			Преобразование двоичных чисел в блоке в двоично-десятичные числа в блоке
0216	SCAL	DSCAL	✓			Операция масштабирования значений
0217	SCLP	DSCLP	✓			Тип параметра операции масштабирования значений
0222	SCLM	DSCLM	✓			Операция масштабирования многоточечной области

- Инструкции передачи данных

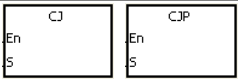


API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0300	MOV	DMOV	✓			Передача данных S: Источник данных D: Место назначения данных
						

3

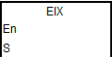
API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0302	\$MOV	—	✓			Передача строковой переменной
0303	CML	DCML	✓	 	 	Инвертирование данных
0304	BMOV	DBMOV	✓	 	 	Передача всех данных
0305	NMOV	DNMOV	✓	 	 	Перенос данных на несколько операндов
0306	XCH	DXCH	✓	 	 	Обмен данными
0307	BXCH	—	✓			Обмен всеми данными
0308	SWAP	DSWAP	✓	 	 	Замена старшего байта младшим байтом
0309	SMOV	—	✓			Передача цифр
0310	MOVB	—	✓			Передача нескольких бит

● Инструкции перехода

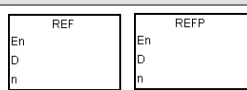
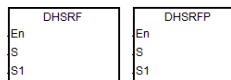
API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0400	CJ	—	✓		Условный переход
0401	JMP	—	—		Безусловный переход
0402	GOEND	—	—		Переход к END

- Инструкции по выполнению программы

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0500	DI	—	—		Отключение прерывания
0501	EI	—	—		Включение прерывания
0503	EIX	—	—		Отключение специального прерывания
0504	DIX	—	—		Включение специального прерывания

- Инструкции обновления входов/выходов

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0600	REF	—	✓		Обновление входов/выходов
0601	—	DHSRF	✓		Обновление значений высокоскоростного сравнения

- Сервисные инструкции

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0700	ALT	—	✓		Альтернативное переключение между ВКЛ и ВЫКЛ
0701	TTMR	—	—		Обучающийся таймер
0702	STMR	—	—		Специальный таймер
0703	RAMP	DRAMP	—		Сигнал циклического скачка

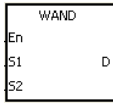
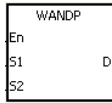
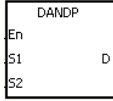
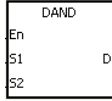
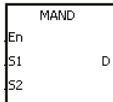
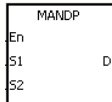
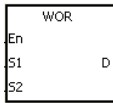
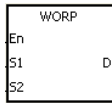

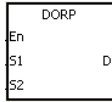
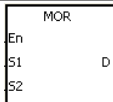
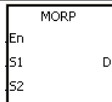
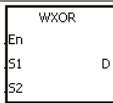
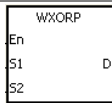

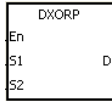
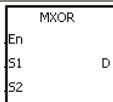
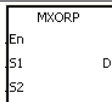
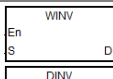
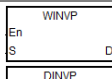
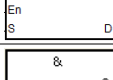
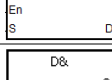

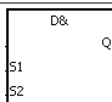

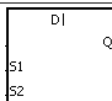
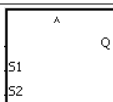
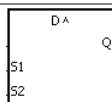

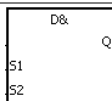
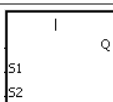
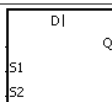
3

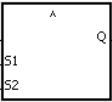
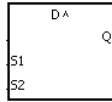
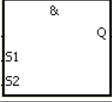

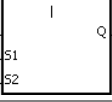
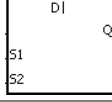
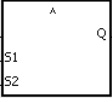
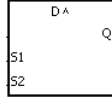
API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
0704	MTR	—	—		Матричный вход
0705	ABSD	DABSD	—		Абсолютный барабанный секвенсор
0706	INCD	—	—		Инкрементный барабанный секвенсор
0708	—	DPIDE	—		ПИД-алгоритм
0709	XCMP	—	—		Настройка сравнения входов нескольких рабочих станций
0710	YOUT	—	—		Сравнение результатов работы нескольких рабочих станций
0711	SUNRS	—	—		Задание времени восхода и захода солнца

● Логические инструкции

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			

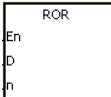
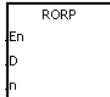
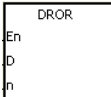
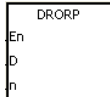
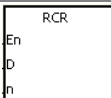
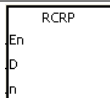
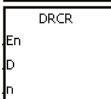

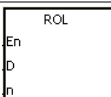
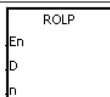
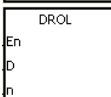
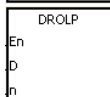
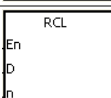
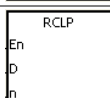
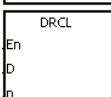
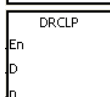

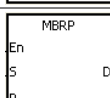


API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0800	WAND	DAND	✓	   	Логическое И	
0801	MAND	—	✓	 	Матричное логическое И	
0802	WOR	DOR	✓	   	Логическое ИЛИ	
0803	MOR	—	✓	 	Матричное ИЛИ	
0804	WXOR	DXOR	✓	   	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ	
0805	MXOR	—	✓	 	Матричное ИСКЛЮЧАЮЩЕЕ ИЛИ	
0808	WINV	DINV	✓	   	Логическое инвертирование INV	
0809	LD&	DLD&	—	 	$S_1 \& S_2$	
0810	LD	DLD	—	 	$S_1   S_2$	
0811	LD^	DLD^	—	 	$S_1 \wedge S_2$	
0812	AND&	DAND&	—	 	$S_1 \& S_2$	
0813	AND	DAND	—	 	$S_1   S_2$	

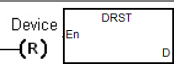
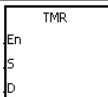
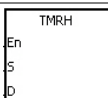
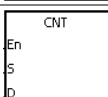
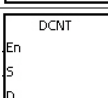
API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0814	AND^	DAND^	—			$S_1 \wedge S_2$
0815	OR&	DOR&	—			$S_1 \& S_2$
0816	OR	DOR	—			$S_1   S_2$
0817	OR^	DOR^	—			$S_1 \wedge S_2$

3

● Инструкции вращения

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
0900	ROR	DROR	✓	   	Вращение вправо	
0901	RCR	DRCR	✓	   	Вращение вправо с флагом переноса	
0902	ROL	DROL	✓	   	Вращение влево	
0903	RCL	DRCL	✓	   	Вращение влево с флагом переноса	
0904	MBR	—	✓	 	Вращение битов матрицы	

● Инструкции таймеров и счетчиков

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1000	RST	DRST	—		Сброс контакта на ВЫКЛ или очистка значения в регистре	
1001	TMR	—	—		16-битовый таймер (шаг: 100 мс)	
1002	TMRH	—	—		16-битовый таймер (шаг: 1 мс)	
1003	CNT	—	—		16-битовый счетчик	
1004	—	DCNT	—		32-битовый счетчик (включает использование высокоскоростного счетчика)	

3

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1005	—	DHSCS	—		Настройка высокоскоростного сравнения
1006	—	DHSCR	—		Сброс высокоскоростного сравнения
1007	—	DHSZ	—		Зонное сравнение высокоскоростных входов
1008	—	DSPD	—		Измерение скорости
1009	PWD	—	—		Измерение ширины импульса
1010	—	DCAP	—		Захват высокоскоростного значения подсчета при внешнем входном прерывании
1011	TMRM	—	—		16-битовый таймер (шаг: 10 мс)
1012	IETS	—	✓		Старт инструкции измерения времени
1013	IETE	—	✓		Завершение инструкции измерения времени

● Инструкции сдвига

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1100	SFTR	—	✓			Сдвиг состояния битовых регистров вправо
1101	SFTL	—	✓			Сдвиг состояния битовых регистров влево
1102	WSFR	—	✓			Сдвиг данных в словных регистрах вправо
1103	WSFL	—	✓			Сдвиг данных в словных регистрах влево
1104	SFWR	—	✓			Сдвиг данных и запись их в словный регистр

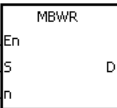
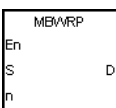

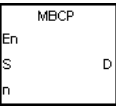
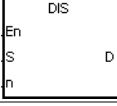
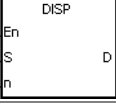
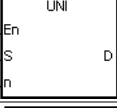
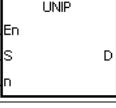

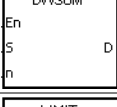
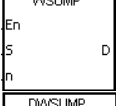
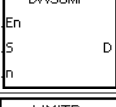
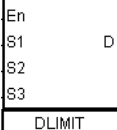
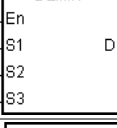
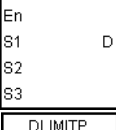
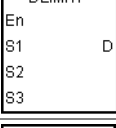
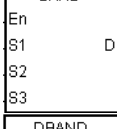
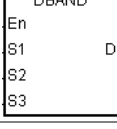
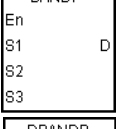
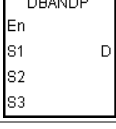
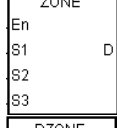
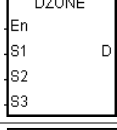
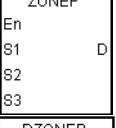
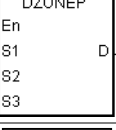




API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1105	SFRD	—	✓			Сдвиг данных и чтение их из словного регистра
1106	SFPO	—	✓			Чтение последних данных из списка данных
1107	SFDEL	—	✓			Удаление данных из списка данных
1108	SFINS	—	✓			Добавление данных в список данных
1109	MBS	—	✓			Сдвиг битов матрицы
1110	SFR	—	✓			Сдвиг значений битов в 16-битовых регистрах на n бит вправо
1111	SFL	—	✓			Сдвиг значений битов в 16-битовых регистрах на n бит влево
1112	BSFR	—	✓			Сдвиг состояний n битовых регистров на один бит вправо
1113	BSFL	—	✓			Сдвиг состояний n битовых регистров на один бит влево
1114	NSFR	—	✓			Сдвиг n регистров вправо
1115	NSFL	—	✓			Сдвиг n регистров влево

- Инструкции обработки данных

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1200	SER	DSER	✓			Поиск данных
1201	SUM	DSUM	✓			Количество включенных битов

3

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1202	DECO	—	✓			Декодировщик
1203	ENCO	—	✓			Кодировщик
1204	SEGD	—	✓			Семисегментное декодирование
1205	SORT	DSORT	✓	 	 	Сортировка данных
1206	ZRST	—	✓			Очистка зоны
1207	BON	DBON	✓	 	 	Проверка состояния бита
1208	MEAN	DMEAN	✓	 	 	Среднее значение
1209	CCD	—	✓			Проверка суммы
1210	ABS	DABS	✓	 	 	Абсолютное значение
1211	MINV	—	✓			Инвертирование битов матрицы
1212	MBRD	—	✓			Чтение битов матрицы

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1213	MBWR	—	✓			Запись битов матрицы
1214	MBC	—	✓			Подсчет битов со значением 0 или 1
1215	DIS	—	✓			Разделение 16-битных данных
1216	UNI	—	✓			Объединение 16-битных данных
1217	WSUM	DWSUM	✓	 	 	Получение суммы
1221	LIMIT	DLIMIT	✓	 	 	Ограничение значения в пределах допустимого диапазона
1222	BAND	DBAND	✓	 	 	Контроль нечувствительности
1223	ZONE	DZONE	✓	 	 	Управление зоной
1224	—	FMEAN	✓			Среднее значение чисел с плавающей точкой
1225	—	FSUM	✓			Сумма чисел с плавающей точкой

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1226	—	DTM	✓		Передача и перемещение данных

● Инструкции по созданию структуры

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16-bit	32-bit			
1300	FOR	—	—		Начало вложенного цикла
1301	NEXT	—	—		Конец вложенного цикла
1302	BREAK	—	—		Завершение цикла FOR-NEXT

● Инструкции для модулей

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1400	FROM	DFROM	✓		Чтение данных из регистра управления в специальном модуле
1401	TO	DTO	✓		Запись данных в регистр управления в специальном модуле
1402	PUCONF	—	✓		Настройка параметров управления выходами модуля PU



API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1403	PUSTAT	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           PUSTAT            En            Module C_Pos            Axis Execute            ZeroS Pause                      Error                      ErrCode         </div>	Чтение состояния выходов модуля PU
1404	—	DPUPLS	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           DPUPLS            En            Module Down            Axis Error            TarPulse ErrCode            TarSpeed         </div>	Импульсный выход модуля PU (без разгона)
1405	—	DPUDRI	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           DPUDRI            En            Module Down            Axis Error            RTarPosi ErrCode            TarSpeed         </div>	Относительная выходная позиция модуля PU (с разгоном и замедлением)
1406	—	DPUDRA	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           DPUDRA            En            Module Down            Axis Error            ATarPosi ErrCode            TarSpeed         </div>	Абсолютный адресный выход модуля PU (с разгоном и замедлением)
1407	PUZRN	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           PUZRN            En            Module Down            Axis Error            Mode ErrCode            TarSpeed            JogSpeed         </div>	Возврат в нулевую точку для модуля PU
1408	PUJOG	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           PUJOG            En            Module Busy            Axis Error            JogSpeed ErrCode         </div>	Выход JOG модуля PU
1409	—	DPUMPG	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           DPUMPG            En            Module OPulse            Axis OSpeed            InMode Error            InPulse ErrCode            InSpeed            Rate         </div>	Выход MPG модуля PU
1410	—	DPUCNT	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">           DPUCNT            En            Module InPulse            InMode InSpeed            Period Error            ZeroS ErrCode         </div>	Функция высокоскоростного счетчика модуля PU

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1415	LCCAL	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     LCCAL                      En                      Group CPoint                      Module Down                      ChNo ADown                      Trigger Error                      TPoint ErrCode                      TWeight                 </div>	Калибровка канала модуля LC (тензомодуля)
1416	LCWEI	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     LCWEI                      En                      Group Weight                      Module Status                      ChNo Error                      Stable ErrCode                      ZeroS                      TareS                      TareW                 </div>	Чтение значения массы через модуль LC (тензомодуль)

3

● Инструкции по операциям с числами с плавающей запятой

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1500	—	FSIN	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FSIN                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FSINP                      En                      S D                 </div>	Синус числа с плавающей запятой
1501	—	FCOS	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FCOS                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FCOSP                      En                      S D                 </div>	Косинус числа с плавающей запятой
1502	—	FTAN	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FTAN                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FTANP                      En                      S D                 </div>	Тангенс числа с плавающей запятой
1503	—	FASIN	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FASIN                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FASINP                      En                      S D                 </div>	Арксинус числа с плавающей запятой
1504	—	FACOS	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FACOS                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FACOSP                      En                      S D                 </div>	Арккосинус числа с плавающей запятой
1505	—	FATAN	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FATAN                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FATANP                      En                      S D                 </div>	Арктангенс числа с плавающей запятой
1506	—	FSINH	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FSINH                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FSINH P                      En                      S D                 </div>	Гиперболический синус числа с плавающей запятой
1507	—	FCOSH	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FCOSH                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FCOSH P                      En                      S D                 </div>	Гиперболический косинус числа с плавающей запятой
1508	—	FTANH	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FTANH                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FTANH P                      En                      S D                 </div>	Гиперболический тангенс числа с плавающей запятой
1509	—	FRAD	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FRAD                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FRAD P                      En                      S D                 </div>	Преобразование градусов в радианы
1510	—	FDEG	✓	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FDEG                      En                      S D                 </div>	<div style="display: inline-block; border: 1px solid black; padding: 2px;">                     FDEG P                      En                      S D                 </div>	Преобразование радиан в градусы

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1511	SQR	DSQR	✓			Квадратный корень двоичного числа
1512	—	FSQR	✓			Квадратный корень числа с плавающей запятой
1513	—	FEXP	✓			Экспонента числа с плавающей запятой
1514	—	FLOG	✓			Логарифм числа с плавающей запятой
1515	—	FLN	✓			Натуральный логарифм числа с плавающей запятой
1516	—	FPOW	✓			Возведение в степень числа с плавающей запятой
1517	RAND	—	✓			Случайное число

- Инструкции для часов реального времени

API	Код инструкции		Импульсная инструкция	Символ		Функционал
	16 бит	32 бит				
1600	TRD	—	✓			Считывание времени
1601	TWR	—	✓			Запись времени
1602	T+	—	✓			Сложение времени
1603	T-	—	✓			Вычитание времени
1604	HOUR	—	—			Счетчик времени работы
1605	TCMP	—	✓			Сравнение времени

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1606	TZCP	—	✓		Сравнение временных зон
1607	DST	—	✓		Летнее время
1608	WWON	—	—		Еженедельная настройка рабочего времени

● Инструкции для периферийных устройств

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1700	TKY	DTKY	—		Десятикнопочная клавиатура
1701	HKY	DHKY	—		Шестнадцати кнопочная клавиатура
1702	DSW	—	—		DIP-переключатель
1703	ARWS	—	—		Клавиши со стрелками
1704	SEGL	—	—		Семи сегментный дисплей с фиксацией

● Инструкции коммуникации

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1806	LRC	—	—		Продольная проверка четности
1807	CRC	—	—		Циклическая проверка избыточности

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1808	MODRW	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> MODRW  En  S1  S2  S3  S  n </div>	Чтение/запись данных MODBUS
1812	COMRS	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> COMRS  En  S1 D1  S2 D2  S3 </div>	Автоопределение параметров отправки и приема данных коммуникации
1813	COMDF	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px; width: 45%;"> COMDF  En  S1  S2  S3  S4  S5  D </div> <div style="border: 1px solid black; padding: 2px; width: 45%;"> COMDFP  En  S1  S2  S3  S4  S5  D </div> </div>	Настройка коммуникации для последовательного коммуникационного порта
1814	VFDRW	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> VFDRW  En  S1  S2  S3  S </div>	Инструкция по последовательной коммуникации для ПЧ Delta
1815	ASDRW	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> ASDRW  En  S1  S2  S3  S </div>	Инструкция по последовательной коммуникации для сервопривода Delta
1816	CCONF	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px; width: 45%;"> CCONF  En  S1  S2  S3  S4  S5  S6  S7  S8  S9  S10  S11 </div> <div style="border: 1px solid black; padding: 2px; width: 45%;"> CCONFP  En  S1  S2  S3  S4  S5  S6  S7  S8  S9  S10  S11 </div> </div>	Установка параметров в таблице обмена данными коммуникационного порта
1817	MODRWE	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> MODRWE  En  S1 D  S2  S3  S4  S  n </div>	Чтение и запись данных Modbus без использования какого-либо флага
1818	DNETRW	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> DNETRW  En  S1 D1  S2 D2  S3 D3  S4 D4  S5 D5  S6  S7  S8  S9  S10 </div>	Чтение и запись данных коммуникации DeviceNet
1819	CANRS	—	—	<div style="border: 1px solid black; padding: 2px; width: fit-content;"> CANRS  En  S1 D1  S2 D2  S3 D3  S4 </div>	Пользовательская настройка коммуникации по CAN, прием и отправка данных

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1820	DMVSH	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     DMVSH                      En                      Mode Shoot1                      Start1 Shoot2                      Start2 RdData                      Ready OK                      ComNo Err                      Id_Ip ErrCode                      Address                      Length                 </div>	Разрешение обнаружения и коммуникации Delta DMV

● Прочие инструкции

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
1900	WDT	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En WDT</div> <div style="border: 1px solid black; padding: 2px;">En WDT P</div> </div>	Сторожевой таймер
1901	DELAY	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S DELAY</div> <div style="border: 1px solid black; padding: 2px;">En S DELAY P</div> </div>	Задержка выполнения программы
1902	GPWM	—	—	<div style="border: 1px solid black; padding: 5px; width: fit-content;">                     GPWM                      En                      S1 S2                      S2                 </div>	Общая ШИМ
1904	EPUSH	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En D EPUSH</div> <div style="border: 1px solid black; padding: 2px;">En D EPUSH P</div> </div>	Сохранение содержимого индексных регистров
1905	EPOP	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En D EPOP</div> <div style="border: 1px solid black; padding: 2px;">En D EPOP P</div> </div>	Считывание данных в индексные регистры
1906	INFO	—	✓	<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D INFO</div> <div style="border: 1px solid black; padding: 2px;">En S D INFO P</div> </div>	Считывание системных данных

● Инструкции по обработке строковых переменных

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
2100	BINDA	DBINDA	✓	<div style="display: flex; flex-direction: column; justify-content: space-around;"> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D BINDA</div> <div style="border: 1px solid black; padding: 2px;">En S D BINDA P</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D DBINDA</div> <div style="border: 1px solid black; padding: 2px;">En S D DBINDA P</div> </div> </div>	Преобразование десятичного числа со знаком в код ASCII
2101	BINHA	DBINHA	✓	<div style="display: flex; flex-direction: column; justify-content: space-around;"> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D BINHA</div> <div style="border: 1px solid black; padding: 2px;">En S D BINHA P</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D DBINHA</div> <div style="border: 1px solid black; padding: 2px;">En S D DBINHA P</div> </div> </div>	Преобразование шестнадцатеричного числа в код ASCII
2102	BCDDA	DBCDDA	✓	<div style="display: flex; flex-direction: column; justify-content: space-around;"> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D BCDDA</div> <div style="border: 1px solid black; padding: 2px;">En S D BCDDA P</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D DBCDDA</div> <div style="border: 1px solid black; padding: 2px;">En S D DBCDDA P</div> </div> </div>	Преобразование двоично-десятичного числа в код ASCII
2103	DABIN	DDABIN	✓	<div style="display: flex; flex-direction: column; justify-content: space-around;"> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D DABIN</div> <div style="border: 1px solid black; padding: 2px;">En S D DABIN P</div> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">En S D DDABIN</div> <div style="border: 1px solid black; padding: 2px;">En S D DDABIN P</div> </div> </div>	Преобразование двоично-десятичного кода ASCII со знаком в двоично-десятичное число со знаком

API	Код инструкции		Импульсная инструкция	Символ		Функционал																													
	16 бит	32 бит																																	
2104	HABIN	DHABIN	✓	<table border="1"> <tr><td>En</td><td>HABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>HABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DHABIN</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DHABINP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	HABIN	D	S			En	HABINP	D	S			En	DHABIN	D	S			En	DHABINP	D	S			Преобразование кода ASCII в шестнадцатеричное число						
En	HABIN	D																																	
S																																			
En	HABINP	D																																	
S																																			
En	DHABIN	D																																	
S																																			
En	DHABINP	D																																	
S																																			
2105	DABCD	DDABCD	✓	<table border="1"> <tr><td>En</td><td>DABCD</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DABCDP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABCD</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>DDABCDP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	DABCD	D	S			En	DABCDP	D	S			En	DDABCD	D	S			En	DDABCDP	D	S			Преобразование кода ASCII в двоично-десятичное число						
En	DABCD	D																																	
S																																			
En	DABCDP	D																																	
S																																			
En	DDABCD	D																																	
S																																			
En	DDABCDP	D																																	
S																																			
2106	\$LEN	—	✓	<table border="1"> <tr><td>En</td><td>\$LEN</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$LENP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$LEN	D	S			En	\$LENP	D	S			Подсчет длины строковой переменной																		
En	\$LEN	D																																	
S																																			
En	\$LENP	D																																	
S																																			
2109	\$FSTR	—	✓	<table border="1"> <tr><td>En</td><td>\$FSTR</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$FSTRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$FSTR	D	S1			S2			En	\$FSTRP	D	S1			S2			Преобразование числа с плавающей точкой в строковую переменную												
En	\$FSTR	D																																	
S1																																			
S2																																			
En	\$FSTRP	D																																	
S1																																			
S2																																			
2110	\$FVAL	—	✓	<table border="1"> <tr><td>En</td><td>\$FVAL</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$FVALP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$FVAL	D	S			En	\$FVALP	D	S			Преобразование строковой переменной в число с плавающей точкой																		
En	\$FVAL	D																																	
S																																			
En	\$FVALP	D																																	
S																																			
2111	\$RIGHT	—	✓	<table border="1"> <tr><td>En</td><td>\$RIGHT</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$RIGHTP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table>	En	\$RIGHT	D	S			n			En	\$RIGHTP	D	S			n			Извлечение символов в строковой переменной начинается справа.												
En	\$RIGHT	D																																	
S																																			
n																																			
En	\$RIGHTP	D																																	
S																																			
n																																			
2112	\$LEFT	—	✓	<table border="1"> <tr><td>En</td><td>\$LEFT</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$LEFTP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> <tr><td>n</td><td></td><td></td></tr> </table>	En	\$LEFT	D	S			n			En	\$LEFTP	D	S			n			Извлечение символов в строковой переменной начинается слева.												
En	\$LEFT	D																																	
S																																			
n																																			
En	\$LEFTP	D																																	
S																																			
n																																			
2113	\$MIDR	—	✓	<table border="1"> <tr><td>En</td><td>\$MIDR</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$MIDRP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> </table>	En	\$MIDR	D	S1			S2			En	\$MIDRP	D	S1			S2			Получение строковой переменной												
En	\$MIDR	D																																	
S1																																			
S2																																			
En	\$MIDRP	D																																	
S1																																			
S2																																			
2115	\$SER	—	✓	<table border="1"> <tr><td>En</td><td>\$SER</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>N</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$SERP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>N</td><td></td><td></td></tr> </table>	En	\$SER	D	S1			S2			N			En	\$SERP	D	S1			S2			N			Поиск строковой переменной						
En	\$SER	D																																	
S1																																			
S2																																			
N																																			
En	\$SERP	D																																	
S1																																			
S2																																			
N																																			
2116	\$RPLC	—	✓	<table border="1"> <tr><td>En</td><td>\$RPLC</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> <tr><td>S4</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$RPLCP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> <tr><td>S4</td><td></td><td></td></tr> </table>	En	\$RPLC	D	S1			S2			S3			S4			En	\$RPLCP	D	S1			S2			S3			S4			Замена символов в строковой переменной
En	\$RPLC	D																																	
S1																																			
S2																																			
S3																																			
S4																																			
En	\$RPLCP	D																																	
S1																																			
S2																																			
S3																																			
S4																																			
2117	\$DEL	—	✓	<table border="1"> <tr><td>En</td><td>\$DEL</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$DELP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> </table>	En	\$DEL	D	S1			S2			S3			En	\$DELP	D	S1			S2			S3			Удаление символов в строковой переменной						
En	\$DEL	D																																	
S1																																			
S2																																			
S3																																			
En	\$DELP	D																																	
S1																																			
S2																																			
S3																																			
2118	\$CLR	—	✓	<table border="1"> <tr><td>En</td><td>\$CLR</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$CLRCP</td><td>D</td></tr> <tr><td>S</td><td></td><td></td></tr> </table>	En	\$CLR	D	S			En	\$CLRCP	D	S			Очистка строковой переменной																		
En	\$CLR	D																																	
S																																			
En	\$CLRCP	D																																	
S																																			
2119	\$INS	—	✓	<table border="1"> <tr><td>En</td><td>\$INS</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>\$INSP</td><td>D</td></tr> <tr><td>S1</td><td></td><td></td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> </table>	En	\$INS	D	S1			S2			S3			En	\$INSP	D	S1			S2			S3			Вставка строковой переменной						
En	\$INS	D																																	
S1																																			
S2																																			
S3																																			
En	\$INSP	D																																	
S1																																			
S2																																			
S3																																			
2122	SPLIT	—	✓	<table border="1"> <tr><td>En</td><td>SPLIT</td><td>D1</td></tr> <tr><td>S1</td><td></td><td>D2</td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> <tr><td>S4</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>SPLITP</td><td>D1</td></tr> <tr><td>S1</td><td></td><td>D2</td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> <tr><td>S4</td><td></td><td></td></tr> </table>	En	SPLIT	D1	S1		D2	S2			S3			S4			En	SPLITP	D1	S1		D2	S2			S3			S4			Разделение строковой переменной
En	SPLIT	D1																																	
S1		D2																																	
S2																																			
S3																																			
S4																																			
En	SPLITP	D1																																	
S1		D2																																	
S2																																			
S3																																			
S4																																			
2123	MERGE	—	✓	<table border="1"> <tr><td>En</td><td>MERGE</td><td>D1</td></tr> <tr><td>S1</td><td></td><td>D2</td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> </table> <table border="1"> <tr><td>En</td><td>MERGECP</td><td>D1</td></tr> <tr><td>S1</td><td></td><td>D2</td></tr> <tr><td>S2</td><td></td><td></td></tr> <tr><td>S3</td><td></td><td></td></tr> </table>	En	MERGE	D1	S1		D2	S2			S3			En	MERGECP	D1	S1		D2	S2			S3			Объединение строковой переменной						
En	MERGE	D1																																	
S1		D2																																	
S2																																			
S3																																			
En	MERGECP	D1																																	
S1		D2																																	
S2																																			
S3																																			

● Инструкции для Ethernet

API	Код инструкции		Импульсная инструкция	Символ		Функционал																																																								
	16 бит	32 бит																																																												
2200	SOPEN	—	✓	<table border="1"> <tr><td colspan="2">SOPEN</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	SOPEN		En		S1		S2		S3		<table border="1"> <tr><td colspan="2">SOPENP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	SOPENP		En		S1		S2		S3		Открытие сокета																																				
SOPEN																																																														
En																																																														
S1																																																														
S2																																																														
S3																																																														
SOPENP																																																														
En																																																														
S1																																																														
S2																																																														
S3																																																														
2201	SSEND	—	✓	<table border="1"> <tr><td colspan="2">SSEND</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	SSEND		En		S1		S2		<table border="1"> <tr><td colspan="2">SSENDP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	SSENDP		En		S1		S2		Отправка данных через сокет																																								
SSEND																																																														
En																																																														
S1																																																														
S2																																																														
SSENDP																																																														
En																																																														
S1																																																														
S2																																																														
2203	SCLOSE	—	✓	<table border="1"> <tr><td colspan="2">SCLOSE</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	SCLOSE		En		S1		S2		<table border="1"> <tr><td colspan="2">SCLOSEP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> </table>	SCLOSEP		En		S1		S2		Закрытие сокета																																								
SCLOSE																																																														
En																																																														
S1																																																														
S2																																																														
SCLOSEP																																																														
En																																																														
S1																																																														
S2																																																														
2204	MSEND	—	✓	<table border="1"> <tr><td colspan="2">MSEND</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	MSEND		En		S1	D	S2		S3		<table border="1"> <tr><td colspan="2">MSENDP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D</td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	MSENDP		En		S1	D	S2		S3		Отправка email																																				
MSEND																																																														
En																																																														
S1	D																																																													
S2																																																														
S3																																																														
MSENDP																																																														
En																																																														
S1	D																																																													
S2																																																														
S3																																																														
2206	INTOA	—	✓	<table border="1"> <tr><td colspan="2">INTOA</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	INTOA		En		S	D	<table border="1"> <tr><td colspan="2">INTOAP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	INTOAP		En		S	D	Преобразование IP-адреса целочисленного типа в IP-адрес строкового типа																																												
INTOA																																																														
En																																																														
S	D																																																													
INTOAP																																																														
En																																																														
S	D																																																													
2207	IATON	—	✓	<table border="1"> <tr><td colspan="2">IATON</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	IATON		En		S	D	<table border="1"> <tr><td colspan="2">IATONP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>D</td></tr> </table>	IATONP		En		S	D	Преобразование IP-адреса строкового типа в IP-адрес целочисленного типа																																												
IATON																																																														
En																																																														
S	D																																																													
IATONP																																																														
En																																																														
S	D																																																													
2208	EIPRW	—	—	<table border="1"> <tr><td colspan="2">EIPRW</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td>D1</td></tr> <tr><td>S2</td><td>D2</td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> <tr><td>S5</td><td></td></tr> <tr><td>S6</td><td></td></tr> <tr><td>S7</td><td></td></tr> <tr><td>n</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	EIPRW		En		S1	D1	S2	D2	S3		S4		S5		S6		S7		n		S		Чтение и запись данных Ethernet/IP																																			
EIPRW																																																														
En																																																														
S1	D1																																																													
S2	D2																																																													
S3																																																														
S4																																																														
S5																																																														
S6																																																														
S7																																																														
n																																																														
S																																																														
2209	SCONF	—	✓	<table border="1"> <tr><td colspan="2">SCONF</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> <tr><td>S5</td><td></td></tr> <tr><td>S6</td><td></td></tr> <tr><td>S7</td><td></td></tr> <tr><td>S8</td><td></td></tr> <tr><td>S9</td><td></td></tr> <tr><td>S10</td><td></td></tr> </table>	SCONF		En		S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		<table border="1"> <tr><td colspan="2">SCONFP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> <tr><td>S5</td><td></td></tr> <tr><td>S6</td><td></td></tr> <tr><td>S7</td><td></td></tr> <tr><td>S8</td><td></td></tr> <tr><td>S9</td><td></td></tr> <tr><td>S10</td><td></td></tr> </table>	SCONFP		En		S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		Задание параметров TCP/UDP сокета								
SCONF																																																														
En																																																														
S1																																																														
S2																																																														
S3																																																														
S4																																																														
S5																																																														
S6																																																														
S7																																																														
S8																																																														
S9																																																														
S10																																																														
SCONFP																																																														
En																																																														
S1																																																														
S2																																																														
S3																																																														
S4																																																														
S5																																																														
S6																																																														
S7																																																														
S8																																																														
S9																																																														
S10																																																														
2210	MCONF	—	✓	<table border="1"> <tr><td colspan="2">MCONF</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> <tr><td>S5</td><td></td></tr> <tr><td>S6</td><td></td></tr> <tr><td>S7</td><td></td></tr> <tr><td>S8</td><td></td></tr> <tr><td>S9</td><td></td></tr> <tr><td>S10</td><td></td></tr> <tr><td>S11</td><td></td></tr> <tr><td>S12</td><td></td></tr> </table>	MCONF		En		S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		S11		S12		<table border="1"> <tr><td colspan="2">MCONF</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> <tr><td>S5</td><td></td></tr> <tr><td>S6</td><td></td></tr> <tr><td>S7</td><td></td></tr> <tr><td>S8</td><td></td></tr> <tr><td>S9</td><td></td></tr> <tr><td>S10</td><td></td></tr> <tr><td>S11</td><td></td></tr> <tr><td>S12</td><td></td></tr> </table>	MCONF		En		S1		S2		S3		S4		S5		S6		S7		S8		S9		S10		S11		S12		Чтение/запись данных Modbus TCP
MCONF																																																														
En																																																														
S1																																																														
S2																																																														
S3																																																														
S4																																																														
S5																																																														
S6																																																														
S7																																																														
S8																																																														
S9																																																														
S10																																																														
S11																																																														
S12																																																														
MCONF																																																														
En																																																														
S1																																																														
S2																																																														
S3																																																														
S4																																																														
S5																																																														
S6																																																														
S7																																																														
S8																																																														
S9																																																														
S10																																																														
S11																																																														
S12																																																														
2211	EMCONF1	—	✓	<table border="1"> <tr><td colspan="2">EMCONF1</td></tr> <tr><td>En</td><td></td></tr> <tr><td>Server</td><td></td></tr> <tr><td>Port</td><td></td></tr> <tr><td>LMail</td><td></td></tr> <tr><td>Sub</td><td></td></tr> <tr><td>Verify</td><td></td></tr> <tr><td>User</td><td></td></tr> <tr><td>Passw</td><td></td></tr> </table>	EMCONF1		En		Server		Port		LMail		Sub		Verify		User		Passw		<table border="1"> <tr><td colspan="2">EMCONF1P</td></tr> <tr><td>En</td><td></td></tr> <tr><td>Server</td><td></td></tr> <tr><td>Port</td><td></td></tr> <tr><td>LMail</td><td></td></tr> <tr><td>Sub</td><td></td></tr> <tr><td>Verify</td><td></td></tr> <tr><td>User</td><td></td></tr> <tr><td>Passw</td><td></td></tr> </table>	EMCONF1P		En		Server		Port		LMail		Sub		Verify		User		Passw		Задание параметров сервера Email																				
EMCONF1																																																														
En																																																														
Server																																																														
Port																																																														
LMail																																																														
Sub																																																														
Verify																																																														
User																																																														
Passw																																																														
EMCONF1P																																																														
En																																																														
Server																																																														
Port																																																														
LMail																																																														
Sub																																																														
Verify																																																														
User																																																														
Passw																																																														



API	Код инструкции		Импульсная инструкция	Символ		Функционал																
	16 бит	32 бит																				
2212	EMCONF 2	—	✓	<table border="1"> <tr><td colspan="2">EMCONF2</td></tr> <tr><td>En</td><td></td></tr> <tr><td>Index</td><td></td></tr> <tr><td>Mail</td><td></td></tr> </table>	EMCONF2		En		Index		Mail		<table border="1"> <tr><td colspan="2">EMCONF2P</td></tr> <tr><td>En</td><td></td></tr> <tr><td>Index</td><td></td></tr> <tr><td>Mail</td><td></td></tr> </table>	EMCONF2P		En		Index		Mail		Задание адреса Email
EMCONF2																						
En																						
Index																						
Mail																						
EMCONF2P																						
En																						
Index																						
Mail																						

- Инструкции для карты памяти

API	Код инструкции		Импульсная инструкция	Символ		Функционал																																
	16 бит	32 бит																																				
2300	MWRIT	—	✓	<table border="1"> <tr><td colspan="2">MWRIT</td></tr> <tr><td>En</td><td></td></tr> <tr><td>C</td><td></td></tr> <tr><td>S</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> </table>	MWRIT		En		C		S		S1		S2		S3		S4		<table border="1"> <tr><td colspan="2">MWRITP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>C</td><td></td></tr> <tr><td>S</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> <tr><td>S4</td><td></td></tr> </table>	MWRITP		En		C		S		S1		S2		S3		S4		Запись данных с ПЛК на карту памяти
MWRIT																																						
En																																						
C																																						
S																																						
S1																																						
S2																																						
S3																																						
S4																																						
MWRITP																																						
En																																						
C																																						
S																																						
S1																																						
S2																																						
S3																																						
S4																																						
2301	MREAD	—	✓	<table border="1"> <tr><td colspan="2">MREAD</td></tr> <tr><td>En</td><td></td></tr> <tr><td>C</td><td>D</td></tr> <tr><td>S</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	MREAD		En		C	D	S		S1		S2		S3		<table border="1"> <tr><td colspan="2">MREADP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>C</td><td>D</td></tr> <tr><td>S</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	MREADP		En		C	D	S		S1		S2		S3		Считывание данных с карты памяти в ПЛК				
MREAD																																						
En																																						
C	D																																					
S																																						
S1																																						
S2																																						
S3																																						
MREADP																																						
En																																						
C	D																																					
S																																						
S1																																						
S2																																						
S3																																						
2302	MTWRIT	—	✓	<table border="1"> <tr><td colspan="2">MTWRIT</td></tr> <tr><td>En</td><td></td></tr> <tr><td>C</td><td></td></tr> <tr><td>S</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	MTWRIT		En		C		S		S1		S2		S3		<table border="1"> <tr><td colspan="2">MTWRITP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>C</td><td></td></tr> <tr><td>S</td><td></td></tr> <tr><td>S1</td><td></td></tr> <tr><td>S2</td><td></td></tr> <tr><td>S3</td><td></td></tr> </table>	MTWRITP		En		C		S		S1		S2		S3		Запись строковых данных на карту памяти				
MTWRIT																																						
En																																						
C																																						
S																																						
S1																																						
S2																																						
S3																																						
MTWRITP																																						
En																																						
C																																						
S																																						
S1																																						
S2																																						
S3																																						
2303	MEMW	—	✓	<table border="1"> <tr><td colspan="2">MEMW</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>n</td></tr> <tr><td>D</td><td></td></tr> </table>	MEMW		En		S	n	D		<table border="1"> <tr><td colspan="2">MEMWP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td>n</td></tr> <tr><td>D</td><td></td></tr> </table>	MEMWP		En		S	n	D		Запись данных в файловый регистр																
MEMW																																						
En																																						
S	n																																					
D																																						
MEMWP																																						
En																																						
S	n																																					
D																																						

- Инструкции по управлению задачами

API	Код инструкции		Импульсная инструкция	Символ		Функционал												
	16 бит	32 бит																
2400	TKON	—	✓	<table border="1"> <tr><td colspan="2">TKON</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	TKON		En		S		<table border="1"> <tr><td colspan="2">TKONP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	TKONP		En		S		Запуск циклической задачи
TKON																		
En																		
S																		
TKONP																		
En																		
S																		
2401	TKOFF	—	✓	<table border="1"> <tr><td colspan="2">TKOFF</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	TKOFF		En		S		<table border="1"> <tr><td colspan="2">TKOFFP</td></tr> <tr><td>En</td><td></td></tr> <tr><td>S</td><td></td></tr> </table>	TKOFFP		En		S		Отключение циклической задачи
TKOFF																		
En																		
S																		
TKOFFP																		
En																		
S																		

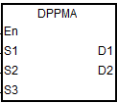
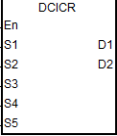
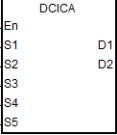
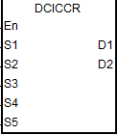
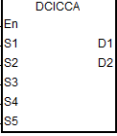
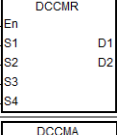
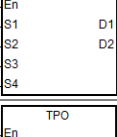
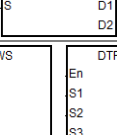
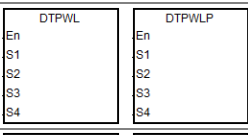
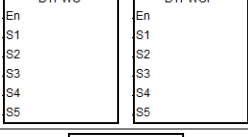
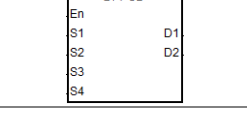

● Инструкции SFC

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
2500	SFCRUN	—	—		Запуск SFC
2501	SFCPSE	—	—		Постановка SFC на паузу
2502	SFCSTP	—	—		Останов SFC

3

● Инструкции высокоскоростных выходов

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
2700	—	DPLSY	—		Высокоскоростной импульсный выход (без рамповой функции)
2701	—	DPLSR	—		Высокоскоростной импульсный выход (с рамповой функцией)
2702	PWM	DPWM	—		ШИМ
2703	JOG	DJOG	—		Выход JOG
2704	—	DZRN	—		Возврат в нулевое значение
2705	—	DPLSV	—		Регулируемый импульсный выход
2706	—	DDRVI	—		Управление относительным позиционированием
2707	—	DDRVA	—		Управление абсолютным позиционированием
2708	CSFO	—	—		Захват скорости и пропорциональный выход
2709	—	DDRVM	—		Метка выравнивания позиционирования
2710	—	DPPMR	—		2-осевая «от точки к точке» синхронизация движения по относительным координатам

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
2711	–	DPPMA	–		2-осевая «от точки к точке» синхронизация движения по абсолютным координатам
2712	–	DCICR	–		2-осевая дуговая интерполяция по часовой стрелке в относительных координатах
2713	–	DCICA	–		2-осевая дуговая интерполяция по часовой стрелке в абсолютных координатах
2714	–	DCICCR	–		2-осевая дуговая интерполяция против часовой стрелки в относительных координатах
2715	–	DCICCA	–		2-осевая дуговая интерполяция против часовой стрелки в абсолютных координатах
2716	–	DCCMR	–		Круговое перемещение в относительных координатах
2717	–	DCCMA	–		Круговое перемещение в абсолютных координатах
2718	TPO	–	–		Управление выходом таблиц позиционирования
2719	–	DTPWS	✓		Задание выходных параметров для одиночной оси в таблице позиционирования
2720	–	DTPWL	✓		Задание параметров линейной интерполяции в таблице позиционирования
2721	–	DPTWC	✓		Задание параметров дуговой интерполяции в таблице позиционирования
2723	–	DPPGB	–		Переход «от точки к точке» вперед и назад

● Инструкции коммуникации CANopen

API	Код инструкции		Импульсная инструкция	Символ	Функционал
	16 бит	32 бит			
2800	INITC	–	–		Инициализация приводов для связи по CANopen
2801	ASDON	–	–		Запуск и отключение приводов
2802	CASD	–	–		Настройка времени разгона и торможения для привода
2803	–	DDRVIC	–		Управление относительным позиционированием для привода
2804	–	DDRVAC	–		Управление абсолютным позиционированием для сервопривода
2805	–	DPLSVC	–		Управление скоростью для привода
2806	ZRNC	–	–		Возврат в начальную позицию (Homing)
2807	COPRW	–	–		Запись и чтение данных коммуникации CANopen
2808	COPWL	DCOPWL	–		Запись нескольких значений параметров по CANopen
2809	RSTD	–	–		Отправка команды сброса или NMT
2810	ZRNM	–	–		Настройка режима возврата в начальную позицию для сервопривода Delta
2811	EMER	–	–		Чтение аварийного сообщения

3

## 3.4.2 Прикладные инструкции (сортировка по алфавиту)

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
Символ	0114	\$+	–	✓	Связывание строковых переменных
	2118	\$CLR	–	✓	Очистка строковой переменной
	2117	\$DEL	–	✓	Удаление символов в строковой переменной
	2109	\$FSTR	–	✓	Преобразование числа с плавающей точкой в строковую переменную
	2110	\$FVAL	–	✓	Преобразование строковой переменной в число с плавающей точкой
	2119	\$INS	–	✓	Вставка строковой переменной
	2112	\$LEFT	–	✓	Извлечение символов в строковой переменной начинается слева
	2106	\$LEN	–	✓	Подсчет длины строковой переменной
	2113	\$MIDR	–	✓	Получение части строковой переменной
	0302	\$MOV	–	✓	Передача строковой переменной
	2111	\$RIGHT	–	✓	Извлечение символов в строковой переменной начинается справа
	2116	\$RPLC	–	✓	Замена символов в строковой переменной
	2115	\$SER	–	✓	Поиск строковой переменной
	0102	*	D*	✓	Умножение двоичных чисел
	0103	/	D/	✓	Деление двоичных чисел
	0100	+	D+	✓	Сложение двоичных чисел
	А	1210	ABS	DABS	✓
0705		ABSD	DABSD	–	Абсолютный барабанный секвенсор
0700		ALT	–	✓	Альтернативное переключение между ВКЛ и ВЫКЛ
0043		AND\$<>	–	–	$S1 \neq S2$
0042		AND\$=	–	–	$S1 = S2$
0812		AND&	DAND&	–	$S1 \& S2$
0814		AND^	DAND^	–	$S1 \wedge S2$
0813		AND	DAND	–	$S1   S2$
0010		AND<	DAND<	–	$S1 < S2$
0011		AND<=	DAND<=	–	$S1 \leq S2$
0007		AND<>	DAND<>	–	$S1 \neq S2$
0006		AND=	DAND=	–	$S1 = S2$
0008		AND>	DAND>	–	$S1 > S2$
0009		AND>=	DAND>=	–	$S1 \geq S2$
0076		ANDZ<	DANDZ<	–	$ S1 - S2  <  S3 $
0077		ANDZ<=	DANDZ<=	–	$ S1 - S2  \leq  S3 $

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	0073	ANDZ<>	DANDZ<>	–	S1-S2 ≠ S3
	0072	ANDZ=	DANDZ=	–	S1-S2 = S3
	0074	ANDZ>	DANDZ>	–	S1-S2  >  S3
	0075	ANDZ>=	DANDZ>=	–	S1-S2 ≥ S3
	1703	ARWS	–	–	Клавиши со стрелками
	2801	ASDON	–	–	Запуск и отключение сервоприводов
	1815	ASDRW	–	–	Инструкция по последовательной коммуникации для сервопривода Delta
В	1222	BAND	DBAND	✓	Контроль нечувствительности
	0200	BCD	DBCD	✓	Преобразование двоичного числа в двоично-десятичное число
	2102	BCDDA	DBCDDA	✓	Преобразование двоично-десятичного числа в код ASCII
	0201	BIN	DBIN	✓	Преобразование двоично-десятичного числа в двоичное число
	2100	BINDA	DBINDA	✓	Преобразование десятичного числа со знаком в код ASCII
	2101	BINHA	DBINHA	✓	Преобразование шестнадцатеричного числа в код ASCII
	0113	BK-	DBK-	✓	Вычитание двоичных чисел в блоках
	0112	BK+	DBK+	✓	Сложение двоичных чисел в блоках
	0214	BKBCD	–	✓	Преобразование двоичных чисел в блоке в десятичные числа в блоке
	0215	BKBIN	–	✓	Преобразование двоичных чисел в блоке в двоично-десятичные числа в блоке
	0304	BMOV	DBMOV	✓	Перенос всех данных
	1207	BON	DBON	✓	Проверка состояния бита
	1302	BREAK	–	–	Завершение цикла FOR-NEXT
	1113	BSFL	–	✓	Сдвиг состояний n битовых регистров на один бит влево
	1112	BSFR	–	✓	Сдвиг состояний n битовых регистров на один бит вправо
0307	BXCH	–	✓	Обмен всеми данными	
С	1819	CANRS	–	–	Пользовательская настройка коммуникации по CAN, прием и отправка данных
	2802	CASD	–	–	Настройка времени разгона и торможения для сервопривода
	1209	CCD	–	✓	Проверка суммы
	1816	CCONF	–	✓	Установка параметров в таблице обмена данными коммуникационного порта
	0065	CHKADR	–	–	Проверка адреса контактного типа регистра указателя
	0400	CJ	–	✓	Условный переход

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	0303	CML	DCML	✓	Инвертирование данных
	0054	CMP	DCMP	✓	Сравнение значений
	0063	CMPT<	–	✓	Сравнение таблиц ON: <
	0064	CMPT<=	–	✓	Сравнение таблиц ON: ≤
	0060	CMPT<>	–	✓	Сравнение таблиц ON: ≠
	0059	CMPT=	–	✓	Сравнение таблиц ON: =
	0061	CMPT>	–	✓	Сравнение таблиц ON: >
	0062	CMPT>=	–	✓	Сравнение таблиц ON: ≥
	1003	CNT	–	–	16-битный счетчик
	1813	COMDF	–	✓	Настройка коммуникации для последовательного коммуникационного порта
	1812	COMRS	–	–	Отправка и прием данных коммуникации
	2807	COPRW	–	–	Запись и чтение данных коммуникации CANopen
	2808	COPWL	DCOPWL	–	Запись нескольких значений параметров CANopen
	1807	CRC	–	–	Циклическая проверка избыточности
	2708	CSFO	–	–	Захват скорости и пропорциональный выход
D	0101	–	D-	✓	Вычитание двоичных чисел $S_1 - S_2 = D$
	2717	–	DCCMA	–	Круговое перемещение в абсолютных координатах
	2716	–	DCCMR	–	Круговое перемещение в относительных координатах
	2713	–	DCICA	–	2-осевая дуговая интерполяция по часовой стрелке в абсолютных координатах
	2715	–	DCICCA	–	2-осевая дуговая интерполяция против часовой стрелки в абсолютных координатах
	2714	–	DCICCR	–	2-осевая дуговая интерполяция против часовой стрелки в относительных координатах
	2712	–	DCICR	–	2-осевая дуговая интерполяция по часовой стрелке в относительных координатах
	1004	–	DCNT	–	32-битный счетчик
	2707	–	DDRVA	–	Управление абсолютным позиционированием
	2804	–	DDRVC	–	Управление абсолютным позиционированием для сервопривода
	2706	–	DDRVI	–	Управление относительным позиционированием

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	2803	–	DDRVIC	–	Управление относительным позиционированием для сервопривода
	2709	–	DDRVM	–	Метка выравнивания позиционирования
	1006	–	DHSCR	–	Сброс высокоскоростного сравнения
	1005	–	DHSCS	–	Настройка высокоскоростного сравнения
	0601	–	DHSRF	✓	Обновление значений высокоскоростного сравнения
	1007	–	DHSZ	–	Сравнение высокоскоростных входных зон
	0708	–	DPIDE	–	ПИД-алгоритм
	2701	–	DPLSR	–	Высокоскоростной импульсный выход (с рамповой функцией)
	2705	–	DPLSV	–	Регулируемый импульсный выход
	2805	–	DPLSVC	–	Управление скоростью для сервопривода
	2700	–	DPLSY	–	Высокоскоростной импульсный выход (без рамповой функции)
	2723	–	DPPGB	–	Переход «от точки к точке» вперед и назад
	2711	–	DPPMA	–	2-осевая «от точки к точке» синхронизация движения по абсолютным координатам
	2710	–	DPPMR	–	2-осевая «от точки к точке» синхронизация движения по относительным координатам
	2721	–	DPTWC	✓	Задание параметров дуговой интерполяции в таблице позиционирования
	1410	–	DPUCNT	–	Функция высокоскоростного счетчика модуля PU
	1406	–	DPUDRA	–	Абсолютный адресный выход модуля PU (с разгоном и замедлением)
	1405	–	DPUDRI	–	Относительная выходная позиция модуля PU (с разгоном и замедлением)
	1409	–	DPUMPG	–	Выход MPG модуля PU
	1404	–	DPUPLS	–	Импульсный выход модуля PU (без разгона)
	1226	–	DTM	✓	Передача и перемещение данных
	2720	–	DTPWL	✓	Задание параметров линейной интерполяции в таблице позиционирования
	2719	–	DTPWS	✓	Задание выходных параметров для одиночной оси в таблице позиционирования
	2704	–	DZRN	–	Возврат в нулевое значение
	2105	DABCD	DDABCD	✓	Преобразование кода ASCII в двоично-десятичное число
	2103	DABIN	DDABIN	✓	Преобразование двоично-десятичного кода ASCII со знаком в двоично-десятичное число со знаком
	0116	DEC	DDEC	✓	Вычитание единицы из двоичного числа
	1202	DECO	–	✓	Декодировщик



Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	1901	DELAY	–	✓	Задержка выполнения программы
	0500	DI	–	–	Отключение прерывания
D	1215	DIS	–	✓	Разделение 16-битных данных
	0118	DIV16	DIV32	✓	Деление двоичных чисел для 16 бит Деление двоичных чисел для 32 бит
	0504	DIX	–	–	Включение специального прерывания
	1820	DMVSH	–	–	Разрешение обнаружения и коммуникации Delta DMV
	1818	DNETRW	–	–	Чтение и запись данных коммуникации DeviceNet
	1607	DST	–	✓	Летнее время
	1702	DSW	–	–	DIP-переключатель
E	0501	EI	–	–	Включение прерывания
	2208	EIPRW	–	–	Чтение и запись данных Ethernet/IP
	0503	EIX	–	–	Отключение специального прерывания
	2211	EMCONF1	–	✓	Задание параметров сервера Email
	2212	EMCONF2	–	✓	Задание адреса Email
	2811	EMER	–	–	Чтение аварийного сообщения
	1203	ENCO	–	✓	Энкодер
	1905	EPOP	–	✓	Чтение данных в индексные регистры
1904	EPUSH	–	✓	Сохранение содержимого индексных регистров	
F	0105	–	F-	✓	Вычитание чисел с плавающей точкой $S_1 - S_2 = D$
	0106	–	F*	✓	Умножение чисел с плавающей точкой $S_1 * S_2 = D$
	0107	–	F/	✓	Деление чисел с плавающей точкой $S_1 / S_2 = D$
	0104	–	F+	✓	Сложение чисел с плавающей точкой $S_1 + S_2 = D$
	1504	–	FACOS	✓	Арккосинус числа с плавающей запятой
	0028	–	FAND<	–	$S_1 < S_2$
	0029	–	FAND<=	–	$S_1 \leq S_2$
	0025	–	FAND<>	–	$S_1 \neq S_2$
	0024	–	FAND=	–	$S_1 = S_2$
	0026	–	FAND>	–	$S_1 > S_2$
	0027	–	FAND>=	–	$S_1 \geq S_2$
	1503	–	FASIN	✓	Арсинус числа с плавающей запятой
	1505	–	FATAN	✓	Арктангенс числа с плавающей запятой
0212	–	FBCD	✓	Преобразование двоичного числа с плавающей точкой в десятичное число с плавающей точкой	

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	0213	–	FBIN	✓	Преобразование десятичного числа с плавающей точкой в двоичное число с плавающей точкой
	0056	–	FCMP	✓	Сравнение чисел с плавающей точкой
	1501	–	FCOS	✓	Косинус числа с плавающей запятой
	1507	–	FCOSH	✓	Гиперболический косинус числа с плавающей запятой
	1510	–	FDEG	✓	Преобразование радиан в градусы
	1513	–	FEXP	✓	Экспонента числа с плавающей запятой
	0022	–	FLD<	–	$S1 < S2$
	0023	–	FLD<=	–	$S1 \leq S2$
	0019	–	FLD<>	–	$S1 \neq S2$
F	0018	–	FLD=	–	$S1 = S2$
	0020	–	FLD>	–	$S1 > S2$
	0021	–	FLD>=	–	$S1 \geq S2$
	1515	–	FLN	✓	Натуральный логарифм числа с плавающей запятой
	1514	–	FLOG	✓	Логарифм числа с плавающей запятой
	1224	–	FMEAN	✓	Среднее значение чисел с плавающей точкой
	0211	–	FNEG	✓	Изменение знака 32-битового числа с плавающей точкой
	0034	–	FOR<	–	$S1 < S2$
	0035	–	FOR<=	–	$S1 \leq S2$
	0031	–	FOR<>	–	$S1 \neq S2$
	0030	–	FOR=	–	$S1 = S2$
	0032	–	FOR>	–	$S1 > S2$
	0033	–	FOR>=	–	$S1 \geq S2$
	1516	–	FPOW	✓	Возведение в степень числа с плавающей запятой
	1509	–	FRAD	✓	Преобразование градусов в радианы
	1500	–	FSIN	✓	Синус числа с плавающей запятой
	1506	–	FSINH	✓	Гиперболический синус числа с плавающей запятой
	1512	–	FSQR	✓	Квадратный корень числа с плавающей запятой
	1225	–	FSUM	✓	The sum of the floating points
	1502	–	FTAN	✓	Тангенс числа с плавающей запятой
1508	–	FTANH	✓	Гиперболический тангенс числа с плавающей запятой	
0057	–	FZCP	✓	Зонное сравнение с плавающей точкой	

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	0202	FLT	DFLT	✓	Преобразование двоичного целого числа в двоичное число с плавающей точкой
	1300	FOR	–	–	Начало вложенного цикла
	1400	FROM	DFROM	✓	Чтение данных из регистра управления в специальном модуле
G	0209	GBIN	DGBIN	✓	Преобразование кода Грея в двоичное число
	0402	GOEND	–	–	Переход к END
	1902	GPWM	–	–	Общая ШИМ
	0208	GRY	DGRY	✓	Преобразование двоичного числа в код Грея
H	2104	HABIN	DHABIN	✓	Преобразование кода ASCII в шестнадцатеричное число
	1701	HKY	DHKY	–	Шестнадцать кнопочная клавиатура
	1604	HOUR	–	–	Счетчик времени работы
I	2207	IATON	–	✓	Преобразование IP-адреса строчного типа в IP-адрес целочисленного типа
	1012	IETS	–	✓	Старт инструкции измерения времени
	1013	IETE	–	✓	Завершение инструкции измерения времени
	0115	INC	DINC	✓	Добавление единицы к двоичному числу
	1906	INFO	–	✓	Считывание системных данных
	0706	INCD	–	–	Инкрементный барабанный секвенсор
I	2800	INITC	–	–	Инициализация сервоприводов для связи по CANopen
	0204	INT	DINT	✓	Преобразование 32-битового числа с плавающей точкой в двоичное целое число
	2206	INTOA	–	✓	Преобразование 16-битового в 32-битовое значение
J	0401	JMP	–	–	Безусловный переход
	2703	JOG	DJOG	–	Выход JOG
L	1415	LCCAL	–	–	Калибровка канала модуля LC (тензомодуля)
	1416	LCWEI	–	–	Чтение значения массы через модуль LC (тензомодуль)
	0037	LD\$<>	–	–	S1≠S2
	0036	LD\$=	–	–	S1 = S2
	0809	LD&	DLD&	–	S1&S2
	0811	LD^	DLD^	–	S1^S2
	0810	LD	DLD	–	S1 S2
	0004	LD<	DLD<	–	S1 < S2
	0005	LD<=	DLD<=	–	S1 ≤ S2
	0001	LD<>	DLD<>	–	S1≠S2

3

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	0000	LD=	DLD=	–	S1 = S2
	0002	LD>	DLD>	–	S1 > S2
	0003	LD>=	DLD>=	–	S1 ≥ S2
	0070	LDZ<	DLDZ<	–	S1-S2  <  S3
	0071	LDZ<=	DLDZ<=	–	S1-S2  ≤  S3
	0067	LDZ<>	DLDZ<>	–	S1-S2  ≠  S3
	0066	LDZ=	DLDZ=	–	S1-S2 = S3
	0068	LDZ>	DLDZ>	–	S1-S2  >  S3
	0069	LDZ>=	DLDZ>=	–	S1-S2  ≥  S3
	1221	LIMIT	DLIMIT	✓	Ограничение значения в пределах допустимого диапазона
	1806	LRC	–	–	Продольная проверка четности
M	0801	MAND	–	✓	Матричное И
	2123	MERGE	–	✓	Объединение строковой переменной
	1214	MBC	–	✓	Подсчет битов со значением 0 или 1
	0904	MBR	–	✓	Вращение битов матрицы
	1212	MBRD	–	✓	Чтение битов матрицы
	1109	MBS	–	✓	Сдвиг битов матрицы
	1213	MBWR	–	✓	Запись битов матрицы
	0058	MCMP	–	✓	Матричное сравнение
	2210	MCONF	–	✓	Чтение/запись данных Modbus TCP
	1208	MEAN	DMEAN	✓	Среднее значение
	2303	MEMW	–	✓	Запись данных в файловый регистр
	1211	MINV	–	✓	Инвертирование битов матрицы
	0206	MMOV	–	✓	Преобразование 16-битового в 32-битовое значение
	1808	MODRW	–	–	Чтение/запись данных MODBUS
	1817	MODRWE	–	–	Чтение и запись данных Modbus без использования какого-либо флага
	0803	MOR	–	✓	Матричное ИЛИ
	0300	MOV	DMOV	✓	Передача данных S: Источник данных D: Место назначения данных
	0310	MOVB	–	✓	Передача нескольких бит
	2301	MREAD	–	✓	Считывание данных с карты памяти в ПЛК
	2204	MSEND	–	✓	Отправка email
0704	MTR	–	–	Матричный вход	

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	2302	MTWRIT	–	✓	Запись строки на карту памяти
	0117	MUL16	MUL32	✓	Умножение двоичных чисел для 16 бит Умножение двоичных чисел для 32 бит
	2300	MWRIT	–	✓	Запись данных с ПЛК на карту памяти
	0805	MXOR	–	✓	Матричное ИСКЛЮЧАЮЩЕЕ ИЛИ
N	0210	NEG	DNEG	✓	Второе дополнение
	1301	NEXT	–	–	Конец вложенного цикла
	0305	NMOV	DNMOV	✓	Перенос данных на несколько регистров
	1115	NSFL	–	✓	Сдвиг n регистров влево
	1114	NSFR	–	✓	Сдвиг n регистров вправо
O	0049	OR\$<>	–	–	$S1 \neq S2$
	0048	OR\$=	–	–	$S1 = S2$
	0815	OR&	DOR&	–	$S1 \& S2$
	0817	OR^	DOR^	–	$S1 \wedge S2$
	0816	OR	DOR	–	$S1   S2$
	0016	OR<	DOR<	–	$S1 < S2$
	0017	OR<=	DOR<=	–	$S1 \leq S2$
	0013	OR<>	DOR<>	–	$S1 \neq S2$
	0012	OR=	DOR=	–	$S1 = S2$
	0014	OR>	DOR>	–	$S1 > S2$
	0015	OR>=	DOR>=	–	$S1 \geq S2$
	0082	ORZ<	DORZ<	–	$ S1 - S2  <  S3 $
	0083	ORZ<=	DORZ<=	–	$ S1 - S2  \leq  S3 $
	0079	ORZ<>	DORZ<>	–	$ S1 - S2  \neq  S3 $
	0078	ORZ=	DORZ=	–	$ S1 - S2  =  S3 $
	0080	ORZ>	DORZ>	–	$ S1 - S2  >  S3 $
0081	ORZ>=	DORZ>=	–	$ S1 - S2  \geq  S3 $	
P	1402	PUCONF	–	✓	Настройка параметров управления выходами модуля PU
	1408	PUJOG	–	–	Выход JOG модуля PU
	1403	PUSTAT	–	–	Чтение состояния выходов модуля PU
	1407	PUZRN	–	–	Возврат в нулевую точку для модуля PU
	1009	PWD	–	–	Измерение ширины импульса
	2702	PWM	DPWM	–	ШИМ
R	0703	RAMP	DRAMP	–	Сигнал циклического скачка
	1517	RAND	–	✓	Случайное число

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	0903	RCL	DRCL	✓	Вращение влево с флагом переноса
	0901	RCR	DRCR	✓	Вращение вправо с флагом переноса
	0600	REF	–	✓	Обновление входов/выходов
	0207	RMOV	–	✓	Преобразование 32-битового в 16-битовое значение
	0902	ROL	DROL	✓	Вращение влево
	0900	ROR	DROR	✓	Вращение вправо
	1000	RST	DRST	–	Сброс контакта на ВЫКЛ или очистка значения в регистре
	2809	RSTD	–	–	Отправка команды Сброс или NMT
S	0216	SCAL	DSCAL	✓	Операция масштабирования значений
	0222	SCLM	DSCLM	✓	Операция масштабирования многоточечной области
	2203	SCLOSE	–	✓	Закрытие сокета
	0217	SCLP	DSCLP	✓	Тип параметра операции масштабирования значений
	2209	SCONF	–	✓	Задание параметров TCP/UDP сокета
	1204	SEGD	–	✓	Семисегментное декодирование
	1704	SEGL	–	–	Семи сегментный дисплей с фиксацией
	1200	SER	DSER	✓	Поиск данных
	2501	SFCPSE	–	–	Постановка SFC на паузу
	2500	SFCRUN	–	–	Запуск SFC
	2502	SFCSTP	–	–	Останов SFC
	1107	SFDEL	–	✓	Удаление данных из списка данных
	1108	SFINS	–	✓	Добавление данных в список данных
	1111	SFL	–	✓	Сдвиг значений битов в 16-битовых регистрах на <b>n</b> бит влево
	1106	SFPO	–	✓	Чтение последних данных из списка данных
	1110	SFR	–	✓	Сдвиг значений битов в 16-битовых регистрах на <b>n</b> бит вправо
	1105	SFRD	–	✓	Сдвиг данных и чтение их из словного регистра
	1101	SFTL	–	✓	Сдвиг состояния битовых регистров влево
	1100	SFTR	–	✓	Сдвиг состояния битовых регистров вправо
	1104	SFWR	–	✓	Сдвиг данных и запись их в словный регистр
	0309	SMOV	–	✓	Передача цифр
	2122	SPLIT	–	✓	Разделение строковой переменной
	2200	SOPEN	–	✓	Открытие сокета
	1205	SORT	DSORT	✓	Сортировка данных
1511	SQR	DSQR	✓	Квадратный корень двоичного числа	

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
	2201	SSEND	–	✓	Отправка данных через сокет
	0702	STMR	–	–	Специальный таймер
	1201	SUM	DSUM	✓	Количество включенных битов
	0711	SUNRS	–	✓	Задание времени восхода и захода солнца
	0308	SWAP	DSWAP	✓	Замена старшего байта младшим байтом
Т	1603	T-	–	✓	Вычитание времени
	1602	T+	–	✓	Добавление времени
	1605	TCMP	–	✓	Сравнение времени
	2401	TKOFF	–	✓	Отключение циклической задачи
	2400	TKON	–	✓	Запуск циклической задачи
	1700	TKY	DTKY	–	Десятиклавиатурная клавиатура
	1001	TMR	–	–	16-битовый таймер (шаг: 100 мс)
	1002	TMRH	–	–	16-битовый таймер (шаг: 1 мс)
	1401	TO	DTO	✓	Запись данных в регистр управления в специальном модуле
	2718	TPO	–	–	Управление выходом таблицей позиционирования
	1600	TRD	–	✓	Считывание времени
	0701	TTMR	–	–	Обучающийся таймер
	1601	TWR	–	✓	Запись времени
	1606	TZCP	–	✓	Сравнение временных зон
U	1216	UNI	–	✓	Объединение 16-битных данных
V	1814	VFDRW	–	–	Инструкция по последовательной коммуникации для ПЧ Delta
W	0800	WAND	DAND	✓	Логическое И
	1608	WWON	–	–	Еженедельная настройка рабочего времени
	1900	WDT	–	✓	Сторожевой таймер
	0808	WINV	DINV	✓	Логическое инвертирование INV
	0802	WOR	DOR	✓	Логическое ИЛИ
	1103	WSFL	–	✓	Сдвиг данных в словных регистрах влево
	1102	WSFR	–	✓	Сдвиг данных в словных регистрах вправо
	1217	WSUM	DWSUM	✓	Получение суммы
X	0804	WXOR	DXOR	✓	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
	0306	XCH	DXCH	✓	Обмен данными
Y	0709	XCMP	–	–	Настройка сравнения входов нескольких рабочих станций
	0710	YOUT	–	–	Сравнение результатов работы нескольких рабочих станций

Буква алфавита	API	Код инструкции		Импульсн. инструкция	Функция
		16-бит	32-бит		
Z	0055	ZCP	DZCP	✓	Зонное сравнение
	1223	ZONE	DZONE	✓	Управление зоной
	2806	ZRNC	–	–	Возврат в начальную позицию (Homing)
	2810	ZRNM	–	–	Настройка режима возврата в начальную позицию для сервопривода Delta
	1206	ZRST	–	✓	Очистка зоны



---

## Глава 4. Структура инструкций

### Содержание

4.1	Состав прикладных инструкций API .....	4-2
4.2	Описание использования операндов .....	4-5
4.3	Ограничения на применение инструкций .....	4-6
4.4	Индексные регистры .....	4-7
4.5	Регистры метки .....	4-9
4.6	Регистры метки таймеров .....	4-11
4.7	Регистры метки 16-битного счетчика .....	4-12
4.8	Регистры метки 32-битного счетчика .....	4-14
4.9	Файловые регистры .....	4-15

## 4.1 Состав прикладных инструкций API

Каждая инструкция имеет свой собственный код и номер API. Номер API инструкции в таблице ниже – 0300, а код (название) инструкции - MOV, функцией данной инструкции является передача данных.

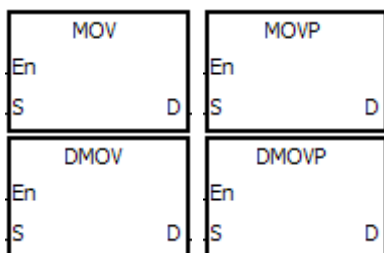
API	Код инструкции			Операнды								Функция				
0300	D	MOV	P	S · D								Передача данных				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○	○	○	○		○
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●		●		●	●	
D		●	●		●	●	●		●		●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символ:



S : Источник данных  
D : Получатель данных

1. Объекты, используемые инструкцией, перечислены в столбце **Операнды**. S, D, n и m используются в качестве операндов в соответствии с их функциями. Когда используется более одного операнда, и эти операнды имеют одну и ту же функцию, они обозначаются дополнительно цифрами. Например, S1, S2.
2. Импульсные инструкции требуют прибавления после кода инструкции буквы 'P'. 32-битные инструкции требуют прибавления в начале кода инструкции буквы 'D', в случае применения 32-битной импульсной инструкции название будет выглядеть как "D\*\*\*P", где \*\*\* код инструкции.
3. Символ F в столбце **Операнды** представляет собой одинарное значение с плавающей запятой (32-битное).
4. Символ ● указывает, что объект может быть изменен с помощью индексного регистра, а символ ○ указывает на то, что объект не может быть изменен с помощью индексного регистра. Например, регистр данных, обозначенный операндом S, может быть изменен индексным регистром.
5. Применимая модель ПЛК указана в таблице. Там же указано, может ли инструкция использоваться как импульсная, 16-битная, 32-битная или 64-битная.
6. Если необходимо использовать инструкцию в функциональном блоке, а выходы, входы и объекты данных назначены операндами, необходимо использовать регистры указателя. Для таймера, 16-битного счетчика и 32-битного счетчика, назначенных операндами, необходимо использовать регистр указателя таймера,

регистр указателя 16-битного счетчика и регистр указателя 32-битного счетчик. См. Разделы 4.4 ~ 4.7 для получения дополнительной информации или Раздел 7.2.4 в ISPSOft.

7. Описание символов, представляющих инструкцию MOV в ISPSOft:

**MOV, MOVP, DMOV, DMOVP:** Коды инструкций

**En:** Разрешено

**S:** Источник данных (возможный формат операнда: слово / двойное слово)

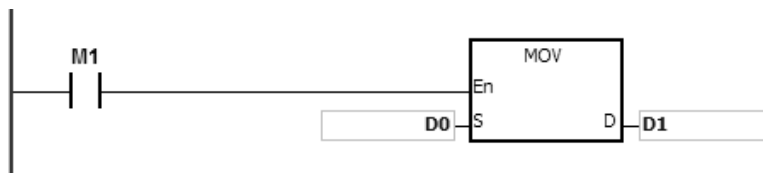
**D:** Получатель данных (возможный формат операнда: слово / двойное слово)

**Составление прикладных инструкций:**

Некоторые прикладные инструкции состоят только из кодов инструкций. Например, инструкции EI, DI, WDT и т. д. Однако, большинство прикладных инструкций состоят из кодов инструкций и нескольких операндов.

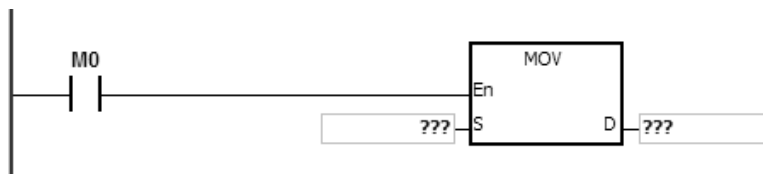
Каждая прикладная инструкция имеет свой собственный номер API и код. Например, код инструкции API0300 - это MOV (передача данных).

Ввод инструкции напрямую: пользователи могут вводить инструкцию с помощью ПО ISPSOft. Для инструкции MOV необходимо только ввести имя (код) инструкции и обозначения операндов «MOV D0 D1».



Ввод инструкции путем перетаскивания: можно перетащить команду MOV из списка API в ISPSOft в область, где можно редактировать LD.

Ввод инструкции на панели инструментов: можно выбрать API / FB Selection на панели инструментов в ISPSOft, а затем выбрать API. Наконец, можно выбрать инструкцию MOV в Data Transfer. Операнды после этого назначаются отдельно.



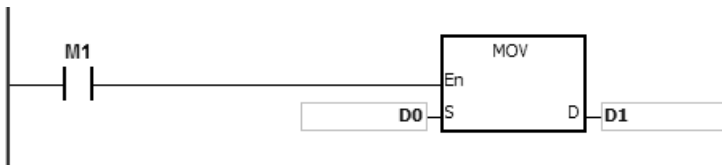
<b>S</b>	Операнд источника данных Если существует более одного операнда источника данных, эти операнды представлены как <b>S1, S2</b> и т.д.
<b>D</b>	Операнд получателя данных Если существует более одного операнда получателя данных, эти операнды представлены как <b>D1, D2</b> и т.д.
Если операндом является константа К/Н или регистр, они представлены как <b>m, m1, m2, n, n1</b> или <b>n2</b> .	

**Длина операндов (16-битные, 32-битные инструкции и инструкции с числами с плавающей запятой):**

**16-битные или 32-битные инструкции**

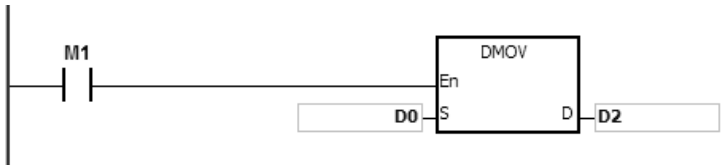
Операнды подразделяются на 2 группы: 16-битные или 32-битные для работы с данными разной длины. Соответственно, инструкции для работы с ними могут быть также 16-битными и 32-битными. Префикс "D" означает 32-битную инструкцию.

16-битная инструкция MOV



Если M1 включен, данные из D0 передаются в D1.

32-битная инструкция DMOV



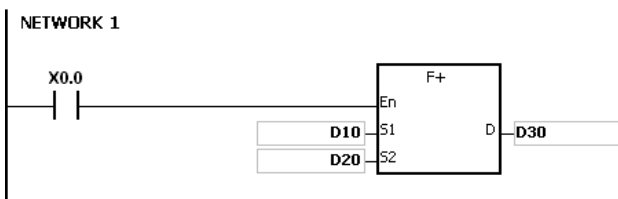
Если M1 включен, данные из (D1, D0) передаются в (D3, D2).

Инструкции для чисел с плавающей запятой

Инструкции для чисел с плавающей запятой могут поддерживать 32-разрядные инструкции с числами с плавающей запятой с одинарной точностью. Обратитесь к Главе 2 за дополнительной информацией о числах с плавающей запятой.

4

32-битная инструкция для чисел с плавающей запятой с одинарной точностью F+

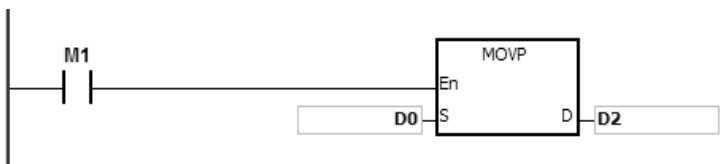


Если X0.0 включен, данные из (D11, D10) и (D21, D20) передаются в (D31, D30).

Непрерывное и импульсное выполнение инструкции:

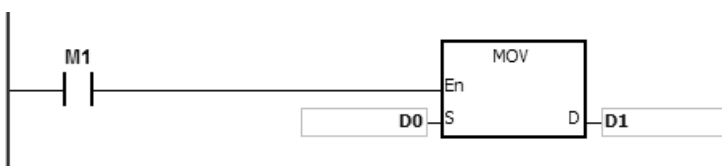
1. Существует два типа выполнения инструкций - непрерывное выполнение и импульсное выполнение. Время цикла программы меньше, когда инструкции не выполняются. Таким образом, с помощью импульсного выполнения можно уменьшить время цикла программы.
2. Импульсное выполнение позволяет включить соответствующую инструкцию по переднему (поднимающемуся) фронту и выполнить ее в одном цикле программы. Кроме того, пока управляющий вход остается включенным, соответствующая инструкция не будет выполняться повторно. Для повторного выполнения управляющий вход должен быть переключен с ВЫКЛ на ВКЛ еще раз.
3. Импульсная инструкция:

Импульсное выполнение



Когда X0 переходит из состояния OFF в ON, команда MOV будет выполнена только один раз и не будет больше выполняться в цикле.

Непрерывное выполнение



Если X1= ON, команда MOV будет повторно выполняться в каждом цикле программы. Это непрерывное выполнение команд.

Когда условный контракт M1 выключен, команда не выполняется, а значение в операнде получателя D не изменяется.

## 4.2 Описание использования операндов

В ПЛК серии AS есть 2 типа операндов; один тип предназначен для задания пользователем, а другой - для задания системой.

### Операнды, назначаемые пользователем:

1. Входные реле: X0.0 ~ X63.15 or X0 ~ X63
2. Выходные реле: Y0.0 ~ Y63.15 or Y0 ~ Y63
3. Внутренние реле: M0 ~ M8191
4. Шаговые реле: S0 ~ S2047
5. Таймеры: T0 ~ T511
6. 16-битные счетчики: C0 ~ C511
7. 32-битные счетчики: HC0 ~ HC255
8. Регистры данных: D0 ~ D29999 or D0.0 ~ D29999.15
9. Файловые регистры: FR0 ~ FR65535
10. Специальные вспомогательные флаги: SM0 ~ SM2047
11. Специальные регистры данных: SR0 ~ SR2047
12. Индексные регистры: E0 ~ E9
13. Константы: Десятичные константы обозначаются как K, а шестнадцатеричные как 16#.
14. Строковые переменные: "\$"
15. Числа с плавающей запятой: Числа с плавающей запятой с одинарной точностью обозначаются как F
16. Длина данных в одном регистре обычно составляет 16 бит. Если пользователю необходимо хранить 32-битные данные в регистре, нужно назначить два последовательных регистра.
17. Если операнд, используемый в 32-битной инструкции, обозначается как D0, то используется 32-битный регистр данных, состоящий из (D1, D0). D1 представляет старшие 16 бит, а D0 - младшие 16 бит. Это же правило применяется к таймеру и 16-разрядному счетчику.
18. Когда 32-битный счетчик HC используется в качестве регистра данных, его можно назначить только операндом, используемым в 32-битной инструкции.
19. Индексные регистры могут использоваться только в 16-битной инструкции.

Дополнительные сведения см. Глава 2 Объекты.

### Операнды, определяемые системой:

1. Система назначает переменные для объявления типа BOOL, WORD, INT и т.л.: U0 ~ U16387 и W0 ~ W29999.
2. Запуск выполнения и останов выполнения задачи: TK0 ~ TK31
3. Переменные символы типа указателя, поддерживаемые объекты и их применение показаны ниже.

Тип указателя	Применение	
Основной (POINTER)	Диапазон	PR0 ~ PR15 · PR0.0~PR15.15
	Макс. количество	В каждом функциональном блоке может использоваться до 16 указателей
	Виды	Переменные типов WORD/DWORD/LWORD/INT/DINT/LINT или регистры данных, входные и выходные реле (X0, Y0 и т.д.)
Для таймеров (T_POINTER)	Диапазон	TR0 ~ TR7
	Макс. количество	В каждом функциональном блоке может использоваться до 8 указателей
	Виды	Переменные и объекты типа таймера
Для счетчиков (C_POINTER)	Диапазон	CR0 ~ CR7
	Макс. количество	В каждом функциональном блоке может использоваться до 8 указателей
	Виды	Переменные и объекты типа счетчика
Для высокоскоростных счетчиков (HC_POINTER)	Диапазон	HCR0 ~ HCR7
	Макс. количество	В каждом функциональном блоке может использоваться до 8 указателей
	Виды	Переменные и объекты типа 32-битного счетчика

### 4.3 Ограничения на применение инструкций

- Инструкции, которые могут использоваться только в функциональных блоках  
API0065 CHKADR, FB\_NP, FB\_PN, NED, ANED, ONED, PED, APED, OPED
- Инструкции, которые не могут использоваться в задачах прерывания  
GOEND
- Инструкции, которые не поддерживаются в функциональных блоках  
LDP, ANDP, ORP, LDF, ANDF, ORF, PLS, PLF, NP, PN, MC/MCR, GOEND и все импульсные инструкции.

При необходимости использования некоторых из приведенных выше инструкций, можно применять заменяющие их инструкции.

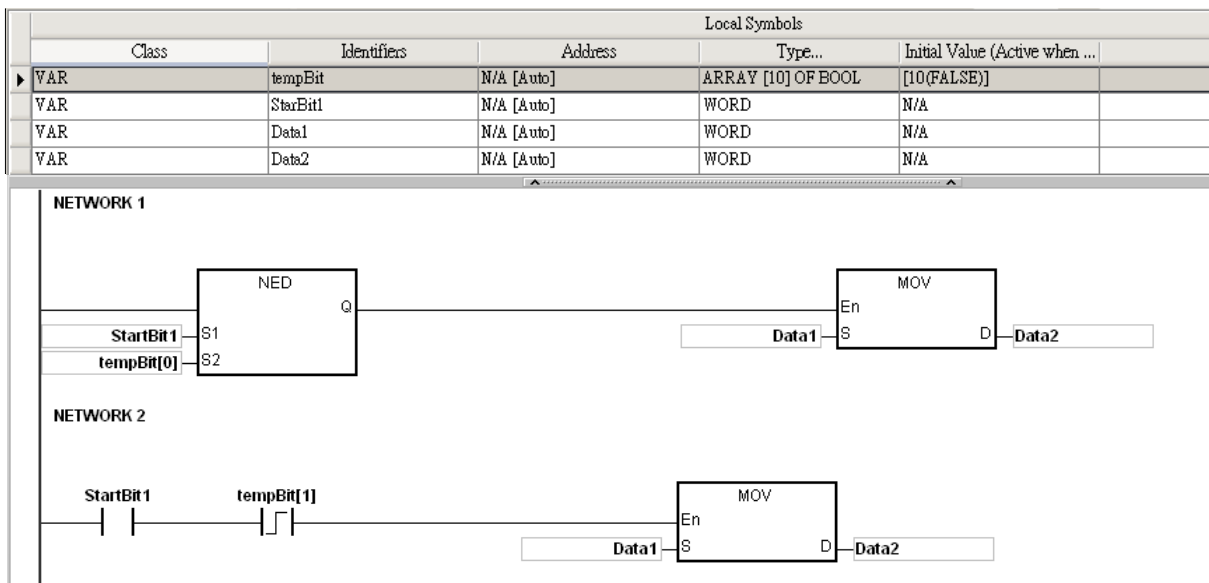
Инструкции, которые не могут использоваться в функциональных блоках	Заменяющие инструкции, которые могут использоваться в функциональных блоках
LDP/ANDP/ORP	NED/ANED/ONED
LDF/ANDF/ORF	PED/APED/OPED
PLS	-
PLF	-

Инструкции, которые не могут использоваться в функциональных блоках	Заменяющие инструкции, которые могут использоваться в функциональных блоках
NP	FB_NP
PN	FB_PN
MC	-
MCR	-
Импульсные инструкции	*1

\*1: Импульсные инструкции не могут использоваться в функциональных блоках. Если необходимо получить функцию импульсной инструкции в функциональном блоке, см. следующий пример.

#### Пример:

1. Сначала объявите 10 системных переменных tempBit [10], используемых в системе.
2. Когда StartBit1 переключается с ВЫКЛ на ВКЛ, метод 1 (сеть 1) и метод 2 (сеть 2) могут выполнять только один раз команду MOV; выберите один для использования.
3. Переменная tempBit, используемая в системе, не может использоваться повторно.



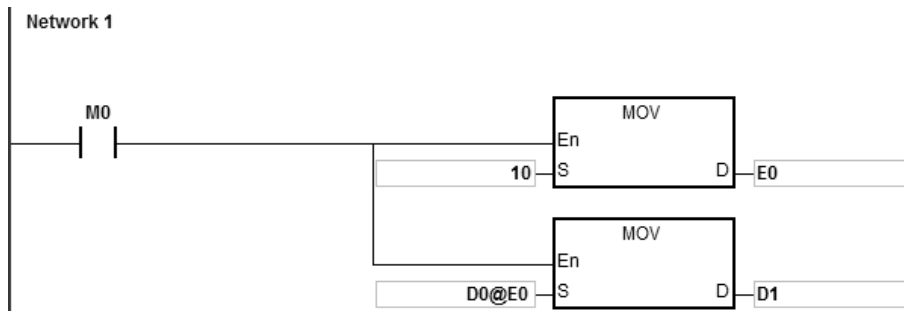
## 4.4 Индексные регистры

Индексный регистр – это 16-битный регистр данных. Он похож на общий регистр, в котором данные могут быть прочитаны и записаны в него, однако в основном используется как индексный регистр. Диапазон индексных регистров E0 ~ E9.

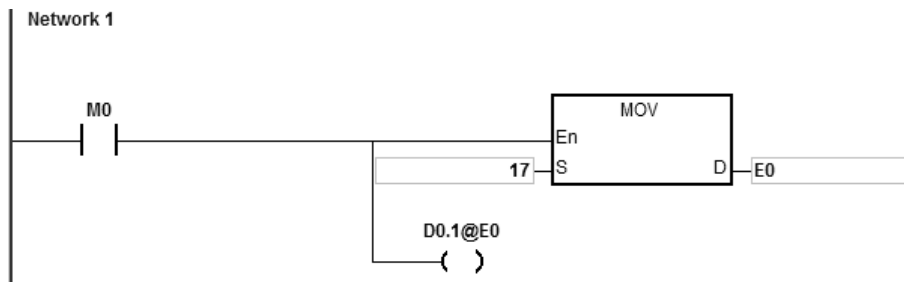
Индексные регистры используются следующим образом.

1. Использование имени регистра для изменения объекта

Когда M0 включен, E0=10, D0@E0=D (0+10)=D10 и D1=D10.



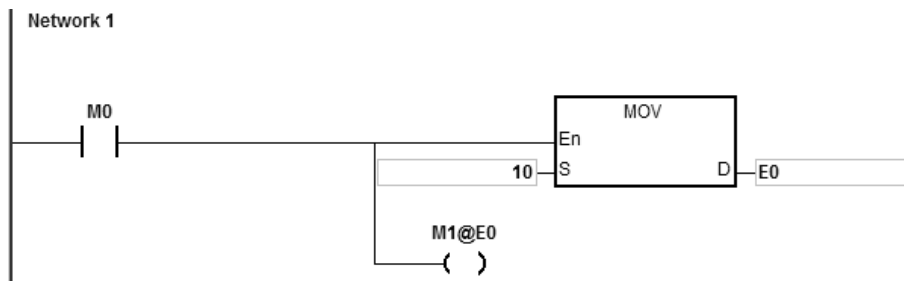
Когда M0 включен, E0=10, E1=17, D1@E0=D (1+10)=D11, D11 включен.



ПРИМЕЧАНИЕ 1. ПЛК серии AS поддерживают использование имени регистра для модификации объекта, например D0.1@E0, но не поддерживает двойную модификацию, например D0 @ E1.1 @ E0.

ПРИМЕЧАНИЕ 2: Когда E0 = 17, D0.1@E0=D0. (1 + 17) = D1.2, а D1.2 будет включен. А часть 1бита @ E0 = (1 + 17) = 18. Однако максимальное число битов равно 15. Поскольку  $m = 18/16 = 1$ , а остаток равен 2, последним результатом модификации является  $D(0 + 1).2 = D1.2$ . D1.2 будет включен.

Когда M0 включен, E0=10 и M1@E0=M (1+10)=M11. M11 включен.



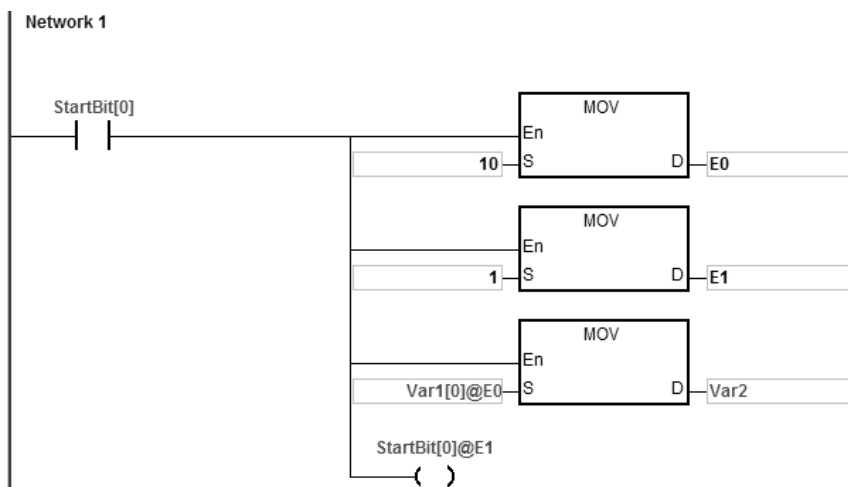
2. Сначала объявляются переменные, затем происходит модификация объекта

- Объявите три переменные StartBit, Var1 и Var2 в ПО ISPSOft.  
 Тип StartBit булевый, его размер 2 бита. Диапазон от StartBit[0] до StartBit[1].  
 Тип Var1 словный, размер 11 слов. Диапазон от Var1[0] до Var1[10].  
 Тип Var2 словный, размер 1 слово.

Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	N/A	
VAR	Var1	N/A [Auto]	ARRAY [11] OF WORD	N/A	
VAR	Var2	N/A [Auto]	WORD	N/A	



- Когда StartBit[0] включен, E0=10, E1=1, Var1[0]@E0=Var1[10], Var2=Var1[10] и StartBit[0]@E1=StartBit[1] включен.



Дополнительное примечание: Когда пользователь задает переменные в ISPSOft, а переменные добавляются к содержимому регистров с фактическими данными для формирования адресов, необходимо проверять адреса, чтобы предотвратить неправильное выполнение программы.

## 4.5 Регистры метки

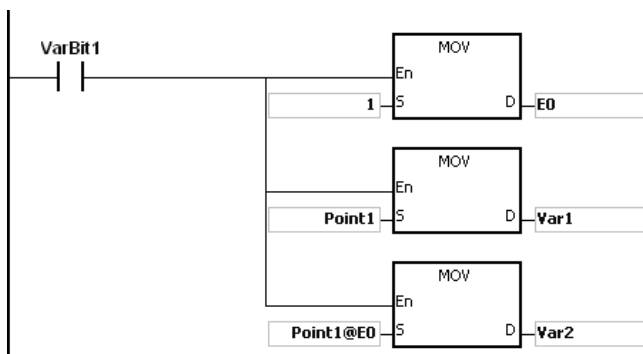
- ISPSOft поддерживает функциональные блоки. Когда тип заданной переменной - VAR\_IN\_OUT, а тип данных - POINTER, переменная является регистром метки. Значение в регистре метки может ссылаться непосредственно на значение, хранящееся в объектах X, Y или D, и регистр метки может автоматически указывать на адрес, связанный с переменной, заданной автоматически в ISPSOft.
- Пользователи могут объявлять 16 регистров метки в каждом функциональном блоке. Диапазон PR0~PR15 или PR0.0~PR15.15.

Пример:

1. Сначала создайте организационную единицу программы (POU) в ISPSOft.
2. Задайте функциональный блок, который называется FB0.



3. Программа в функциональном блоке FB0



4. Задание переменных в функциональном блоке FB0.

Выберите VAR\_IN\_OUT как заявленный тип, Point1 в качестве идентификатора, POINTER в качестве типа данных. Переменной является регистр метки.

Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	VarBit1	N/A [Auto]	BOOL	FALSE	
VAR	Var1	N/A [Auto]	WORD	0	
VAR	Var2	N/A [Auto]	WORD	0	
▶ VAR_IN_OUT	Point1	N/A [Auto]	POINTER	N/A	

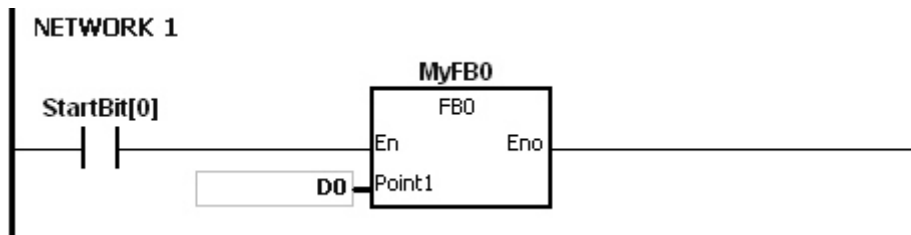
5. Задайте переменные в организационной единице программы (POU).

Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	N/A	
VAR	CVar1	N/A [Auto]	ARRAY [2] OF WORD	N/A	
▶ VAR	MyFB0	N/A [Auto]	FB0	N/A	

6. Вызовите функциональный блок FB0 в POU.

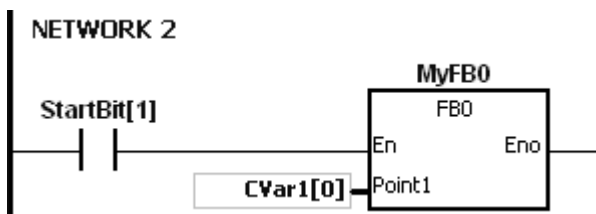
7. Программа в POU

Схема 1: Когда StartBit[0] включен, адрес D0 передается в Point 1 в FB0.



Когда VarBit1 в FB0 включен, E0=1, Var1=D0, Point1@E0=D (0+1)=D1 и Var2=D1..

Схема 2: Когда StartBit[1] включен, адрес CVar1[0] передается в Point1 в FB0.



Var2=CVar1[1]. Когда VarBit1 в FB0 включен, E0=1, Var1=CVar1[0], Point1@E0=CVar1 (0+1)=Cvar1[1] и Var2=CVar1[1].

## 4.6 Регистры метки таймеров

- ISPSoft поддерживает функциональные блоки. Если необходимо использовать таймер в функциональном блоке, в функциональном блоке задается регистр метки таймера. Адрес таймера передается в регистр метки таймера при вызове функционального блока.
- Если заявленный тип переменной VAR\_IN\_OUT и тип данных T\_POINTER, переменная является регистром метки таймера. Значение в регистре метки таймера может ссылаться непосредственно на значение, хранящееся в объекте T или в переменной, которая является таймером в ISPSoft.
- Пользователь может назначить 8 регистров метки таймеров в каждом функциональном блоке. Диапазон TR0~TR7.
- Если нужно использовать инструкцию в функциональном блоке, а таймер задан среди операндов, необходимо использовать регистр метки таймера.

### Пример:

1. Сначала создайте POU в ISPSoft.
2. Создайте функциональный блок, обозначенный как FB0.

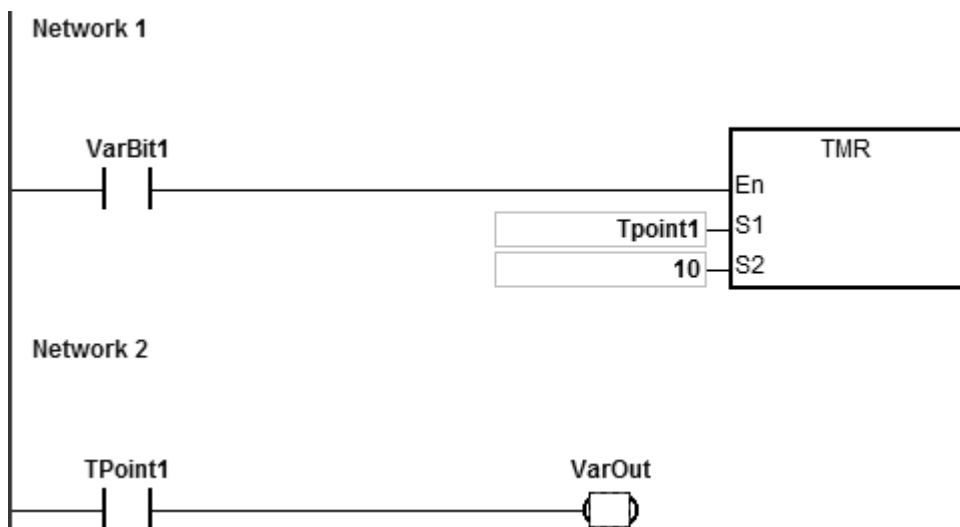


3. Создайте переменные в функциональном блоке FB0.

Задайте VAR\_IN\_OUT как заявленный тип, TPoint1 как идентификатор, T\_POINTER как тип данных. Переменной будет регистр метки таймера.

Local Symbols					
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	VarBit1	N/A [Auto]	BOOL	FALSE	
VAR_IN_OUT	TPoint1	N/A [Auto]	T_POINTER	N/A	
▶ VAR	VarOut	N/A [Auto]	BOOL	FALSE	

4. Программа в функциональном блоке FB0



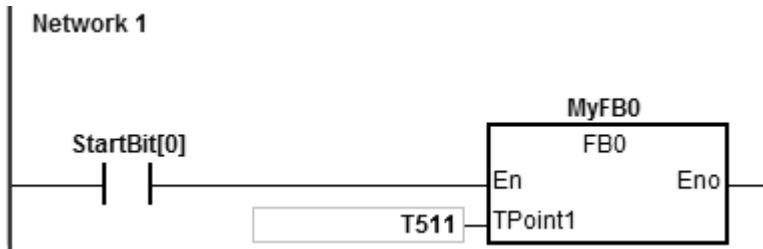
5. Задайте переменную в POU.

Тип данных CVar1 назначается как TIMER.

Local Symbols						
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...	
VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	[2(FALSE)]		
VAR	CVar1	T0	TIMER	N/A		
▶ VAR	MyFB0	N/A [Auto]	FB0	N/A		

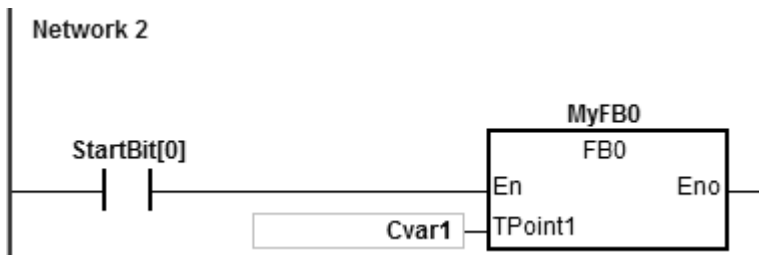
6. Вызовите функциональный блок FB0 в POU.
7. Программа в POU

Схема 1: Когда StartBit[0] включен, адрес T511 передается в TPoint1 в FB0.



Когда VarBit1 в FB0 включен, инструкция TMR выполняется, а TPoint1 (T511) начинает отсчет. Когда значение TPoint1 достигает заданного, включается VarOut.

Схема 2: Когда StartBit[1] включен, адрес CVar1[0] передается в TPoint1 в FB0.



Когда VarBit1 в FB0 включен, инструкция TMR выполняется, а TPoint (CVar1) начинает отсчет. Когда значение TPoint1 достигает заданного, включается VarOut.

## 4.7 Регистры метки 16-битного счетчика

- ISPSOft поддерживает функциональные блоки. Если необходимо использовать таймер в функциональном блоке, в функциональном блоке задается регистр метки 16-битного счетчика. Адрес таймера передается в регистр метки 16-битного счетчика при вызове функционального блока.
- Если заявленный тип переменной VAR\_IN\_OUT и тип данных C\_POINTER, переменная является регистром метки 16-битного счетчика. Значение в регистре метки 16-битного счетчика может ссылаться непосредственно на значение, хранящееся в объекте T или в переменной, которая является счетчиком в ISPSOft.
- Пользователь может назначить 8 регистров метки 16-битных счетчиков в каждом функциональном блоке. Диапазон CR0~CR7.
- Если нужно использовать инструкцию в функциональном блоке, а счетчик задан среди операндов, необходимо использовать регистр метки 16-битного счетчика.

**Пример:**

1. Сначала создайте POU в ISPSOft.
2. Создайте функциональный блок, обозначенный как FB0.



3. Создайте переменные в функциональном блоке FB0.

Задайте VAR\_IN\_OUT как заявленный тип, CPoint1 как идентификатор, C\_POINTER как тип данных. Переменной будет регистр метки 16-битного счетчика.

Local Symbols						
	Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
	VAR	VarBit1	N/A [Auto]	BOOL	FALSE	
▶	VAR_IN_OUT	CPoint1	N/A [Auto]	C_POINTER	N/A	

4. Программа в функциональном блоке FB0



5. Создайте переменную в POU.

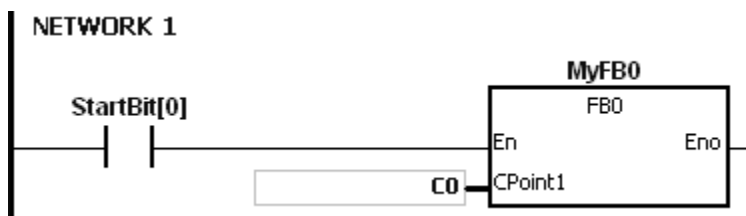
Тип данных CVar1 задается как COUNTER.

Local Symbols						
	Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
	VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	[2(FALSE)]	
	VAR	CVar1	C1	COUNTER	N/A	
▶	VAR	MyFB0	N/A [Auto]	FB0	N/A	

6. Вызовите функциональный блок FB0 в POU.

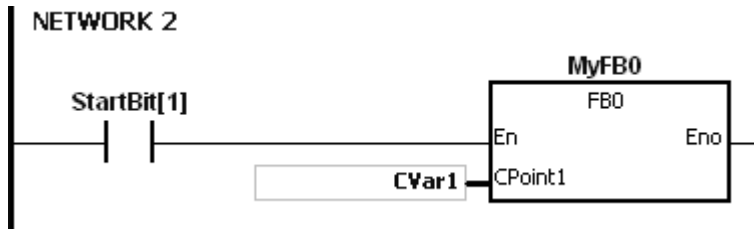
7. Программа в POU

Схема 1: Когда StartBit[0] включен, адрес C0 передается в CPoint1 в FB0.



Когда VarBit1 в FB0 включен, CPoint1 (C0) включается.

Network 2: Когда StartBit[1] включен, адрес CVar1 передается в CPoint1 в FB0.



Когда VarBit1 в FB0 включен, CPoint1 (CVar1) включается.

## 4.8 Регистры метки 32-битного счетчика

- ISPSOft поддерживает функциональные блоки. Если необходимо использовать таймер в функциональном блоке, в функциональном блоке задается регистр метки 32-битного счетчика. Адрес таймера передается в регистр метки 32-битного счетчика при вызове функционального блока.
- Если заявленный тип переменной VAR\_IN\_OUT и тип данных HC\_POINTER, переменная является регистром метки 32-битного счетчика. Значение в регистре метки 32-битного счетчика может ссылаться непосредственно на значение, хранящееся в объекте HC или в переменной, которая является счетчиком в ISPSOft.
- Пользователь может назначить 8 регистров метки 32-битных счетчиков в каждом функциональном блоке. Диапазон HCR0-HCR7.
- Если нужно использовать инструкцию в функциональном блоке, а 32-битный счетчик задан среди операндов, необходимо использовать регистр метки 32-битного счетчика.

### Пример:

1. Сначала создайте POU в ISPSOft.
2. Создайте функциональный блок, обозначенный как FB0.



3. Создайте переменные в функциональном блоке FB0.

Задайте VAR\_IN\_OUT как заявленный тип, HCPoint1 как идентификатор, HC\_POINTER как тип данных. Переменной будет регистр метки 32-битного счетчика.

Local Symbols						
Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...	
VAR	VarBit1	N/A [Auto]	BOOL	FALSE		
▶ VAR_IN_OUT	HCPoint1	N/A [Auto]	HC_POINTER	N/A		

4. Программа в функциональном блоке FB0



5. Создайте переменную в POU.

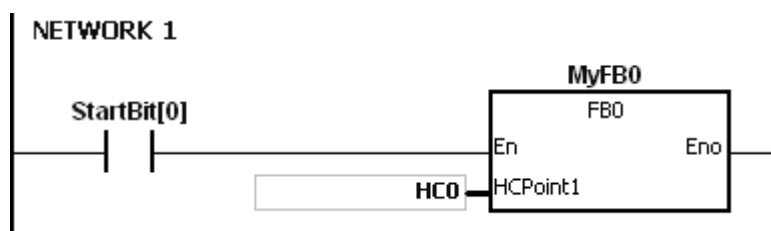
Тип данных CVar1 задается как COUNTER, и пользователь должен заполнить столбец адресов реальным адресом 32-битного счетчика.

Local Symbols						
	Class	Identifiers	Address	Type...	Initial Value	Identifier Comment...
	VAR	StartBit	N/A [Auto]	ARRAY [2] OF BOOL	[2(FALSE)]	
	VAR	CVar1	HC1	COUNTER	N/A	
▶	VAR	MyFB0	N/A [Auto]	FB0	N/A	

8. Вызовите функциональный блок FB0 в POU.

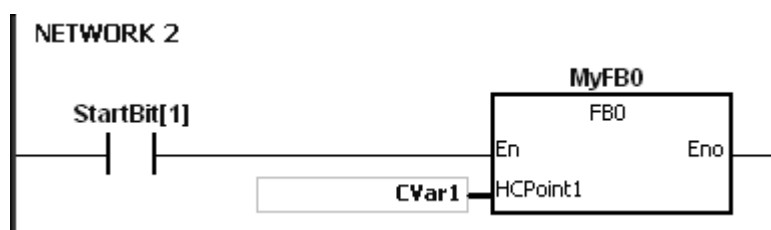
1. Программа в POU

Схема 1: Когда StartBit[0] включен, адрес HC0 передается в HCPoint1 в FB0.



Когда VarBit1 в FB0 включен, HCPoint1 (HC0) включается.

Схема 2: Когда StartBit[1] включен, адрес CVar1 передается в HCPoint1 в FB0.



Когда VarBit1 в FB0 включен, HCPoint1 (CVar1) включается.

## 4.9 Файловые регистры

- ПЛК серии AS имеют файловые регистры (FR) для хранения большого количества параметров.
- Пользователь может редактировать, выгружать, загружать параметры в файловые регистры через ПО ISPSOft.
- Значения в файловом регистре FR можно считывать во время работы ПЛК. См. описание инструкции API2303 MEMW для получения дополнительной информации о том, как записывать данные в файловый регистр FR.





## Глава 5. Базовые инструкции

### Содержание

5.1	Таблица базовых инструкций .....	5-2
5.2	Описание базовых инструкций .....	5-3

## 5.1 Таблица базовых инструкций

Код инструкции	Функция	Операнды	Время выполнения (мкс)
<u>LD/AND/OR</u>	Вывод контакта А / подключение контакта А последовательно / подключение контакта А параллельно	DX, X, Y, M, SM, S, T, C, HC, D	0.025
<u>LDI/ANI/ORI</u>	Вывод контакта В / подключение контакта В последовательно / подключение контакта В параллельно	DX, X, Y, M, SM, S, T, C, HC, D	0.03
<u>OUT</u>	Управление катушкой	DY, Y, M, SM, S, T, C, HC, D	0.04
<u>SET</u>	Включение выхода с фиксацией	DY, Y, M, SM, S, T, C, HC, D	0.04
<u>MC/MCR</u>	Настройка / отмена Мастер-контроля	N	0.24
<u>LDP/ANDP/ORP</u>	Запуск обнаружения по переднему фронту / Последовательное подключение обнаружения по переднему фронту / Параллельное подключение обнаружения по переднему фронту	DX, X, Y, M, SM, S, T, C, HC, D	0.22
<u>LDF/ANDF/ORF</u>	Запуск обнаружения по заднему фронту / Последовательное подключение обнаружения по заднему фронту / Параллельное подключение обнаружения по заднему фронту	DX, X, Y, M, SM, S, T, C, HC, D	0.22
<u>PED/APED/OPED</u>	Запуск обнаружения по переднему фронту / Последовательное подключение обнаружения по переднему фронту / Параллельное подключение обнаружения по переднему фронту	X, Y, M, SM, S, T, C, HC, D	0.22
<u>NED/ANED/ONED</u>	Запуск обнаружения по заднему фронту / Последовательное подключение обнаружения по заднему фронту / Параллельное подключение обнаружения по заднему фронту	X, Y, M, SM, S, T, C, HC, D	0.22
<u>PLS</u>	Импульс по переднему фронту	Y, M, SM, S	0.22
<u>PLF</u>	Импульс по заднему фронту	Y, M, SM, S	0.22
<u>INV</u>	Инвертирование результата логической операции	–	0.22
<u>NP</u>	Цель переключается по переднему фронту	–	0.24
<u>PN</u>	Цель переключается по заднему фронту	–	0.24
<u>FB_NP</u>	Цель переключается по переднему фронту	Y, M, S, D	0.24
<u>FB_PN</u>	Цель переключается по заднему фронту	Y, M, S, D	0.24

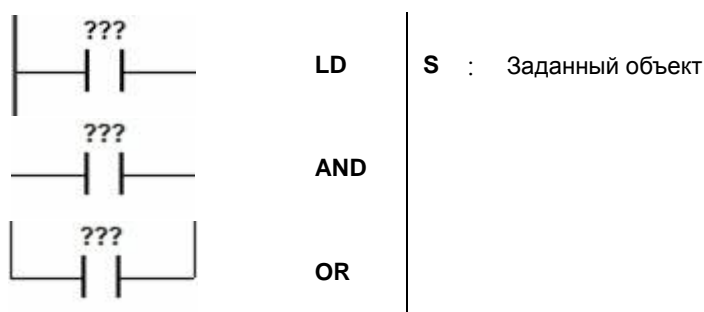
## 5.2 Описание базовых инструкций

Код инструкции	Операнды	Функция
LD/AND/OR	S	Вывод контакта A / подключение контакта A последовательно / подключение контакта A параллельно

Объект	DX	DY	X	Y	M	SM	S	T	C	HC	D
S	●		●	●	●	○	●	●	●	●	●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

Символ:

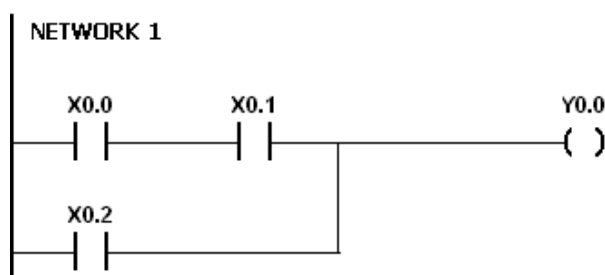


Описание:

1. Инструкция LD применяется к контакту A, который используется для программирования начала логических цепочек или является началом контактной цепи. В контактных схемах инструкция всегда расположена слева и соединяется непосредственно с шиной питания или начинает новую цепь. Состояние контакта сохраняется в специальном аккумулятивном регистре.
2. Инструкция AND используется для последовательного подключения контакта A для программирования операции логического умножения (И) с предыдущим элементом. Результат сохраняется в специальном аккумулятивном регистре.
3. Инструкция OR используется для параллельного подключения контакта A для программирования операции логического сложения (ИЛИ) с предыдущим элементом. Результат сохраняется в специальном аккумулятивном регистре.

Пример:

1. Контакт A X0.0 начальный, контакт A X0.1 подключен последовательно, контакт A X0.2 подключен параллельно, катушка Y0.0 является управляемым объектом.
2. Когда X0.0 и X0.1 включены, или включен X0.2, включается Y0.0.

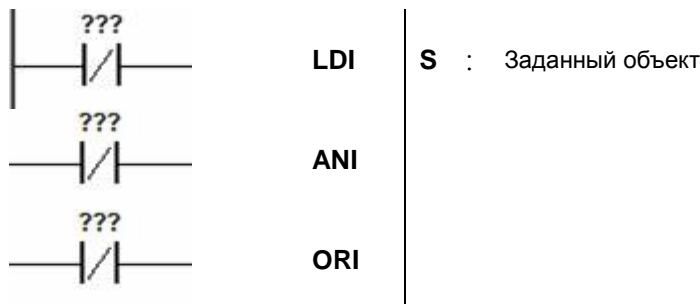


Код инструкции	Операнды	Функция
LDI/ANDI/ORI	S	Вывод контакта В / подключение контакта В последовательно / подключение контакта В параллельно

Объект	DX	DY	X	Y	M	SM	S	T	C	HC	D
S	●		●	●	●	○	●	●	●	●	●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

**Символ:**

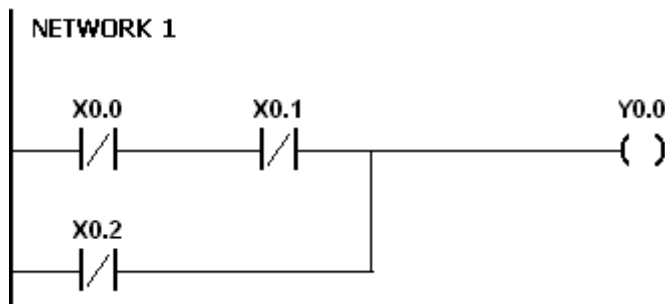


**Описание:**

1. Инструкция LDI применяется к контакту В, который используется для программирования начала логических цепочек или является началом контактной цепи. В контактных схемах инструкция всегда расположена слева и соединяется непосредственно с шиной питания или начинает новую цепь. Состояние контакта сохраняется в специальном аккумулятивном регистре.
2. Инструкция ANDI используется для последовательного подключения контакта В для программирования операции логического умножения (И) с предыдущим элементом. Результат сохраняется в специальном аккумулятивном регистре.
3. Инструкция ORI используется для параллельного подключения контакта В для программирования операции логического сложения (ИЛИ) с предыдущим элементом. Результат сохраняется в специальном аккумулятивном регистре.

**Пример:**

1. Контакт В X0.0 начальный, контакт В X0.1 подключен последовательно, контакт В X0.2 подключен параллельно, катушка Y0.0 является управляемым объектом.
2. Когда X0.0 и X0.1 включены, или включен X0.2, включается Y0.0.

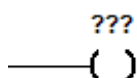


Код инструкции	Операнды	Функция
OUT	D	Управление катушкой (выход)

Объект	DX	DY	X	Y	M	SM	S	T	C	HC	D
D		●		●	●	○	●				●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●												

Символ:



D : Заданный объект

Описание:

- В указанном объекте сохраняется результат последней перед инструкцией OUT логической операции. Инструкция OUT служит для включения или отключения выхода в зависимости от результата логической операции.
- Действия на выходной катушке:

Результат логической операции	Выход		
	Состояние катушки	Контакт	
		Контакт А (Н/О)	Контакт В (Н/З)
Ложь	ВЫКЛ	ВЫКЛ	ВКЛ
Истина	ВКЛ	ВКЛ	ВЫКЛ

Example:

- Контакт В X0.0 начальный, контакт А X0.1 подключен последовательно, катушка Y0.0 является управляемым объектом.
- Когда X0.0 выключен и X0.1 включен, включается Y0.0.



Код инструкции		Операнды						Функция					
SET		D						Включение выхода с фиксацией					
Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D		
D		●		●	●	○	●				●		
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●												

Символ:



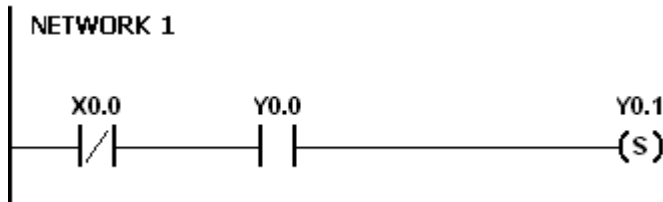
Описание:

Когда выполняется входное условие для инструкции SET, то она включает стоящий за ней операнд и фиксирует его состояние независимо от того, действует входное условие или нет. Отключить операнд (заданный объект) можно с помощью инструкции RST.

5

Пример:

1. Контакт В X0.0 начальный, контакт А Y0.0 подключен последовательно, Y0.1 фиксируется включенным.
2. Если X0.0 выключен и Y0.0 включен, Y0.1 включен. Даже если результат операции меняется, Y0.1 остается включенным.



Код инструкции	Операнды	Функция
MC/MCR	N	Настройка / отмена Мастер-контроля

Символ:



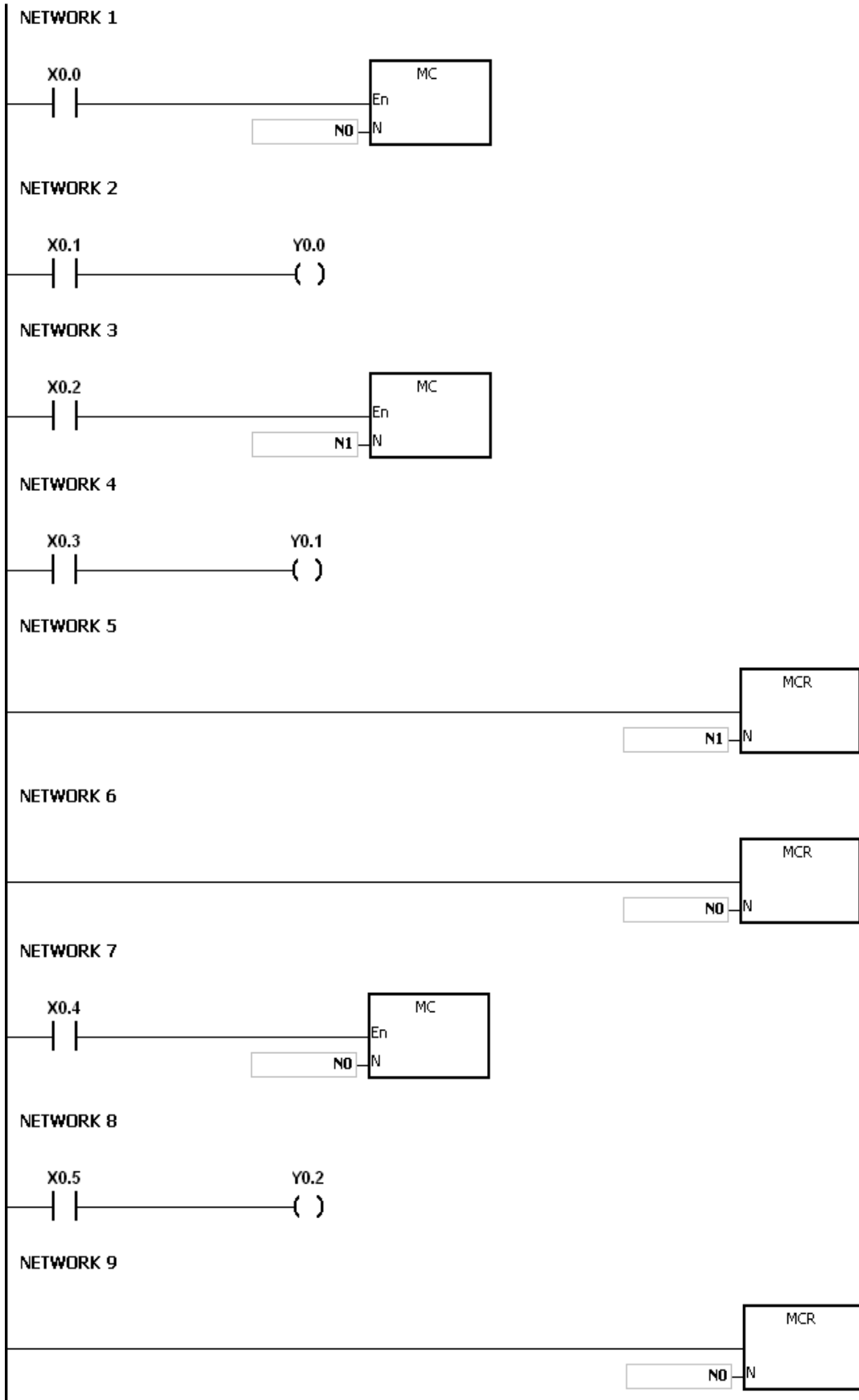
Описание:

- Инструкция MCR используется для установки Мастер-управления. Когда выполняется MC, инструкции между MC и MCR выполняются, как обычно. Когда MC отключена, действия инструкций между MC и MCR следующие.

Тип инструкции	Описание
Таймер общего назначения	Значение таймера сбрасывается до нуля. Катушка и контакт выключены.
Таймер, используемый в функциональном блоке	Значение таймера сбрасывается до нуля. Катушка и контакт выключены.
Аккумулятивный таймер	Катушка выключена. Значение таймера и состояние контакта остаются неизменными.
Счетчик	Катушка выключена. Значение таймера и состояние контакта остаются неизменными.
Все катушки управляют выходами	Все катушки выключены.
Объекты, управляемые SET и RST	Состояние регистров остается неизменным.
Прикладная инструкция	Все прикладные инструкции не выполняются. Цикл FOR / NEXT повторяется N раз, но действия инструкций внутри цикла FOR / NEXT соответствуют инструкциям команд между MC и MR.

- Инструкция MCR используется для сброса Мастер-управления и помещается в конец Мастер-управляющей программы. Перед MCR не должно быть никаких контактных инструкций.
- MC / MCR поддерживает вложенную структуру программы. Существует не более 32 уровней вложенных программных структур (N0 ~ N31). См. пример ниже.

Пример:



5

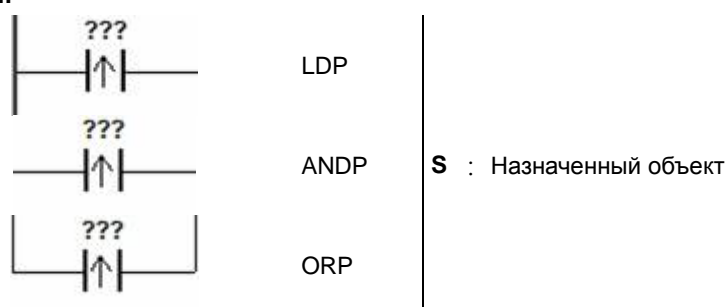


Код инструкции	Операнды	Функция
LDP/ANDP/ORP	S	Запуск обнаружения по переднему фронту / Последовательное подключение обнаружения по переднему фронту / Параллельное подключение обнаружения по переднему фронту

Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D
S	●		●	●	●	○	●	●	●	●	●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

Символ:

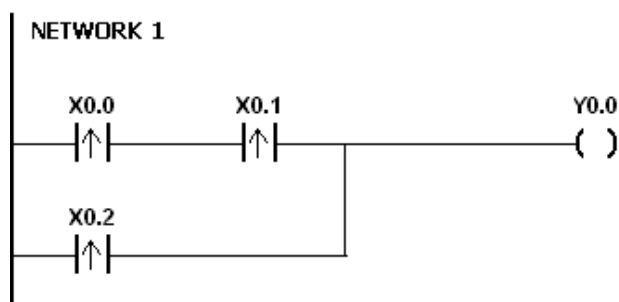


Описание:

1. Инструкция LDP резервирует текущее содержимое и сохраняет обнаружение нового контакта в накопительном регистре.
2. Инструкция ANDP используется для последовательного подключения обнаруженного по переднему фронту контакта.
3. Инструкция ORP используется для одновременного параллельного подключения обнаруженного по переднему фронту контакта.
4. Состояние регистра может быть определено только при сканировании LDP / ANDP / ORP и изменение состояния регистра можно определить только повторным сканированием LDP / ANDP / ORP.
5. В подпрограмме используйте инструкции PED, APED и OPED.

Пример:

1. Первоначальное обнаружение - контакт X0.0, обнаруженный контакт X0.1 подключается последовательно, одновременно обнаруженный контакт X0.2 подключается параллельно, катушка Y0.0 активируется.
2. Когда X0.0 и X0.1 включаются, или когда включается X0., Y0.0 включается для проведения цикла сканирования.

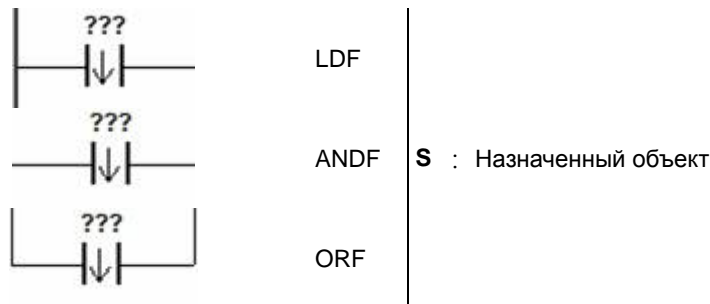


Код инструкции	Операнды	Функция
LDF/ANDF/ORF	S	Запуск обнаружения по заднему фронту / Последовательное подключение обнаружения по заднему фронту / Параллельное подключение обнаружения по заднему фронту

Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D
S	●		●	●	●	○	●	●	●	●	●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

**Символ:**

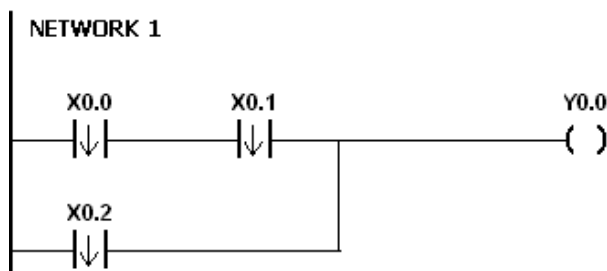


**Описание:**

1. Инструкция LDF резервирует текущее содержимое и сохраняет обнаружение нового контакта в накопительном регистре.
2. Инструкция ANDF используется для последовательного подключения обнаруженного по заднему фронту контакта.
3. Инструкция ORF используется для одновременного параллельного подключения обнаруженного по заднему фронту контакта.
4. Состояние регистра может быть определено только при сканировании LDF / ANDF / ORF и изменение состояния регистра можно определить только повторным сканированием LDF / ANDF / ORF.
5. В подпрограмме используйте инструкции NED, ANED и ONED.

**Пример:**

1. Первоначальное обнаружение - контакт X0.0, обнаруженный контакт X0.1 подключается последовательно, одновременно обнаруженный контакт X0.2 подключается параллельно, катушка Y0.0 активируется.
2. Когда X0.0 и X0.1 включаются, или когда включается X0., Y0.0 включается для проведения цикла сканирования.

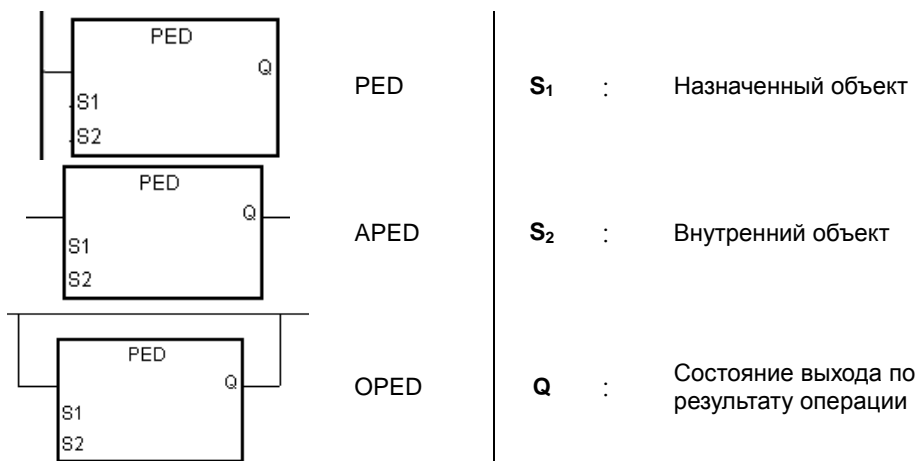


Код инструкции	Операнды	Функция
PED/APED/OPED	$S_1 \cdot S_2$	Запуск обнаружения по переднему фронту / Последовательное подключение обнаружения по переднему фронту / Параллельное подключение обнаружения по переднему фронту

Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D
$S_1$			●	●	●	○	●	●	●	●	●
$S_2$				●	●		●				●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S$	●												

Символ:

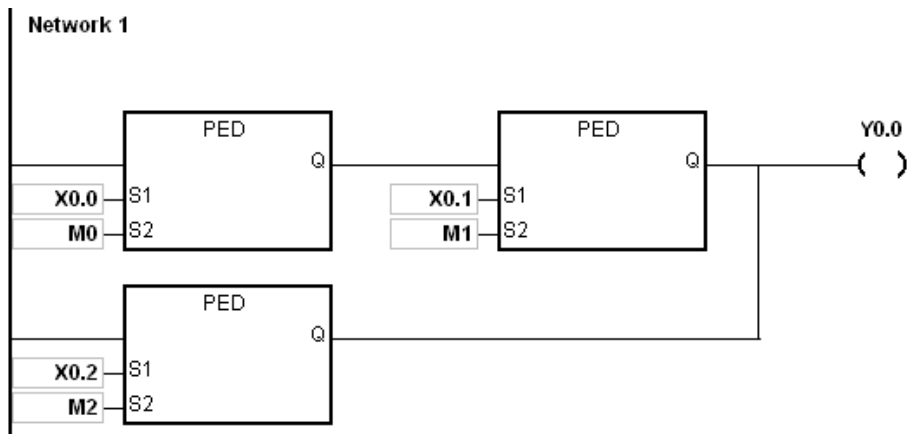


Описание:

1. Инструкции PED / APED / OPED соответствуют инструкциям LDP / ANDP / ORP. Единственная разница между PED / APED / OPED и LDP / ANDP / ORP заключается в том, что пользователю необходимо задать внутренний битовый объект  $S_2$ , в котором сохраняется предыдущее состояние контакта при выполнении инструкций PED / APED / OPED. Не используйте регистр  $S_2$  в программе. В противном случае результат выполнения будет неправильный.
2. Инструкция APED используется для последовательного подключения обнаруженного по переднему фронту контакта.
3. Инструкция OPED используется для одновременного параллельного подключения обнаруженного по переднему фронту контакта.
4. Состояние регистра может быть определено только при сканировании PED / APED / OPED, можно получить состояние регистра, и изменение состояния регистра можно определить только повторным сканированием PED / APED / OPED.
5. PED / APED / OPED можно использовать только в функциональном блоке.
6. Состояние результата операции будет выводиться автоматически после выполнения инструкции. Пользователю не нужно задавать этот объект самостоятельно.

**Пример:**

1. Первоначальное обнаружение - контакт X0.0, обнаруженный контакт X0.1 подключается последовательно, одновременно обнаруженный контакт X0.2 подключается параллельно, катушка Y0.0 активируется.
2. Когда X0.0 и X0.1 включаются, или когда включается X0., Y0.0 включается для проведения цикла сканирования.

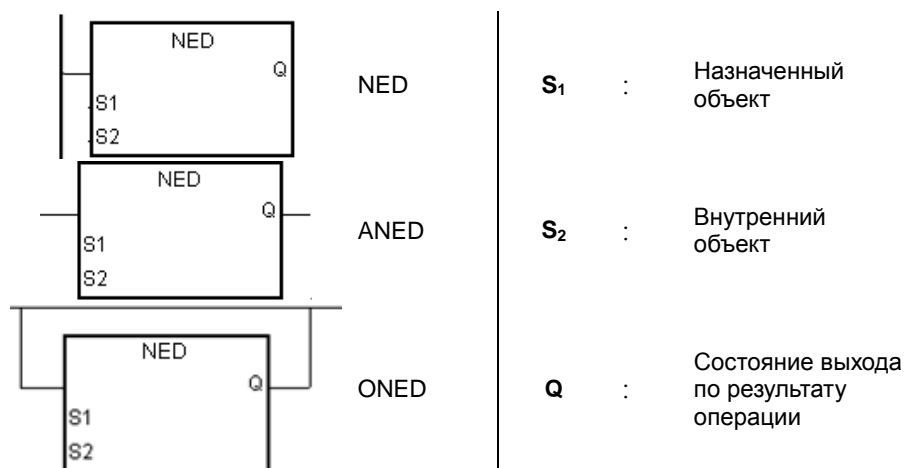


Код инструкции	Операнды	Функция
NED/ANED/ONED	S <sub>1</sub> , S <sub>2</sub>	Запуск обнаружения по заднему фронту / Последовательное подключение обнаружения по заднему фронту / Параллельное подключение обнаружения по заднему фронту

Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D
S <sub>1</sub>			●	●	●	○	●	●	●	●	●
S <sub>2</sub>				●	●		●				●

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

Символ:

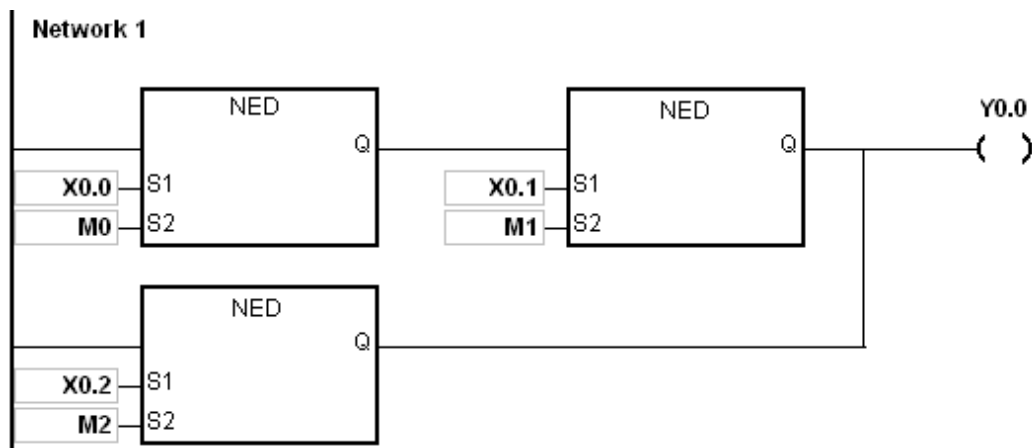


Описание:

1. Инструкции NED/ANED/ONED соответствуют инструкциям LDF/ANDF/ORF. Единственная разница между NED/ANED/ONED и LDF/ANDF/ORF заключается в том, что пользователю необходимо задать внутренний битовый объект S<sub>2</sub>, в котором сохраняется предыдущее состояние контакта при выполнении инструкций NED/ANED/ONED. Не используйте регистр S<sub>2</sub> в программе. В противном случае результат выполнения будет неправильный.
2. Инструкция ANED используется для последовательного подключения обнаруженного по заднему фронту контакта.
3. Инструкция ONED используется для одновременного параллельного подключения обнаруженного по заднему фронту контакта.
4. Состояние регистра может быть определено только при сканировании NED/ANED/ONED, можно получить состояние регистра, и изменение состояния регистра можно определить только повторным сканированием NED/ANED/ONED.
5. Инструкции NED/ANED/ONED можно использовать только в функциональном блоке.
6. Состояние результата операции будет выводиться автоматически после выполнения инструкции. Пользователю не нужно задавать этот объект самостоятельно.

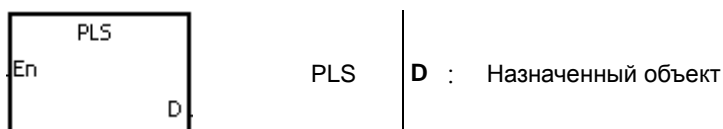
**Пример:**

1. Первоначальное обнаружение - контакт X0.0, обнаруженный контакт X0.1 подключается последовательно, одновременно обнаруженный контакт X0.2 подключается параллельно, катушка Y0.0 активируется.
2. Когда X0.0 и X0.1 включаются, или когда включается X0., Y0.0 включается для проведения цикла сканирования.



Код инструкции		Операнды										Функция	
PLS		D										Импульс по переднему фронту	
Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D		
D				●	●	○	●						
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●												

Символ:

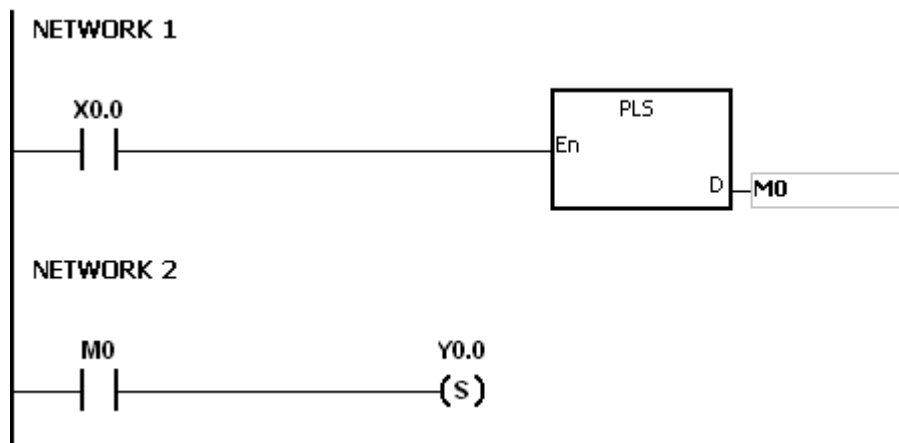


Описание:

1. Когда заданный контакт включается, выполняется команда PLS, назначенный объект **D** отправляет импульс для цикла сканирования.
2. Не используйте инструкцию PLS в функциональном блоке.

Пример:

При включении X0.0, M0 включается для цикла сканирования. Если включен M0, Y0.0 также включается.



Временная диаграмма:



Код инструкции		Операнды										Функция	
PLF		D										Импульс по заднему фронту	
Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D		
D				●	●	○	●						
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

Символ:

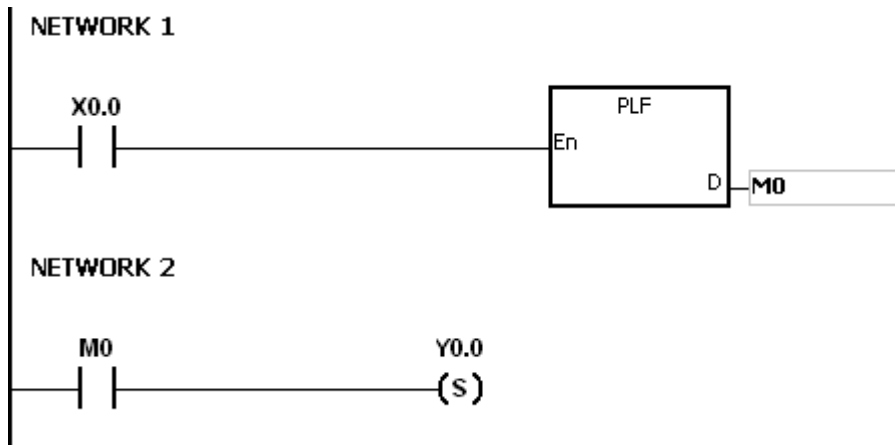


Описание:

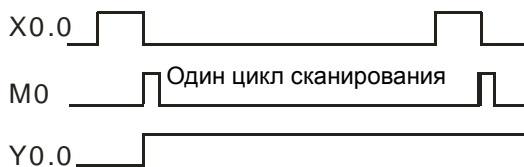
1. Когда заданный контакт включается, выполняется команда PLF, назначенный объект D отправляет импульс для цикла сканирования.
2. Не используйте инструкцию PLF в функциональном блоке.

Пример:

При включении X0.0, M0 включается для цикла сканирования. Если включен M0, Y0.0 также включается.



Временная диаграмма:





Код инструкции	Операнды	Функция
INV	-	Инвертирование результата логической операции

**Символ:**

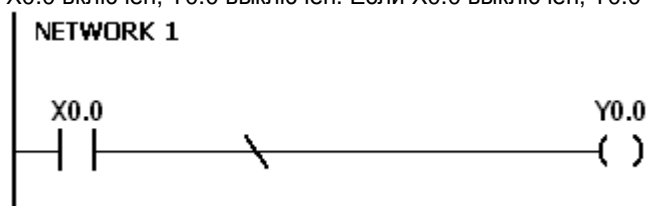


**Описание:**

Результат логической операции, предшествующий инструкции INV, инвертируется, и результат инверсии сохраняется в накопительном регистре.

**Пример:**

Если X0.0 включен, Y0.0 выключен. Если X0.0 выключен, Y0.0 включен.



Код инструкции	Операнды	Функция
NP	-	Цепь переключается по переднему фронту.

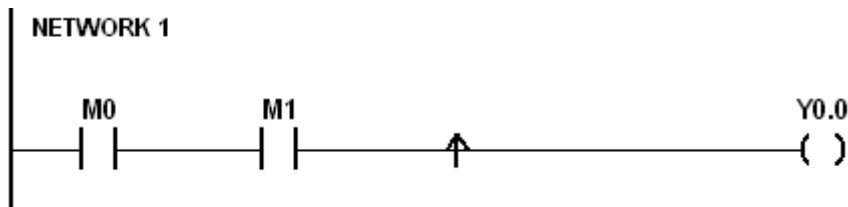
Символ:



Описание:

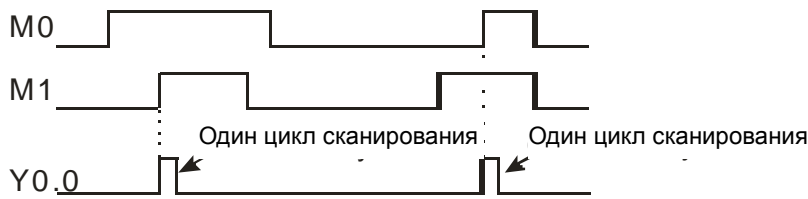
1. Когда значение в накопительном регистре изменяется от 0 до 1, инструкция NP сохраняет значение 1 в накопительном регистре для цикла сканирования. После завершения второго цикла сканирования значение в накопительном регистре изменяется на 0.
2. Обратите внимание, в функциональном блоке используется инструкция FB\_NP.

Пример:



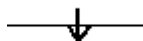
Инструкция:	Операция:
LD M0	Контакт А для M0 подключен.
AND M1	Контакт А для M1 соединен последовательно.
NP	Цепь переключается по переднему фронту.
OUT Y0.0	Катушка Y0.0 активируется.

Временная диаграмма:



Код инструкции	Операнды	Функция
PN	-	Цепь переключается по заднему фронту.

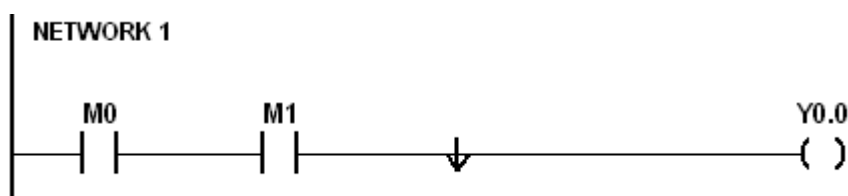
**Символ:**



**Описание:**

1. Когда значение в накопительном регистре изменяется от 1 до 0, инструкция PN сохраняет значение 1 в накопительном регистре для цикла сканирования. После завершения второго цикла сканирования значение в накопительном регистре изменяется на 0.
2. Обратите внимание, в функциональном блоке используется инструкция FB\_PN.

**Пример:**



Инструкция:	Операция:
LD M0	Контакт А для M0 подключен.
AND M1	Контакт А для M1 соединен последовательно.
PN	Цепь переключается по заднему фронту.
OUT Y0.0	Катушка Y0.0 активируется.

**Временная диаграмма:**



Код инструкции		Операнды								Функция			
FB_NP		<b>S</b>								Цепь переключается по переднему фронту.			
Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D		
<b>S</b>				●	●		●				●		
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>	●												

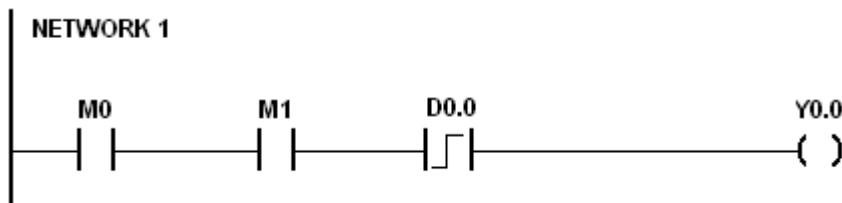
**Символ:**



**Описание:**

1. Когда значение в накопительном регистре переходит от 0 к 1, инструкция FB\_NP сохраняет значение 1 в накопительном регистре для цикла сканирования. После завершения второго цикла сканирования значение в накопительном регистре изменяется на 0.
2. Предыдущее состояние контакта сохраняется во внутреннем битовом объекте **S**. Пожалуйста, не используйте **S** повторно в программе. В противном случае результат выполнения будет неправильный.
3. Инструкция FB\_NP может использоваться только в функциональном блоке.

**Пример:**



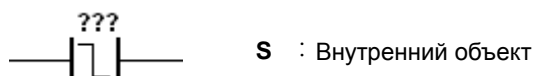
Инструкция:		Операция:
LD	M0	Контакт А для M0 подключен.
AND	M1	Контакт А для M1 соединен последовательно.
<b>FB_NP</b>	D0.0	Цепь переключается по переднему фронту.
OUT	Y0.0	Катушка Y0.0 активируется.

**Временная диаграмма:**



Код инструкции		Операнды							Функция				
FB_PN		S							The circuit is falling edge-triggered.				
Объекты	DX	DY	X	Y	M	SM	S	T	C	HC	D		
S				●	●		●				●		
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												

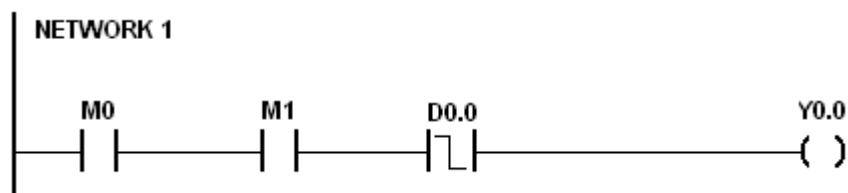
Символ:



Описание:

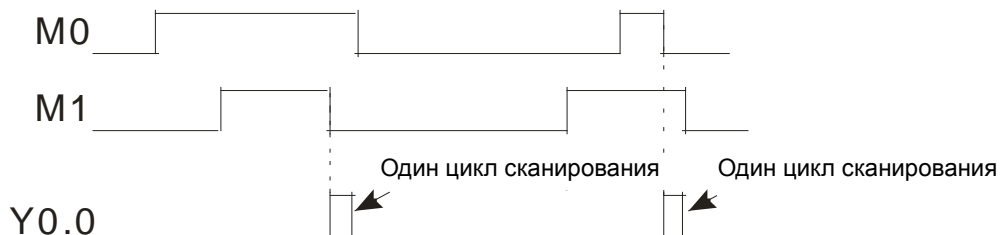
1. Когда значение в накопительном регистре переходит от 1 к 0, инструкция FB\_PN сохраняет значение 1 в накопительном регистре для цикла сканирования. После завершения второго цикла сканирования значение в накопительном регистре изменяется на 0.
2. Предыдущее состояние контакта сохраняется во внутреннем битовом объекте **S**. Пожалуйста, не используйте **S** повторно в программе. В противном случае результат выполнения будет неправильный.
3. Инструкция FB\_PN может использоваться только в функциональном блоке.

Пример:



Инструкция:		Операция:
LD	M0	Контакт A для M0 подключен.
AND	M1	Контакт A для M1 соединен последовательно.
<b>FB_PN</b>	D0.0	Цепь переключается по заднему фронту.
OUT	Y0.0	Катушка Y0.0 активируется.

Временная диаграмма:



---

## Глава 6 Прикладные инструкции

### Содержание

<b>6.1 Инструкции сравнения</b> .....	<b>6-3</b>
6.1.1 Описание инструкций сравнения .....	6-3
<b>6.2 Арифметические инструкции</b> .....	<b>6-43</b>
6.2.1 Описание арифметических инструкций .....	6-43
<b>6.3 Инструкции преобразования данных</b> .....	<b>6-74</b>
6.3.1 Описание инструкций преобразования данных .....	6-74
<b>6.4 Инструкции передачи данных</b> .....	<b>6-115</b>
6.4.1 Описание инструкций передачи данных .....	6-115
<b>6.5 Инструкции перехода</b> .....	<b>6-143</b>
6.5.1 Описание инструкций перехода .....	6-143
<b>6.6 Исполняемые программой инструкции</b> .....	<b>6-151</b>
6.6.1 Описание исполняемых программой инструкций .....	6-151
<b>6.7 Инструкции обновления входов/выходов</b> .....	<b>6-163</b>
6.7.1 Описание инструкций обновления входов/выходов.....	6-163
<b>6.8 Вспомогательные инструкции</b> .....	<b>6-168</b>
6.8.1 Описание вспомогательных инструкций .....	6-168
<b>6.9 Логические инструкции</b> .....	<b>6-216</b>
6.9.1 Описание логических инструкций .....	6-216
<b>6.10 Инструкции вращения</b> .....	<b>6-237</b>
6.10.1 Описание инструкций вращения .....	6-237
<b>6.11 Инструкции таймеров и счетчиков</b> .....	<b>6-248</b>
6.11.1 Описание инструкций таймеров и счетчиков.....	6-248
<b>6.12 Инструкции сдвига</b> .....	<b>6-283</b>
6.12.1 Описание инструкций сдвига .....	6-283
<b>6.13 Инструкции по обработке данных</b> .....	<b>6-320</b>
6.13.1 Описание инструкций по обработке данных.....	6-320
<b>6.14 Инструкции по созданию структуры</b> .....	<b>6-377</b>
6.14.1 Описание инструкций по созданию структуры.....	6-377

<b>6.15 Инструкции модулей</b> .....	<b>6-385</b>
6.15.1 Описание инструкций модулей .....	6-385
<b>6.16 Инструкции значений с плавающей запятой</b> .....	<b>6-420</b>
6.16.1 Описание инструкций значения с плавающей запятой.....	6-420
<b>6.17 Инструкции для часов реального времени</b> .....	<b>6-455</b>
6.17.1 Описание инструкций для часов реального времени .....	6-455
<b>6.18 Инструкции периферийных устройств</b> .....	<b>6-485</b>
6.18.1 Описание инструкций периферийных устройств .....	6-485
<b>6.19 Инструкции коммуникации</b> .....	<b>6-501</b>
6.19.1 Описание инструкций коммуникации .....	6-501
6.19.2 Описание флагов и регистров для коммуникации.....	6-584
<b>6.20 Прочие инструкции</b> .....	<b>6-587</b>
6.20.1 Описание прочих инструкций .....	6-587
<b>6.21 Инструкции по обработке строковых переменных</b> .....	<b>6-599</b>
6.21.1 Описание инструкций по обработке строковых переменных .....	6-599
<b>6.22 Инструкции для Ethernet</b> .....	<b>6-659</b>
6.22.1 Описание инструкций для Ethernet .....	6-659
<b>6.23 Инструкции для карты памяти</b> .....	<b>6-701</b>
6.23.1 Описание инструкций для карты памяти.....	6-701
<b>6.24 Инструкции управления задачами</b> .....	<b>6-718</b>
6.24.1 Описание инструкций управления задачами .....	6-718
<b>6.25 Инструкции SFC</b> .....	<b>6-722</b>
6.25.1 Описание инструкций SFC .....	6-722
<b>6.26 Инструкции высокочастотных выходов</b> .....	<b>6-729</b>
6.26.1 Описание инструкций высокочастотных выходов .....	6-729
<b>6.27 Инструкции коммуникации CANopen</b> .....	<b>6-825</b>
6.27.1 Описание инструкций коммуникации CANopen .....	6-825

## 6.1 Инструкции сравнения

### 6.1.1 Описание инструкций сравнения

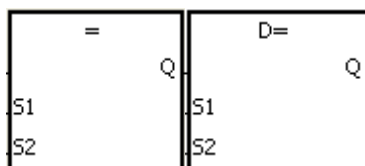
API	Код инструкции			Операнды								Функция					
0000~0005	D	LD※		$S_1 \cdot S_2$								Сравнение значений LD※					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ:



$S_1$  : Источник данных 1

$S_2$  : Источник данных 1

Инструкции для примера: LD= и DLD=

Описание:

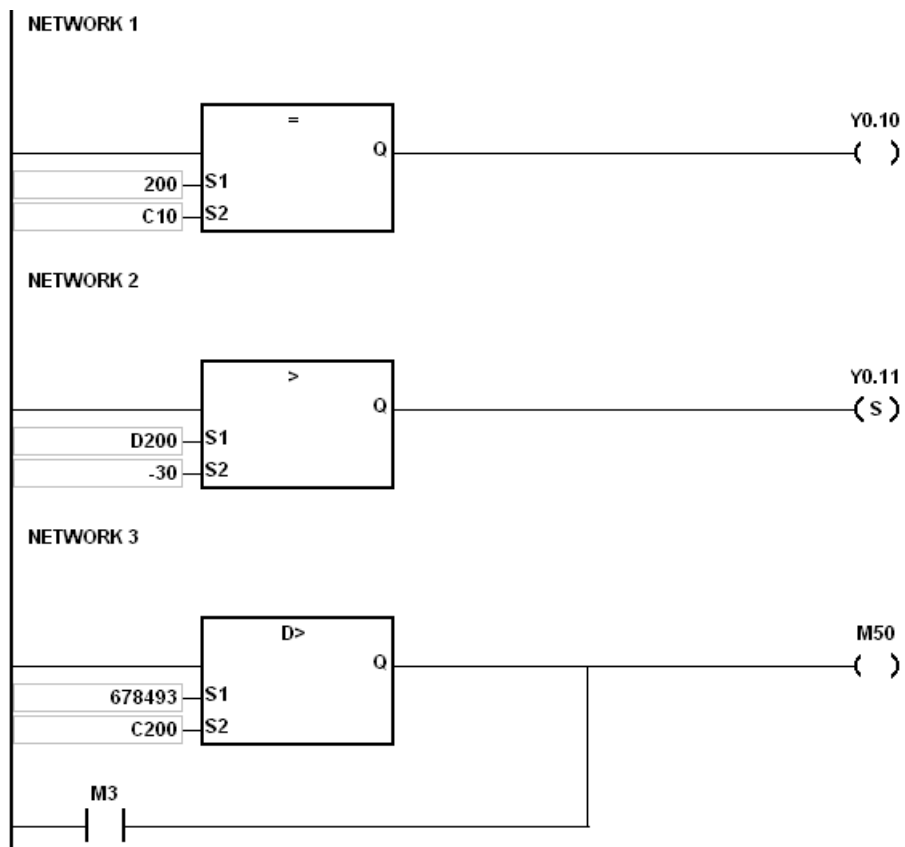
- Данные инструкции сравнивают значения  $S_1$  и  $S_2$ . Возьмем для примера инструкцию LD=. Если результатом сравнения является равенство значений  $S_1$  и  $S_2$ , это является условием непрерывности цепи. Если результатом сравнения является неравенство значений  $S_1$  и  $S_2$ , это является условием прерывания цепи.
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

API	16-битная инструкция	32-битная инструкция	Условие непрерывности	Условие прерывания
0000	LD =	DLD =	$S_1 = S_2$	$S_1 \neq S_2$
0001	LD < >	DLD < >	$S_1 \neq S_2$	$S_1 = S_2$
0002	LD >	DLD >	$S_1 > S_2$	$S_1 \leq S_2$
0003	LD > =	DLD > =	$S_1 \geq S_2$	$S_1 < S_2$
0004	LD <	DLD <	$S_1 < S_2$	$S_1 \geq S_2$
0005	LD < =	DLD < =	$S_1 \leq S_2$	$S_1 > S_2$

Пример:

- Если C10 равен 200, Y0.10 включается.
- Когда значение в D200 больше -30, Y0.11 сохраняет включенное состояние.
- Когда значение (C201, C200) меньше 678,493 или включен M3, M50 включается.





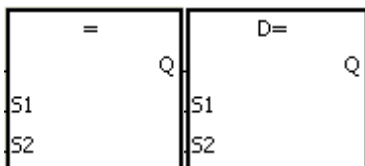
API	Код инструкции			Операнды								Функция				
0006~0011	D	AND※		$S_1 \cdot S_2$								Сравнение значений AND※				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●				●	●	
S <sub>2</sub>		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ:



S<sub>1</sub> : Источник данных 1

S<sub>2</sub> : Источник данных 2

Инструкции для примера: AND= и DAND=

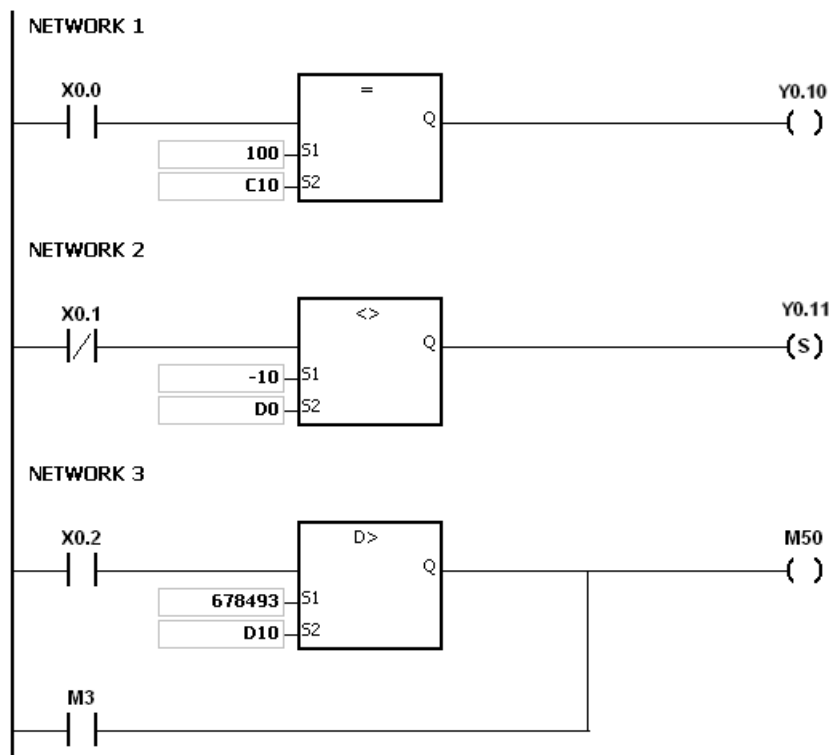
Описание:

- Данные инструкции сравнивают значения S<sub>1</sub> и S<sub>2</sub>. Возьмем для примера инструкцию AND=. Если результатом сравнения является равенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием непрерывности цепи. Если результатом сравнения является неравенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием прерывания цепи.
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

API	16-битная инструкция	32-битная инструкция	Условие непрерывности	Условие прерывания
0006	AND =	DAND =	$S_1 = S_2$	$S_1 \neq S_2$
0007	AND < >	DAND < >	$S_1 \neq S_2$	$S_1 = S_2$
0008	AND >	DAND >	$S_1 > S_2$	$S_1 \leq S_2$
0009	AND > =	DAND > =	$S_1 \geq S_2$	$S_1 < S_2$
0010	AND <	DAND <	$S_1 < S_2$	$S_1 \geq S_2$
0011	AND < =	DAND < =	$S_1 \leq S_2$	$S_1 > S_2$

Пример:

- Если X0.0 включен и текущее значение в C10 равно 100, Y0.10 включается.
- Если X0.1 выключен и значение в D0 не равно -10, Y0.11 сохраняет включенное состояние.
- Если X0.2 включен и значение в (D11, D10) меньше 678,493 или если M3 включен, M50 включается.



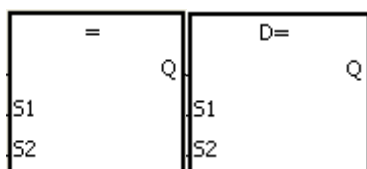
API	Код инструкции			Операнды								Функция				
0012~0017	D	OR※		$S_1 \cdot S_2$								Сравнение значений OR※				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ:



$S_1$  : Источник данных 1

$S_2$  : Источник данных 2

Инструкции для примера: OR= и DOR=

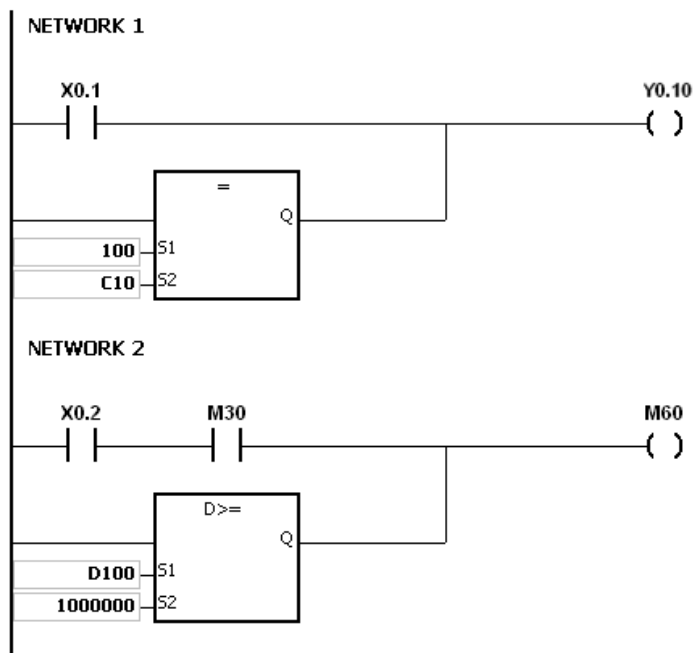
Описание:

1. Данные инструкции сравнивают значения  $S_1$  и  $S_2$ . Возьмем для примера инструкцию OR=. Если результатом сравнения является равенство значений  $S_1$  и  $S_2$ , это является условием непрерывности цепи. Если результатом сравнения является неравенство значений  $S_1$  и  $S_2$ , это является условием прерывания цепи.
2. 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

API	16-битная инструкция	32-битная инструкция	Условие непрерывности	Условие прерывания
0012	OR =	DOR =	$S_1 = S_2$	$S_1 \neq S_2$
0013	OR < >	DOR < >	$S_1 \neq S_2$	$S_1 = S_2$
0014	OR >	DOR >	$S_1 > S_2$	$S_1 \leq S_2$
0015	OR > =	DOR > =	$S_1 \geq S_2$	$S_1 < S_2$
0016	OR <	DOR <	$S_1 < S_2$	$S_1 \geq S_2$
0017	OR < =	DOR < =	$S_1 \leq S_2$	$S_1 > S_2$

Пример:

1. Если X0.1 включен или текущее значение в C10 равно 100, Y0.10 включается.
2. Когда X0.2 и M30 включены или значение (D101, D100) больше или равно 1000,000, M60 включается.



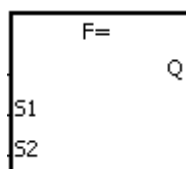
API	Код инструкции			Операнды								Функция				
0018~0023	FLD※			$S_1 \cdot S_2$								Comparing the floating-point numbers LD※				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○					○
$S_2$	●	●			●	●	●	●	●		○					○

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символ:



$S_1$  : Источник данных 1

$S_2$  : Источник данных 2

Инструкции для примера: FLD= и DFLD=

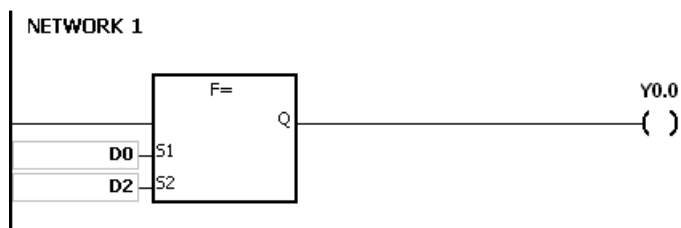
Описание:

1. Эти инструкции представляют собой 32-битные инструкции сравнения с плавающей запятой с одним разрядом.
2. Данные инструкции сравнивают значения  $S_1$  и  $S_2$ , являющиеся значениями с плавающей запятой. Возьмем для примера инструкцию FLD=. Если результатом сравнения является равенство значений  $S_1$  и  $S_2$ , это является условием непрерывности цепи. Если результатом сравнения является неравенство значений  $S_1$  и  $S_2$ , это является условием прерывания цепи.

API	16-битная инструкция	Условие непрерывности	Условие прерывания
0018	FLD =	$S_1 = S_2$	$S_1 \neq S_2$
0019	FLD < >	$S_1 \neq S_2$	$S_1 = S_2$
0020	FLD >	$S_1 > S_2$	$S_1 \leq S_2$
0021	FLD > =	$S_1 \geq S_2$	$S_1 < S_2$
0022	FLD <	$S_1 < S_2$	$S_1 \geq S_2$
0023	FLD < =	$S_1 \leq S_2$	$S_1 > S_2$

**Пример:**

Инструкция для примера FLD = . Если значение в D0 равно значению в D2, Y0.0 включается.

**Примечание:**

1. Если значения  $S_1$  или  $S_2$  превышают допустимый диапазон для значений с плавающей запятой, контакт отключается, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2013.

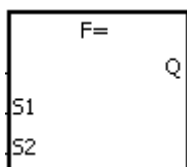
API	Код инструкции			Операнды								Функция					
0024~0029	FAND※			$S_1 \cdot S_2$								Сравнение чисел с плавающей запятой AND※					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○					○
$S_2$	●	●			●	●	●	●	●		○					○

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символ:



$S_1$  : Источник данных 1

$S_2$  : Источник данных 2

Инструкции для примера: FAND= и DFAND=

Описание:

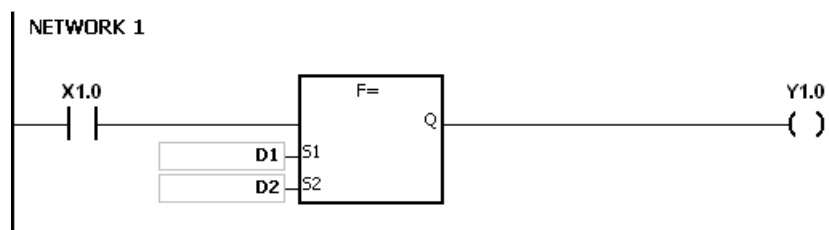
1. Эти инструкции представляют собой 32-битные инструкции сравнения с плавающей запятой.
2. Данные инструкции сравнивают значения  $S_1$  и  $S_2$ , являющиеся значениями с плавающей запятой. Возьмем для примера инструкцию FAND=. Если результатом сравнения является равенство значений  $S_1$  и  $S_2$ , это является условием непрерывности цепи. Если результатом сравнения является неравенство значений  $S_1$  и  $S_2$ , это является условием прерывания цепи.

API	16-битная инструкция	Условие непрерывности	Условие прерывания
0024	FAND =	$S_1 = S_2$	$S_1 \neq S_2$
0025	FAND < >	$S_1 \neq S_2$	$S_1 = S_2$
0026	FAND >	$S_1 > S_2$	$S_1 \leq S_2$
0027	FAND > =	$S_1 \geq S_2$	$S_1 < S_2$
0028	FAND <	$S_1 < S_2$	$S_1 \geq S_2$
0029	FAND < =	$S_1 \leq S_2$	$S_1 > S_2$



**Пример:**

Инструкция для примера FAND = . Если X1.0 включен и значение в D1 равно значению в D2, Y1.0 включается.

**Примечание:**

1. Если значения  $S_1$  или  $S_2$  превышают допустимый диапазон для значений с плавающей запятой, контакт отключается, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2013.

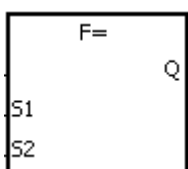
API	Код инструкции			Операнды							Функция						
0030~ 0035	FOR※			$S_1 \cdot S_2$							Сравнение чисел с плавающей запятой OR※						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○					○
$S_2$	●	●			●	●	●	●	●		○					○

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символ:



$S_1$  : Источник данных 1

$S_2$  : Источник данных 2

Инструкции для примера: FOR= и DFOR=

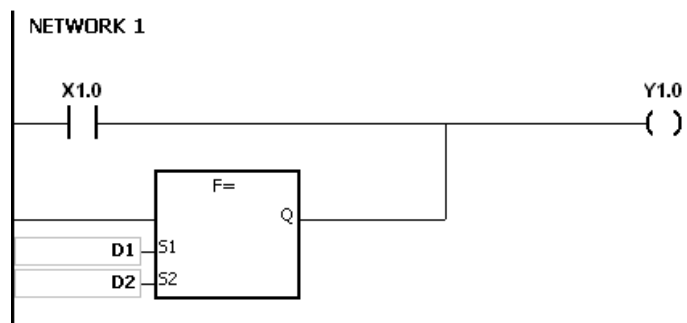
Описание:

1. Эти инструкции представляют собой 32-битные инструкции сравнения с плавающей запятой.
2. Данные инструкции сравнивают значения  $S_1$  и  $S_2$ , являющиеся значениями с плавающей запятой. Возьмем для примера инструкцию FOR=. Если результатом сравнения является равенство значений  $S_1$  и  $S_2$ , это является условием непрерывности цепи. Если результатом сравнения является неравенство значений  $S_1$  и  $S_2$ , это является условием прерывания цепи.

API	16-битная инструкция	Условие непрерывности	Условие прерывания
0030	FOR =	$S_1 = S_2$	$S_1 \neq S_2$
0031	FOR < >	$S_1 \neq S_2$	$S_1 = S_2$
0032	FOR >	$S_1 > S_2$	$S_1 \leq S_2$
0033	FOR > =	$S_1 \geq S_2$	$S_1 < S_2$
0034	FOR <	$S_1 < S_2$	$S_1 \geq S_2$
0035	FOR < =	$S_1 \leq S_2$	$S_1 > S_2$

**Пример:**

Если X1.0 включен или если значение в in D1 равно значению в D2, Y1.0 включается.

**Примечание:**

1. Если значения  $S_1$  или  $S_2$  превышают допустимый диапазон для значений с плавающей запятой, контакт отключается, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2013.

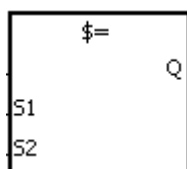
API	Код инструкции			Операнды							Функция						
0036~0037		LD\$※		$S_1 \cdot S_2$							Сравнение строковых переменных LD\$※						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●		●	●						○	
S <sub>2</sub>	●	●			●	●		●	●						○	

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>													●
S <sub>2</sub>													●

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



S<sub>1</sub> : Источник данных 1

S<sub>2</sub> : Источник данных 2

Инструкция для примера:

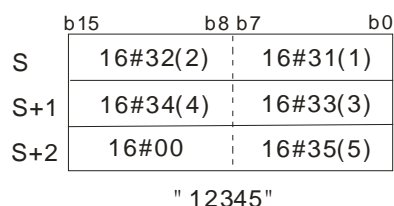
LD\$=

Описание:

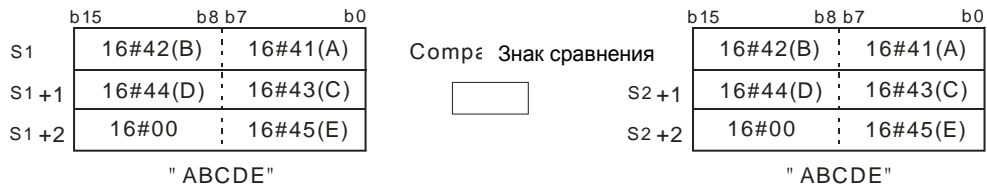
1. Эти инструкции сравнивают данные в S<sub>1</sub> с данными в S<sub>2</sub>, данные представляют собой строковые переменные.
2. Операнды S<sub>1</sub> и S<sub>2</sub> могут содержать до 256 символов (до конечного символа 16#00).
3. Возьмем для примера инструкцию LD\$=. Если результатом сравнения является равенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием непрерывности цепи. Если результатом сравнения является неравенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием прерывания цепи.

API	16-битная инструкция	Условие непрерывности	Условие прерывания
0036	LD\$ =	S <sub>1</sub> = S <sub>2</sub>	S <sub>1</sub> ≠ S <sub>2</sub>
0037	LD\$ < >	S <sub>1</sub> ≠ S <sub>2</sub>	S <sub>1</sub> = S <sub>2</sub>

4. Только когда данные в S~S+n (значение n показывает n<sup>й</sup> объект, до 256 значений) включают в себя значение 16#00, данные считаются полной строковой переменной. Например:



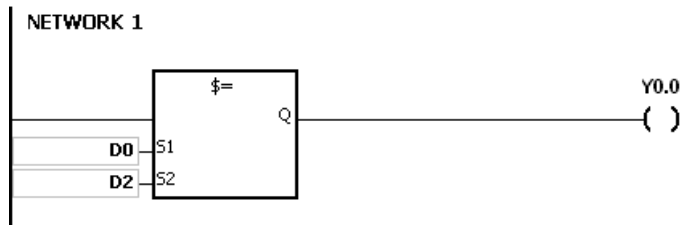
5. При совпадении двух строковых переменных результат сравнения приведен ниже:



Символ сравнения	Результат сравнения
\$ =	Непрерывность
\$ < >	Прерывание

**Пример:**

Когда строковая переменная в диапазоне D0~16#00 равна строковой переменной в диапазоне D2~16#00, Y0.0 включается.



**Примечание:**

1. Если строковая переменная содержит более 256 символов или не заканчивается символом 16 # 00, инструкция не будет выполнена, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16 # 200E.
2. Во время сравнения строковых переменных, когда отображается конечный символ 16 # 00, таким образом отмечается конец строковой переменной.

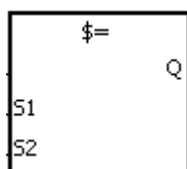
API	Код инструкции	Операнды	Функция
0042~0043	AND\$※	S <sub>1</sub> · S <sub>2</sub>	Сравнение строковых переменных AND\$※

Объект	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●		●	●						○	
S <sub>2</sub>	●	●			●	●		●	●						○	

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>													●
S <sub>2</sub>													●

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



S<sub>1</sub> : Источник данных 1

S<sub>2</sub> : Источник данных 2

Инструкция для примера AND\$=

Описание:

1. Эти инструкции сравнивают данные в S<sub>1</sub> с данными в S<sub>2</sub>, данные представляют собой строковые переменные.
2. Операнды S<sub>1</sub> и S<sub>2</sub> могут содержать до 256 символов (до конечного символа 16#00).
3. Возьмем для примера инструкцию AND\$=. Если результатом сравнения является равенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием непрерывности цепи. Если результатом сравнения является неравенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием прерывания цепи.

API	16-битная инструкция	Условие непрерывности	Условие прерывания
0042	AND\$ =	S <sub>1</sub> = S <sub>2</sub>	S <sub>1</sub> ≠ S <sub>2</sub>
0043	AND\$ < >	S <sub>1</sub> ≠ S <sub>2</sub>	S <sub>1</sub> = S <sub>2</sub>

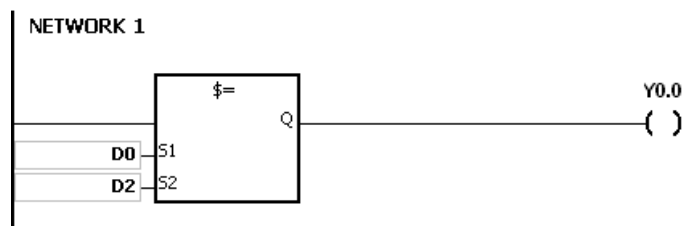
4. Только когда данные в S~S+n (значение n показывает n<sup>й</sup> объект, до 256 значений) включают в себя значение 16#00, данные считаются полной строковой переменной.
5. При совпадении двух строковых переменных результат сравнения приведен ниже:



Символ сравнения	Результат сравнения
\$ =	Непрерывность
\$ < >	Прерывание

**Пример:**

Когда строковая переменная в диапазоне D0~16#00 равна строковой переменной в диапазоне D2~16#00, Y0.0 включается.

**Примечание:**

1. Если строковая переменная содержит более 256 символов или не заканчивается символом 16 # 00, инструкция не будет выполнена, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16 # 200E.
2. Во время сравнения строковых переменных, когда отображается конечный символ 16 # 00, таким образом отмечается конец строковой переменной.

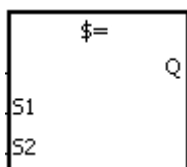
API	Код инструкции	Операнды	Функция
0048~0049	OR\$※	S <sub>1</sub> · S <sub>2</sub>	Сравнение строковых переменных OR\$※

Объект	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●		●	●						○	
S <sub>2</sub>	●	●			●	●		●	●						○	

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>													●
S <sub>2</sub>													●

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ :



S<sub>1</sub> : Источник данных 1  
S<sub>2</sub> : Источник данных 2

Инструкция для примера:  
OR\$=

**Описание:**

1. Эти инструкции сравнивают данные в S<sub>1</sub> с данными в S<sub>2</sub>, данные представляют собой строковые переменные.
2. Операнды S<sub>1</sub> и S<sub>2</sub> могут содержать до 256 символов (до конечного символа 16#00).
3. Возьмем для примера инструкцию OR\$=. Если результатом сравнения является равенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием непрерывности цепи. Если результатом сравнения является неравенство значений S<sub>1</sub> и S<sub>2</sub>, это является условием прерывания цепи.

API	16-битная инструкция	Условие непрерывности	Условие прерывания
0048	OR\$ =	S <sub>1</sub> = S <sub>2</sub>	S <sub>1</sub> ≠ S <sub>2</sub>
0049	OR\$ < >	S <sub>1</sub> ≠ S <sub>2</sub>	S <sub>1</sub> = S <sub>2</sub>

4. Только когда данные в S~S+n (значение n показывает n<sup>й</sup> объект, до 256 значений) включают в себя значение 16#00, данные считаются полной строковой переменной.
5. При совпадении двух строковых переменных результат сравнения приведен ниже:

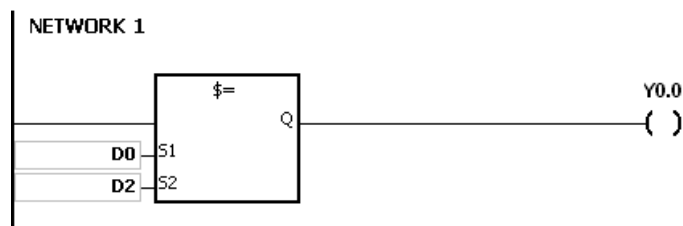




Символ сравнения	Результат сравнения
\$ =	Непрерывность
\$ < >	Прерывание

**Пример:**

Когда строковая переменная в диапазоне D0~16#00 равна строковой переменной в диапазоне D2~16#00, Y0.0 включается.

**Примечание:**

1. Если строковая переменная содержит более 256 символов или не заканчивается символом 16 # 00, инструкция не будет выполнена, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16 # 200E.
2. Во время сравнения строковых переменных, когда отображается конечный символ 16 # 00, таким образом отмечается конец строковой переменной.

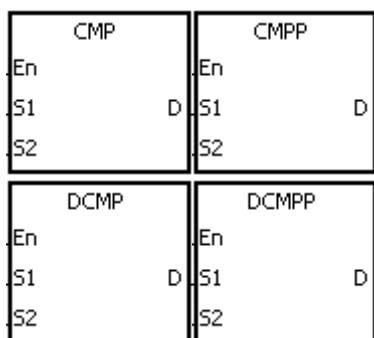
API	Код инструкции			Операнды							Функция							
0054	D	CMP		P	$S_1 \cdot S_2 \cdot D$							Сравнение значений						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символ :



$S_1$  : Сравниваемое значение 1

$S_2$  : Сравниваемое значение 2

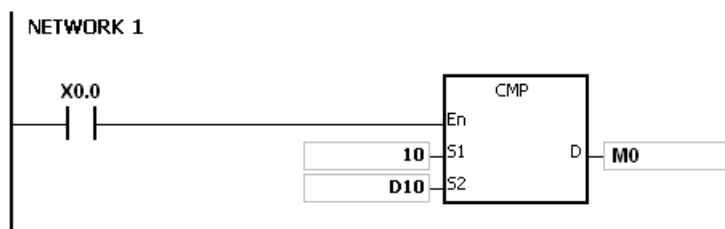
D : Результат сравнения

Описание :

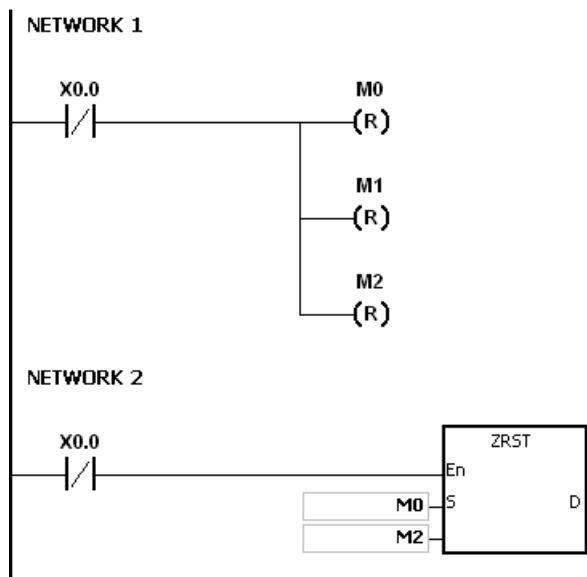
- Инструкция применяется для сравнения значений в  $S_1$  и в  $S_2$ , формат сравниваемых значений - десятичный. Результат сравнения сохраняется в D.
- Операнд D занимает 3 последовательных регистра. Результат сравнения сохраняется в D, D+1 и D+2. Если сравниваемое значение в  $S_1$  больше сравниваемого значения в  $S_2$ , включается D. Если значение в  $S_1$  равно значению в  $S_2$ , включается D+1. Если значение в  $S_1$  меньше значения в  $S_2$ , включается D+2.
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

Пример:

- Если операндом D является M0, результат сравнения сохраняется в M0, M1 и M2, как показано ниже.
- Когда X0.0 включен, инструкция CMP выполняется. M0, M1 или M2 также включается. Если X0.0 выключается, выполнение инструкции CMP останавливается. Состояние M0, M1 и M2 остается неизменным.



3. Если необходимо очистить результат сравнения, используйте инструкции RST или ZRST.



**Примечание:**

1. Если операнд **D** задается в ПО ISPSOft, тип данных будет: ARRAY [3] для BOOL.
2. Если значение **D+2** выходит за допустимый предел, инструкция выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2003.

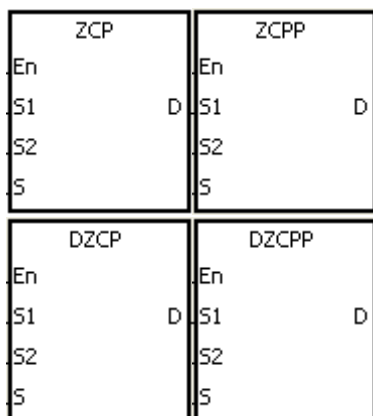
API	Код инструкции			Операнды							Функция						
0055	D	ZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$							Зонное сравнение						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
$S$	●	●			●	●	●	●	●		○	○	○	○		
$D$		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
$S$		●	●		●	●	●				●	●	
$D$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символ :



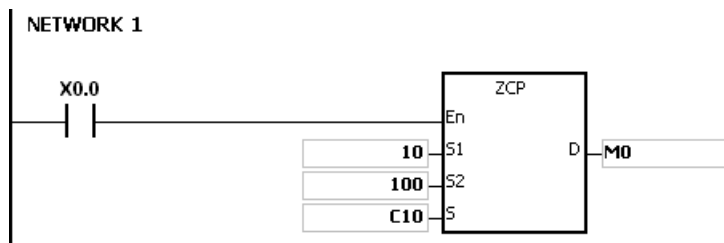
- $S_1$  : Минимальное значение зонного сравнения
- $S_2$  : Максимальное значение зонного сравнения
- $S$  : Сравнимое значение
- $D$  : Результат сравнения

Описание :

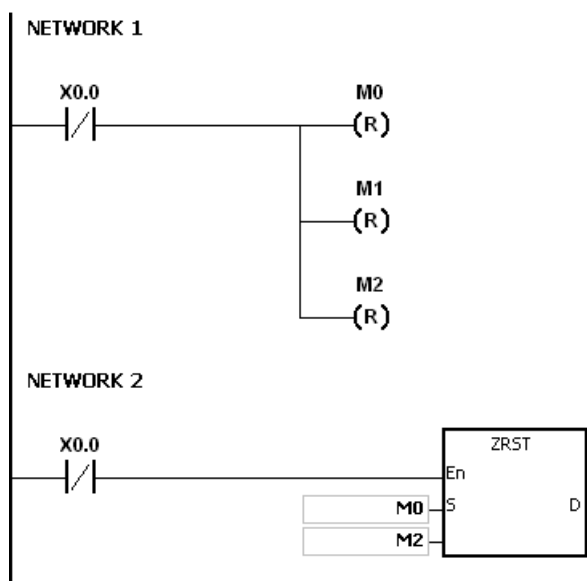
- Эта инструкция сравнивает значение в  $S$  со значением в  $S_1$ , а также значение в  $S$  со значением в  $S_2$ . Сравнимые значения являются десятичными, результат сравнения сохраняется в  $D$ .
- Значение в  $S_1$  должно быть меньше значения в  $S_2$ . Если значение в  $S_1$  больше значения в  $S_2$ ,  $S_1$  будет определяться как максимальное/минимальное значение при выполнении инструкции ZCP.
- Операнд  $D$  занимает 3 последовательных регистра. Результат сравнения сохраняется в  $D$ ,  $D+1$  и  $D+2$ . Если сравниваемое значение в  $S_1$  больше сравниваемого значения в  $S_2$ , включается  $D$ . Если значение в  $S_1$  равно значению в  $S_2$ , включается  $D+1$ . Если значение в  $S_1$  меньше значения в  $S_2$ , включается  $D+2$ .
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

**Пример:**

1. Если операндом **D** является M0, результат сравнения сохраняется в M0, M1 и M2, как показано ниже.
2. Когда X0.0 включен, инструкция ZCP выполняется. M0, M1 или M2 также включается. Если X0.0 выключается, выполнение инструкции ZCP останавливается. Состояние M0, M1 и M2 остается неизменным.



3. Если необходимо очистить результат сравнения, используйте инструкции RST или ZRST.



**Примечание:**

1. Если операнд **D** задается в ПО ISPSOft, тип данных будет: ARRAY [3] для BOOL.
2. Если значение **D+2** выходит за допустимый предел, инструкция выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2003.

6

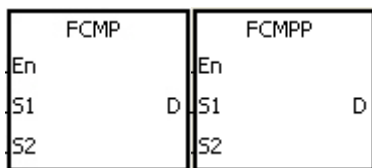
API	Код инструкции			Операнды							Функция						
0056		FCMP	P	$S_1 \cdot S_2 \cdot D$							Сравнение чисел с плавающей запятой						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○					○
$S_2$	●	●			●	●	●	●	●		○					○
D		●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символ :**



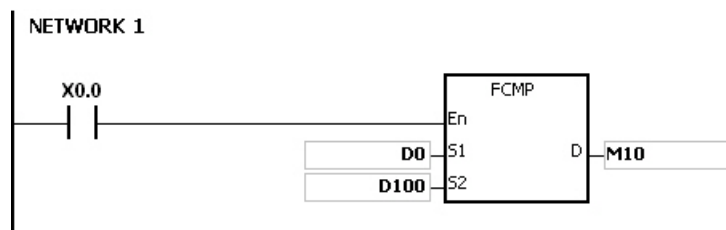
- $S_1$  : Сравнимое значение с плавающей запятой 1
- $S_2$  : Сравнимое значение с плавающей запятой 2
- D : Результат сравнения

**Описание :**

- Инструкция FCMP сравнивает число с плавающей запятой в  $S_1$  с числом с плавающей запятой в  $S_2$ . Результат сравнения ( $>$  ·  $=$  ·  $<$ ) сохраняется в D.
- Операнд D занимает 3 последовательных регистра. Результат сравнения сохраняется в D, D+1 и D+2. Если сравниваемое значение в  $S_1$  больше сравниваемого значения в  $S_2$ , включается D. Если значение в  $S_1$  равно значению в  $S_2$ , включается D+1. Если значение в  $S_1$  меньше значения в  $S_2$ , включается D+2.

**Пример:**

- Если операндом D является M10, результат сравнения сохраняется в M10, M11 и M12, как показано ниже.
- Если X0.0 включен, инструкция FCMP выполняется. Включается M10, M11 или M12. При выключении X0.0 инструкция FCMP не выполняется. Состояние M10, M11 и M12 остается неизменным.
- Если необходимо получить результат сравнения  $\geq$ ,  $\leq$  или  $\neq$ , M10~M12 можно подключать последовательно или параллельно.
- Если необходимо очистить результат сравнения, используйте инструкции RST или ZRST.

**Примечание:**

1. Если значение в  $S_1$  или  $S_2$  выходит за допустимый предел, инструкция выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2013.
2. Если операнд  $D$  задается в ПО ISPSOft, тип данных будет: ARRAY [3] для BOOL.
3. If  $D+2$  выходит за допустимый предел, инструкция выполняться не будет, включается специальное вспомогательное реле SM0 и в специальном регистре SR0 сохраняется код ошибки 16#2003.

API	Код инструкции			Операнды						Функция					
0057		FZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$						Зонное сравнение чисел с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○					○
$S_2$	●	●			●	●	●	●	●		○					○
$S$	●	●			●	●	●	●	●		○					○
$D$		●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				
$S$									●				
$D$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

Символ :

FZCP		FZCPP	
En		En	
S1	D	S1	D
S2		S2	
S		S	

- $S_1$  : Минимальное значение зонного сравнения
- $S_2$  : Максимальное значение зонного сравнения
- $S$  : Сравнимое значение
- $D$  : Результат сравнения

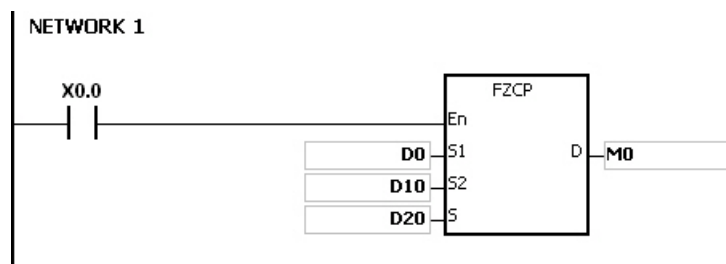
Описание :

- Эта инструкция сравнивает значение в  $S$  со значением в  $S_1$ , а также значение в  $S$  со значением в  $S_2$ . Сравнимые значения являются десятичными, результат сравнения сохраняется в  $D$ .
- Значение в  $S_1$  должно быть меньше значения в  $S_2$ . Если значение в  $S_1$  больше значения в  $S_2$ ,  $S_1$  будет определяться как максимальное/минимальное значение при выполнении инструкции FZCP.
- Операнд  $D$  занимает 3 последовательных регистра. Результат сравнения сохраняется в  $D$ ,  $D+1$  и  $D+2$ . Если сравниваемое значение в  $S_1$  больше сравниваемого значения в  $S_2$ , включается  $D$ . Если значение в  $S_1$  равно значению в  $S_2$ , включается  $D+1$ . Если значение в  $S_1$  меньше значения в  $S_2$ , включается  $D+2$ .

Пример:

- Если операндом  $D$  является  $M0$ , результат сравнения сохраняется в  $M0$ ,  $M1$  и  $M2$ , как показано ниже.
- Когда  $X0.0$  включен, инструкция FZCP выполняется.  $M0$ ,  $M1$  или  $M2$  также включается. Если  $X0.0$  выключается, выполнение инструкции FZCP останавливается. Состояние  $M0$ ,  $M1$  и  $M2$  остается неизменным.
- Если необходимо очистить результат сравнения, используйте инструкции RST или ZRST



**Примечание:**

1. Если значение в **S<sub>1</sub>**, **S<sub>2</sub>** или **S** выходит за допустимый предел, инструкция выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2013.
2. Если операнд **D** задается в ПО ISPSOft, тип данных будет: ARRAY [3] для BOOL.
3. If **D+2** выходит за допустимый предел, инструкция выполняться не будет, включается специальное вспомогательное реле SM0 и в специальном регистре SR0 сохраняется код ошибки 16#2003.

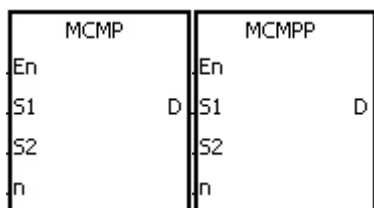
API	Код инструкции			Операнды							Функция						
0058		MCMP	P	$S_1 \cdot S_2 \cdot n \cdot D$							Матричное сравнение						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●		●	●		○					
$S_2$	●	●			●	●		●	●		○					
n	●	●			●	●		●	●		○		○	○		
D		●			●	●		●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ :**



- $S_1$  : Источник матричных данных 1
- $S_2$  : Источник матричных данных 2
- n : Длина массива
- D : Указатель

**Описание :**

- Поиск битов, состояния которых различны, начинается с битов, заданных номером, полученным путем добавления единицы к текущему значению в **D**. После нахождения битов, состояния которых различны, номер бита сохраняется в **D**, а сравнение завершается.
- Значение операнда **n** находится в диапазоне от 1 до 256.
- Когда SM607 включен, сравниваются эквивалентные значения. Когда SM607 выключен, сравниваются различные значения. При нахождении совпадающих битов сравнение немедленно останавливается и включается SM610. При совпадении последних битов в сравнении включается SM608, а номер битов сохраняется в **D**. В следующем цикле сканирования сравнение начинается с 0-го бита и включается SM609. Когда значение в **D** превышает допустимый диапазон, включается флаг SM611.
- Когда выполняется команда MCMP, пользователю необходимо определить 16-разрядный регистр, чтобы указать определенный бит среди битов в матрице. Этот регистр называется указателем. Значение в регистре находится в диапазоне от 0 до 16n-1 и соответствует биту в диапазоне между b0 и b16n-1. Во время выполнения операции нельзя изменять значение указателя, если это влияет на поиск соответствующих битов. Если значение указателя превышает допустимый диапазон, включается флаг SM611, а инструкция MCMP не выполняется.
- Если события, соответствующие флаги SM608 и SM610 происходят одновременно, эти реле включаются также одновременно.



- SM609: Если SM609 включен, сравнение начинается с бита 0.
- SM610: Флаг поиска битов матрицы. Когда сравниваемые биты совпадают, сравнение немедленно прекращается, а SM610 включается.
- SM611: Флаг ошибки матричного указателя. Когда значение указателя превышает допустимое, включается флаг SM611.

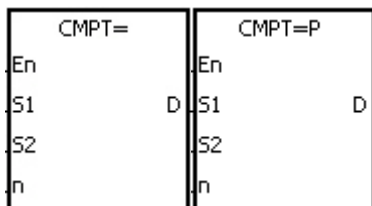
API	Код инструкции		Операнды								Функция						
0059~0064		CMPT※	P	$S_1 \cdot S_2 \cdot n \cdot D$								Сравнение таблиц					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●		○		○	○		
$S_2$	●	●			●	●		●	●		○					
$n$	●	●			●	●		●	●		○	○	○	○		
$D$		●	●	●				●		●						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$n$		●			●	●							
$D$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

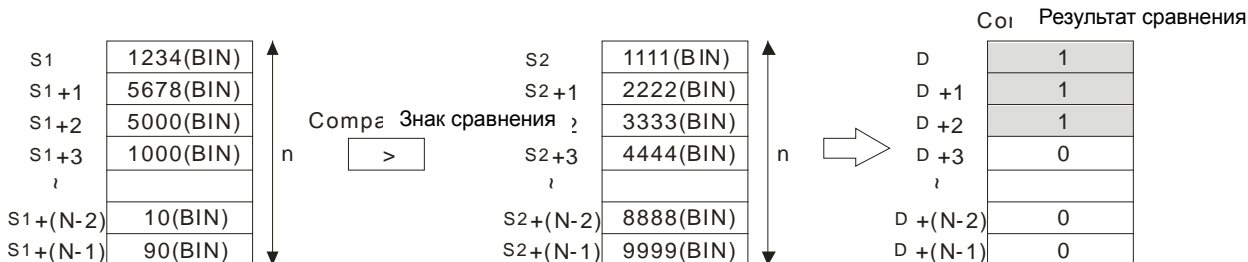
Символ :



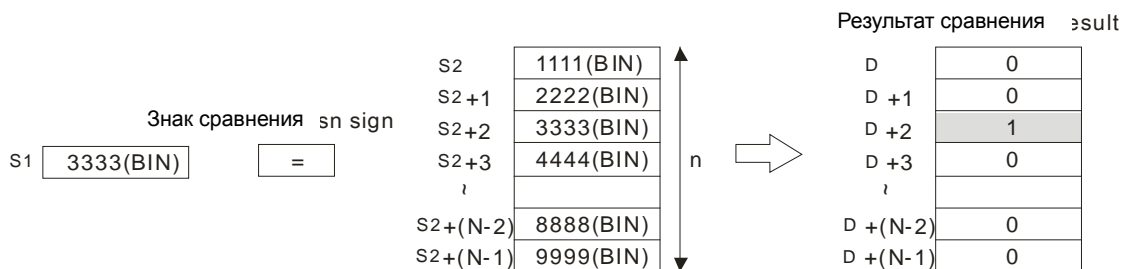
- $S_1$  : Источник 1
- $S_2$  : Источник 2
- $n$  : Длина данных
- $D$  : Результат сравнения

Описание:

- Инструкция сравнивает  $n$  частей данных от операнда  $S_1$  с данными от операнда  $S_2$ . Сравниваемые значения являются десятичными, результат сохраняется в  $D$ .
- Значение операнда  $n$  находится в диапазоне от 1 до 256.
- Значение, которое записывается в операнд  $D$ , является однобитовым.
- Если при выполнении инструкции CMPT# все флаги включены, также включается флаг SM620, в других случаях флаг SM620 выключен.
- Если операндом  $S_1$  является программный объект, операция сравнения выглядит следующим образом:



6. Если операндом  $S_1$  является константа в диапазоне -32768...32767, операция сравнения выглядит следующим образом:

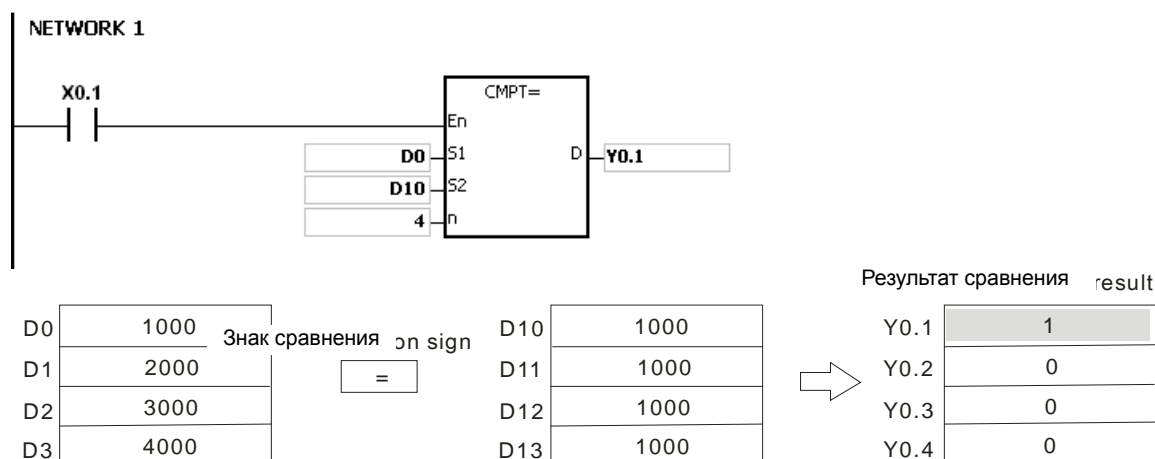


7. Соответствующие результаты операции сравнения приведены ниже.

API	16-битная инструкция	Результат сравнения	
		Непрерывн.	Прерывание
0059	CMPT =	$S_1 = S_2$	$S_1 \neq S_2$
0060	CMPT < >	$S_1 \neq S_2$	$S_1 = S_2$
0061	CMPT >	$S_1 > S_2$	$S_1 \leq S_2$
0062	CMPT > =	$S_1 \geq S_2$	$S_1 < S_2$
0063	CMPT <	$S_1 < S_2$	$S_1 \geq S_2$
0064	CMPT < =	$S_1 \leq S_2$	$S_1 > S_2$

**Пример:**

Данные в D0~D3 сравниваются с данными в D10~D13. Если данные D0~D3 равны данным D10~D13, включаются Y0.1~Y0.4.



**Примечание:**

1. Если значение операнда  $n$  находится вне диапазона от 1 до 256, инструкция выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#200B.
2. Если число объектов, определенных  $S_1 \sim S_1+n$ ,  $S_2 \sim S_2+n$  или  $D$  недостаточно, инструкция выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2003.

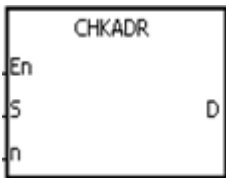
API	Код инструкции	Операнды	Функция
0065	CHKADR	S · n · D	Проверка адреса контактного типа регистра указателя

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S																
n	●	●			●	●		●	●		○	○	○	○		
D		●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S													
n		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ :



S : Регистр указателя

n : Число объектов

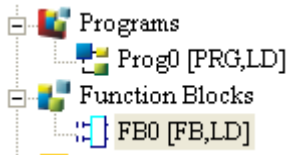
D : Результат проверки

Описание :

1. Инструкция CHKADR используется для проверки того, превышает ли значение в S и (значение в S) + n-1 допустимый диапазон. Если не превышает, D будет включен. В противном случае он будет отключен.
2. S поддерживает регистры указателей D, T, C, HC (POINTER/T\_POINTER/C\_POINTER/HC\_POINTER).
3. Значение операнда n находится в диапазоне от 1 до 1024.
4. Инструкция CHKADR может использоваться только в функциональном блоке. Она используется в начальной фазе разработки программы или для проверки превышения допустимых диапазонов для программных объектов. После написания программы эту инструкцию можно удалить.

Пример:

1. Задайте программу (Prog0) и функциональный блок (FB0) в ПО ISPSOft.



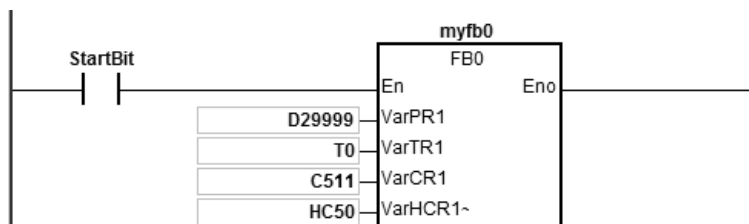
Задайте две переменные в программе.

Local Symbols				
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR	myfb0	N/A [Auto]	FB0	N/A
VAR	StartBit	N/A [Auto]	BOOL	FALSE

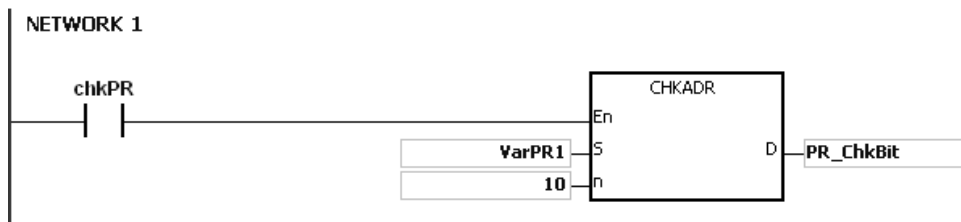
- Задайте VarPR1, VarTR1, VarCR1 и VarHCR1 в функциональном блоке и назначьте для них типы данных POINTER, T\_POINTER, C\_POINTER и HC\_POINTER соответственно.

Local Symbols				
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR_IN_OUT	VarPR1	N/A [Auto]	POINTER	N/A
VAR_IN_OUT	VarTR1	N/A [Auto]	T_POINTER	N/A
VAR_IN_OUT	VarCR1	N/A [Auto]	C_POINTER	N/A
VAR_IN_OUT	VarHCR1	N/A [Auto]	HC_POINTER	N/A
VAR	PR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	TR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	CR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	HCR_ChkBit	N/A [Auto]	BOOL	FALSE
VAR	chkPR	N/A [Auto]	BOOL	N/A
VAR	chkTR	N/A [Auto]	BOOL	N/A
VAR	chkCR	N/A [Auto]	BOOL	N/A
VAR	chkHCR	N/A [Auto]	BOOL	N/A

- Вызовите функциональный блок FB0 в программе и назначьте D29999, T0, C511 и HC50 для VarPR1, VarTR1, VarCR1 и VarHCR1 в функциональном блоке FB0 соответственно.

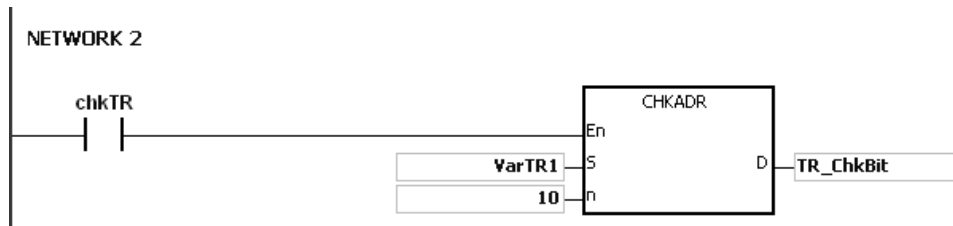


- Используйте инструкцию CHKADR для проверки соответствия диапазонам VarPR1, VarTR1, VarCR1 и VarHCR1.
- Если chkPR включен, объект, представленный переменной VarPR1, будет D29999. Т.к. допустимые диапазоны для этих объектов: D0...D29999 и D29999+10-1=D30008, то диапазон превышен и PR\_ChkBit выключен.

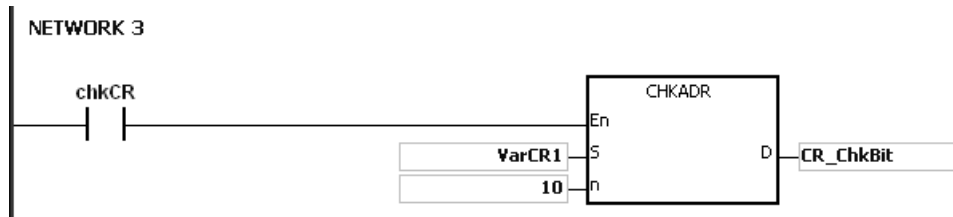


- Если chkTR включен, объект, представленный переменной VarPR1, будет T0. Т.к. допустимые диапазоны для этих объектов: T0...T511 и T0+10-1=T9, то диапазон не превышен и TR\_ChkBit включен.

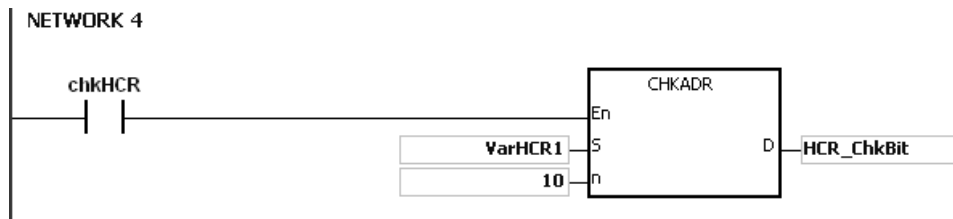




7. Если chkCR включен, объект, представленный переменной VarPR1, будет C511. .к. допустимые диапазоны для этих объектов: C0...C511 и C511+10-1=C520, то диапазон превышен и CR\_ChkBit выключен.



8. Если chkHCR включен, представленный переменной VarPR1, будет HC50. .к. допустимые диапазоны для этих объектов: HC0... HC255 и HC50+10-1=HC59, то диапазон не превышен и HCR\_ChkBit включен.



**Примечание:**

1. Если значение (адрес объекта) в S превышает допустимый диапазон, инструкция CHKADR выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#2003.
2. Если значение операнда n находится вне диапазона от 1 до 1024, инструкция CHKADR выполняться не будет, включается специальное вспомогательное реле SM и в специальном регистре SR0 сохраняется код ошибки 16#200B.

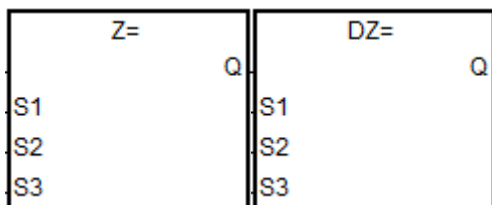
API	Код инструкции			Операнды								Функция				
0066~0071	D	LDZ※		$S_1 \cdot S_2 \cdot S_3$								Сравнение абсолютных значений контактного типа LDZ※				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
$S_3$	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
$S_3$		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ :



- $S_1$  : Источник данных 1
- $S_2$  : Источник данных 2
- $S_3$  : Результат сравнения

Инструкции для примера: LDZ= и DLDZ=

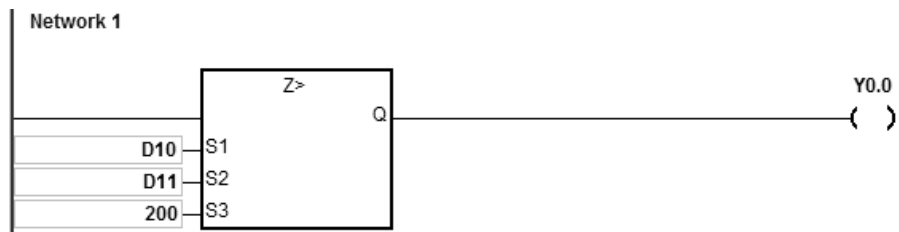
Описание:

- Абсолютное значение разности между  $S_1$  и  $S_2$  сравнивается с абсолютным значением  $S_3$ . Возьмем инструкцию LDZ= для примера. Если результатом сравнения является равенство значения разности  $S_1$  и  $S_2$  и значения  $S_3$ , это является условием непрерывности цепи. Если результатом сравнения является неравенство значения разности  $S_1$  и  $S_2$  и значения  $S_3$ , это является условием прерывания цепи.
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

API	16-битная инструкция	32-битная инструкция	Условие непрерывности	Условие прерывания
0066	LDZ =	DLDZ =	$ S_1 - S_2  =  S_3 $	$ S_1 - S_2  \neq  S_3 $
0067	LDZ < >	DLDZ < >	$ S_1 - S_2  \neq  S_3 $	$ S_1 - S_2  =  S_3 $
0068	LDZ >	DLDZ >	$ S_1 - S_2  >  S_3 $	$ S_1 - S_2  \leq  S_3 $
0069	LDZ > =	DLDZ > =	$ S_1 - S_2  \geq  S_3 $	$ S_1 - S_2  <  S_3 $
0070	LDZ <	DLDZ <	$ S_1 - S_2  <  S_3 $	$ S_1 - S_2  \geq  S_3 $
0071	LDZ < =	DLDZ < =	$ S_1 - S_2  \leq  S_3 $	$ S_1 - S_2  >  S_3 $

**Пример:**

1. Если абсолютное значение разности D10 и D11 больше 200, Y0.0 включается. Если абсолютное значение разности меньше 200, Y0.0 выключается.



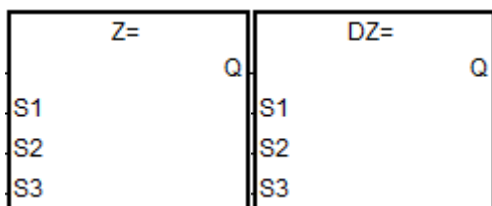
API	Код инструкции			Операнды								Функция					
0072~0077	D	ANDZ※		$S_1 \cdot S_2 \cdot S_3$								Сравнение абсолютных значений контактного типа ANDZ※					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>3</sub>	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●				●	●	
S <sub>2</sub>		●	●		●	●	●				●	●	
S <sub>3</sub>		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ :



- S<sub>1</sub> : Источник данных 1
- S<sub>2</sub> : Источник данных 2
- S<sub>3</sub> : Comparison result

Инструкции для примера: ANDZ= и DANDZ=

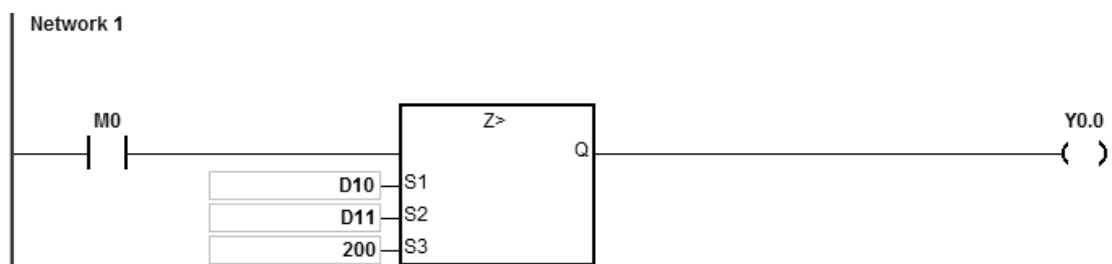
Описание:

- Абсолютное значение разности между S<sub>1</sub> и S<sub>2</sub> сравнивается с абсолютным значением S<sub>3</sub>. Возьмем инструкцию ANDZ= для примера. Если результатом сравнения является равенство значения разности S<sub>1</sub> и S<sub>2</sub> и значения S<sub>3</sub>, это является условием непрерывности цепи. Если результатом сравнения является неравенство значения разности S<sub>1</sub> и S<sub>2</sub> и значения S<sub>3</sub>, это является условием прерывания цепи.
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

API	16-битная инструкция	32-битная инструкция	Условие непрерывности	Условие прерывания
0072	ANDZ =	DANDZ =	$ S_1 - S_2  =  S_3 $	$ S_1 - S_2  \neq  S_3 $
0073	ANDZ < >	DANDZ < >	$ S_1 - S_2  \neq  S_3 $	$ S_1 - S_2  =  S_3 $
0074	ANDZ >	DANDZ >	$ S_1 - S_2  >  S_3 $	$ S_1 - S_2  \leq  S_3 $
0075	ANDZ > =	DANDZ > =	$ S_1 - S_2  \geq  S_3 $	$ S_1 - S_2  <  S_3 $
0076	ANDZ <	DANDZ <	$ S_1 - S_2  <  S_3 $	$ S_1 - S_2  \geq  S_3 $
0077	ANDZ < =	DANDZ < =	$ S_1 - S_2  \leq  S_3 $	$ S_1 - S_2  >  S_3 $

**Пример:**

1. Когда M0 включен и абсолютная разность между D10 и D11 больше 200, Y0.0 включен. Когда абсолютная разность меньше 200, Y0.0 выключен.



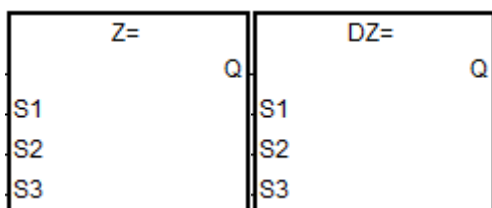
API	Код инструкции			Операнды								Функция				
0078~0083	D	ORZ※		$S_1 \cdot S_2 \cdot S_3$								Сравнение абсолютных значений контактного типа ORZ※				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>3</sub>	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●				●	●	
S <sub>2</sub>		●	●		●	●	●				●	●	
S <sub>3</sub>		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ :



- S<sub>1</sub> : Источник данных 1
- S<sub>2</sub> : Источник данных 2
- S<sub>3</sub> : Результат сравнения

Инструкции для примера: ORZ= и DORZ=

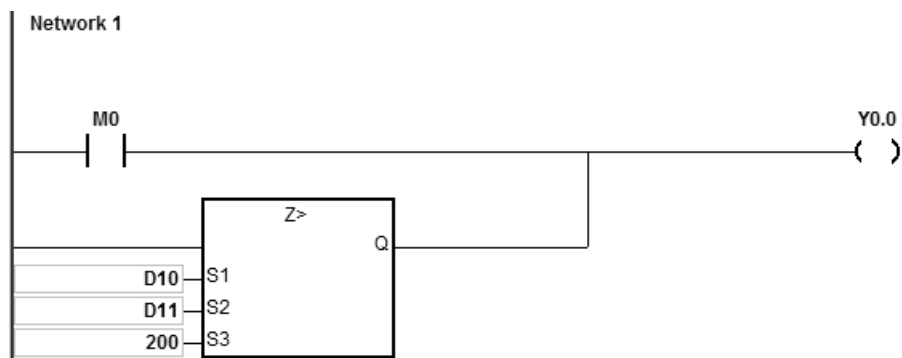
Описание:

- Абсолютное значение разности между S<sub>1</sub> и S<sub>2</sub> сравнивается с абсолютным значением S<sub>3</sub>. Возьмем инструкцию ORZ= для примера. Если результатом сравнения является равенство значения разности S<sub>1</sub> и S<sub>2</sub> и значения S<sub>3</sub>, это является условием непрерывности цепи. Если результатом сравнения является неравенство значения разности S<sub>1</sub> и S<sub>2</sub> и значения S<sub>3</sub>, это является условием прерывания цепи.
- 32-битные счетчики используются только 32-битными инструкциями, объект E такие счетчики не использует.

API	16-битная инструкция	32-битная инструкция	Условие непрерывности	Условие прерывания
0078	ORZ =	DORZ =	$ S_1 - S_2  =  S_3 $	$ S_1 - S_2  \neq  S_3 $
0079	ORZ < >	DORZ < >	$ S_1 - S_2  \neq  S_3 $	$ S_1 - S_2  =  S_3 $
0080	ORZ >	DORZ >	$ S_1 - S_2  >  S_3 $	$ S_1 - S_2  \leq  S_3 $
0081	ORZ > =	DORZ > =	$ S_1 - S_2  \geq  S_3 $	$ S_1 - S_2  <  S_3 $
0082	ORZ <	DORZ <	$ S_1 - S_2  <  S_3 $	$ S_1 - S_2  \geq  S_3 $
0083	ORZ < =	DORZ < =	$ S_1 - S_2  \leq  S_3 $	$ S_1 - S_2  >  S_3 $

**Пример:**

1. Когда M0 включен и абсолютная разность между D10 и D11 больше 200, Y0.0 включен. Когда абсолютная разность меньше 200, Y0.0 выключен.



## 6.2 Арифметические инструкции

### 6.2.1 Описание арифметических инструкций

API	Код инструкции			Операнды								Функция				
0100	D	+	P	$S_1 \cdot S_2 \cdot D$								Сложение двоичных чисел				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символ:

+		+P	
En		En	
S1	D	S1	D
S2		S2	

$S_1$  : Слагаемое

$S_2$  : Слагаемое

D : Сумма

D+		D+P	
En		En	
S1	D	S1	D
S2		S2	

Описание:

1. Двоичное значение в  $S_2$  складывается с двоичным значением в  $S_1$ , сумма сохраняется в D.
2. 32-битный счетчик может использоваться только 32-битной инструкцией D+, объекты E такие счетчики не поддерживают.
3. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования) и SM602 (флаг переноса)
4. Если результат сложения равен 0, включается флаг нуля SM600.
5. Сложение 16-битных двоичных значений:



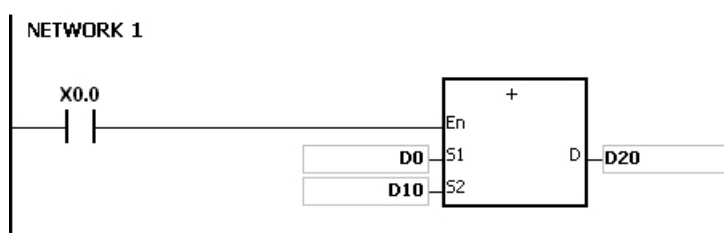
Если результат сложения выходит за допустимый для 16-битных значений предел, включается флаг переноса SM602.

6. Сложение 32-битных двоичных значений:

Если результат сложения выходит за допустимый для 32-битных значений предел, включается флаг переноса SM602.

**Пример 1:**

Сложение 16-битных двоичных значений: Когда X0.0 включен, значение в D10 складывается со значением в D0, сумма сохраняется в D20.

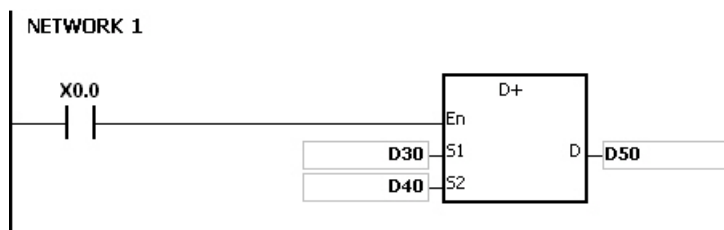


- Когда значения в D0 и D10 равны 100 и 10 соответственно, сумма равна 110, результат сохраняется в D20.
- Когда значения в D0 и D10 равны 16#7FFF и 16#1 соответственно, сумма равна 16#8000, результат сохраняется в D20.
- Когда значения в D0 и D10 равны 16#FFFF и 16#1 соответственно, сумма равна 16#10000. Т.к. результат выходит за допустимый диапазон 16-битных значений, включается флаг переноса SM602 и сохраненный в D20 результат равен 16#0. Т.к. результат равен 16#0, также включается флаг нуля SM600.

6

**Пример 2:**

Сложение 32-битных двоичных значений: Когда X0.0 включен, значение в (D41, D40) складывается со значением в (D31, D30), сумма сохраняется в (D51, D50). (Данные в D30, D40 и D50 это младшие 16 бит данных, в D31, D41 и D51 это старшие 16 бит данных).



- Когда значения в (D31, D30) и (D41, D40) равны 11111111 и 44444444 соответственно, сумма равна 55555555, результат сохраняется в (D51, D50).

- Когда значения в (D31, D30) и (D41, D40) равны 16#80000000 и 16#FFFFFFF соответственно, (D31, D30) сумма равна 16#17FFFFFFF. Т.к. результат выходит за допустимый диапазон 32-битных значений, включается флаг переноса SM602, сохраненный в (D51, D50) результат равен 16#7FFFFFFF.

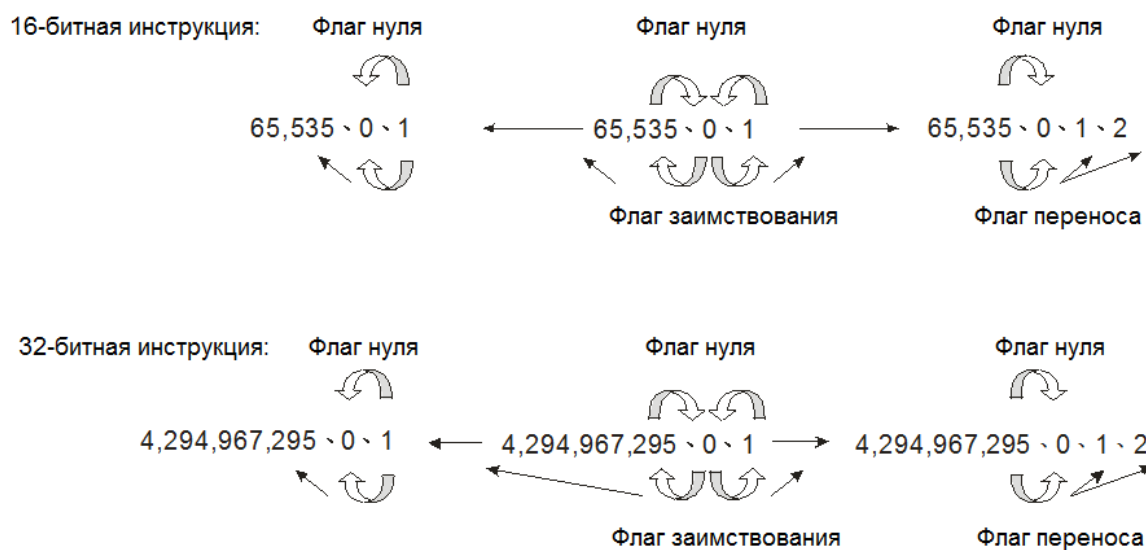
**Флаги:**

16-битная инструкция:

1. Если результат равен 0, включается флаг нуля SM600.
2. Если результат превышает 65 535, включается флаг переноса SM602.

32-битная инструкция:

1. Если результат равен 0, включается флаг нуля SM600.
2. Если результат превышает 4,294,967,295, включается флаг переноса SM602.



API	Код инструкции			Операнды						Функция					
0101	D	-	P	$S_1 \cdot S_2 \cdot D$						Вычитание двоичных чисел					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**

-		-P	
En		En	
S1	D	S1	D
S2		S2	
D-		D-P	
En		En	
S1	D	S1	D
S2		S2	

$S_1$  : Уменьшаемое

$S_2$  : Вычитаемое

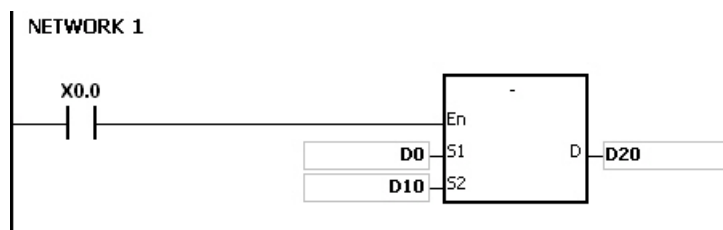
D : Разность

**Описание:**

1. Двоичное значение в  $S_2$  вычитается из двоичного значения  $S_1$ , разность сохраняется в D.
2. 32-битный счетчик может использоваться только 32-битной инструкцией D-, объекты E такие счетчики не поддерживают.
3. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования) и SM602 (флаг переноса)
4. Если результат вычитания равен 0, включается флаг нуля SM600.
5. Когда во время арифметической операции происходит заимствование, включается флаг SM601.

**Пример 1:**

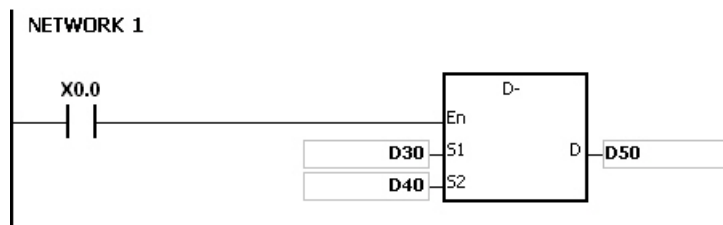
Вычитание 16-битных чисел: Когда X0.0 включен, вычитаемое в D10 вычитается из уменьшаемого в D0, разность сохраняется в D20.



- Когда значения в D0 и D10 равны 100 и 10 соответственно, разность равна 90 и сохраняется в D20.
- Когда значения в D0 и D10 равны 16#8000 и 16#1 соответственно, разность равна 16#7FFF и сохраняется в D20.
- Когда значения в D0 и D10 равны 16#1 и 16#2 соответственно, разность равна 16#FFFF. Поскольку происходит заимствование, включается флаг SM601 и в D20 сохраняется значение 16#FFFF.
- Когда значения в D0 и D10 равны 16#0 и 16#FFFF соответственно, разность равна 16#F0001. Поскольку происходит заимствование, включается флаг SM601 и в D20 сохраняется значение 16#1.

#### Пример 2:

Вычитание 32-битных чисел: Когда X0.0 включен, вычитаемое в (D41, D40) вычитается из уменьшаемого (D31, D30), разность сохраняется в (D51, D50). (Данные в D30, D40 и D50 – младшие 16 бит данных, данные в D31, D41 и D51 – старшие 16 бит данных).



- Когда значения в (D31, D30) и (D41, D40) соответственно равны 55555555 и 11111111, разность составляет 44444444 и сохраняется в (D51, D50).
- Когда значения в (D31, D30) и (D41, D40) соответственно равны 16#80000000 и 16#FFFFFFF разность составляет 16#F80000001. Поскольку происходит заимствование, включается флаг SM601, в (D51, D50) сохраняется значение 16#80000001.

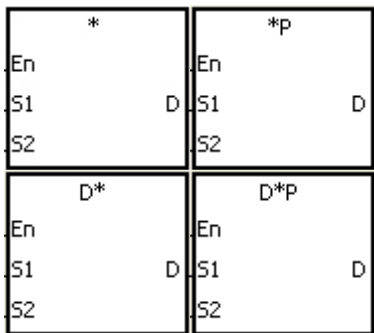
API	Код инструкции			Операнды						Функция					
0102	D	*	P	$S_1 \cdot S_2 \cdot D$						Умножение двоичных чисел					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

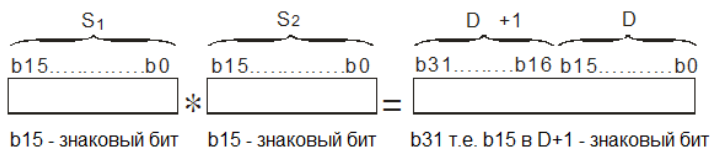
**Символ:**



- $S_1$  : Множимое
- $S_2$  : Множитель
- D : Произведение

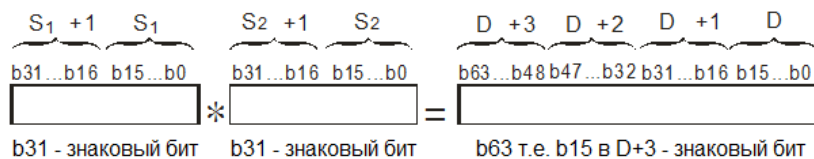
**Описание:**

1. Двоичное значение со знаком в  $S_1$  перемножается с двоичным значением со знаком в  $S_2$ , произведение сохраняется в D.
2. 32-битный счетчик может использоваться только 32-битной инструкцией D\*.
3. Умножение 16-битных двоичных значений:



Произведение представляет собой 32-битное значение и хранится в регистре (D+1, D), который состоит из 32 бит данных. Когда знаковый бит b31 равен 0, произведение является положительным значением. Когда знаковый бит b31 равен 1, произведение является отрицательным значением.

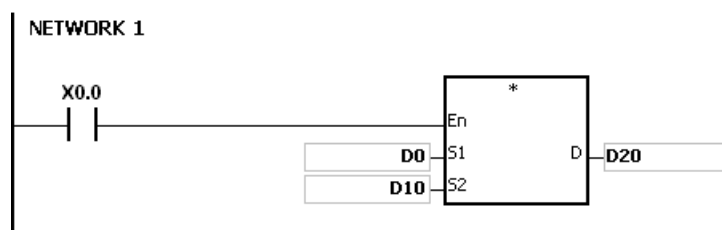
4. Умножение 32-битных двоичных значений:



Произведение является 64-битным значением и хранится в регистре (D+3, D+2, D+1, D0), который состоит из 64 бит. Когда знаковый бит b63 равен 0, произведение является положительным значением. Когда знаковый бит b63 равен 1, произведение является отрицательным значением.

**Пример:**

16-битное значение в D0 умножается на 16-битное значение в D10, а 32-битное произведение сохраняется в (D21, D20). Данные в D21 - это старшие 16 бит данных, данные в D20 - это младшие 16 бит данных. Имеет ли результат положительное или отрицательное значение, зависит от состояния самого старшего бита b31. Когда b31 выключен (равен 0), результат будет положительным. Когда b31 включен (равен 1), результатом является отрицательное значение.



$D0 \times D10 = (D21, D20)$

16-битное значение  $\times$  16-битное значение = 32-битное значение

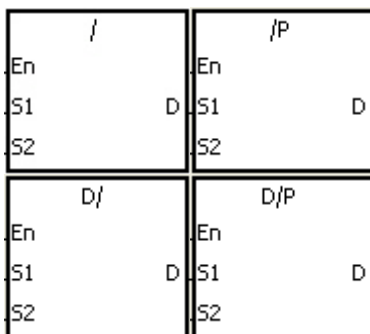
API	Код инструкции			Операнды								Функция				
0103	D	/	P	$S_1 \cdot S_2 \cdot D$								Деление двоичных чисел				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

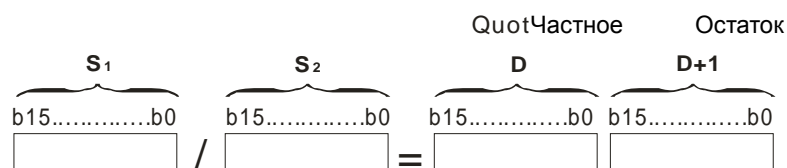
**Символ:**



- $S_1$  : Делимое
- $S_2$  : Делитель
- D : Частное, остаток

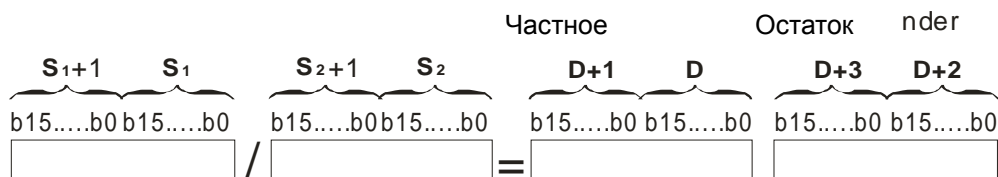
**Описание:**

1. Двоичное значение со знаком в  $S_1$  делится нас двоичное значение со знаком в  $S_2$ , частное сохраняется в D.
2. 32-битный счетчик может использоваться только 32-битной инструкцией D/.
3. Если знаковый бит равен 0, значение положительное. Если знаковый бит равен 1, значение отрицательное.
4. Деление 16-битных значений:



Операнд D занимает 2 последовательных регистра. Частное сохраняется в D, остаток сохраняется в D+1.

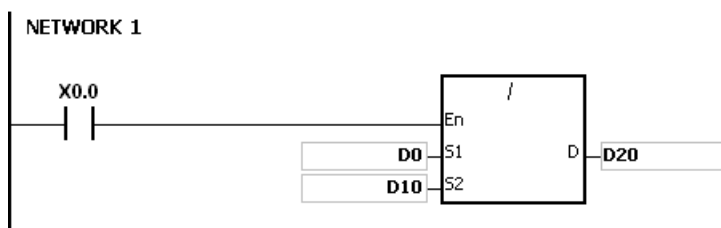
5. Деление 32-битных значений:



Операнд **D** занимает 2 последовательных регистра. Частное сохраняется в (**D+1, D**), остаток сохраняется в (**D+3, D+2**).

**Пример:**

Когда X0.0 включен, значение делимого в D0 делится на значение делителя в D10, частное сохраняется в D20, а остаток сохраняется в D21. Знак результата определяется значением его старшего бита.



**Примечание:**

1. Если значение операнда выходит за допустимые пределы, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение делителя равно 0, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2012.
3. Если операнд **D**, используемый с 16-битной инструкцией, задается в ПО ISPSOft, тип данных будет ARRAY [2] для WORD/INT.
4. Если операнд **D**, используемый с 32-битной инструкцией, задается в ПО ISPSOft, тип данных будет ARRAY [2] для DWORD/DINT.



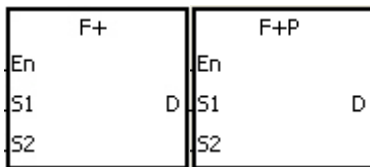
API	Код инструкции			Операнды							Функция					
0104		F+	P	$S_1 \cdot S_2 \cdot D$							Сложение чисел с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							○
$S_2$	●	●			●	●	●	●	●							○
<b>D</b>		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символ:**



$S_1$  : Слагаемое

$S_2$  : Слагаемое

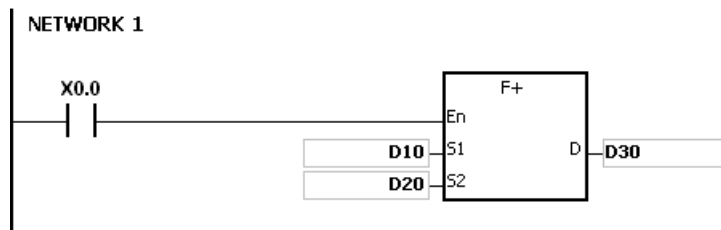
**D** : Сумма

**Описание:**

- Эта инструкция предназначена для сложения 32-битных чисел одинарной точности с плавающей запятой.
- Число с плавающей запятой в  $S_2$  складывается с числом с плавающей запятой в  $S_1$ , сумма сохраняется в **D**.
- Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), SM602 (флаг переноса)
  - Если сумма равна 0, включается флаг нуля SM600.
  - Когда абсолютное значение результата меньше минимального значения для чисел с плавающей запятой, значение в **D** равно 16#FF800000 и включается флаг заимствования SM601.
  - Когда абсолютное значение результата больше максимального значения для чисел с плавающей запятой, значение в **D** равно 16#7F800000 включается флаг переноса SM602.

**Пример:**

Сложение чисел одинарной точности с плавающей запятой: Когда X0.0 включен, значение 16#4046B852 в (D21, D20) складывается со значением 16#3FB9999A в (D11, D10), сумма, равная 16#4091C28F, сохраняется в (D31, D30). 16#4046B852, 16#3FB9999A и 16#4091C28F представляют собой числа 3,105, 1,450 и 4,55 соответственно.

**Примечание:**

Если значения в  $S_1$  или  $S_2$  выходят за допустимые для чисел с плавающей запятой пределы, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2013.

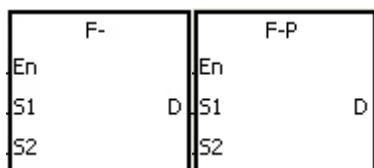
API	Код инструкции			Операнды							Функция					
0105		F-	P	$S_1 \cdot S_2 \cdot D$							Вычитание чисел с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							○
$S_2$	●	●			●	●	●	●	●							○
D		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

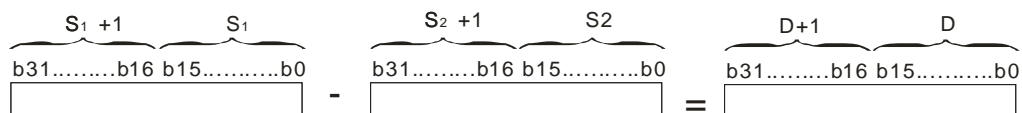
**Символ:**



- $S_1$  : Уменьшаемое
- $S_2$  : Вычитаемое
- D : Разность

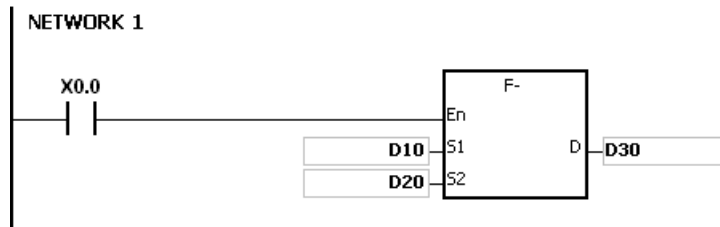
**Описание:**

4. Эта инструкция предназначена для вычитания 32-битных чисел одинарной точности с плавающей запятой.
5. Из числа с плавающей запятой в  $S_2$  вычитается число с плавающей запятой в  $S_1$ , разность сохраняется в D.
6. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), SM602 (флаг переноса)
  - Если разность равна 0, включается флаг нуля SM600.
  - Когда абсолютное значение результата меньше минимального значения для чисел с плавающей запятой, значение в D равно 16#FF800000 и включается флаг заимствования SM601.
  - Когда абсолютное значение результата больше максимального значения для чисел с плавающей запятой, значение в D равно 16#7F800000 включается флаг переноса SM602.



**Пример:**

Вычитание чисел одинарной точности с плавающей запятой: когда X0.0 включен, значение в (D21, D20) вычитается из уменьшаемого (D21, D20), разность сохраняется в (D31, D30).

**Примечание:**

Если значения в **S<sub>1</sub>** или **S<sub>2</sub>** выходят за допустимые для чисел с плавающей запятой пределы, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2013.

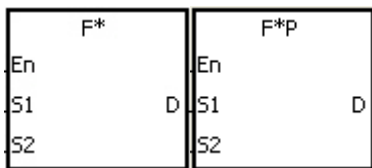
API	Код инструкции			Операнды							Функция					
0106		F*	P	$S_1 \cdot S_2 \cdot D$							Умножение чисел с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							○
$S_2$	●	●			●	●	●	●	●							○
<b>D</b>		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

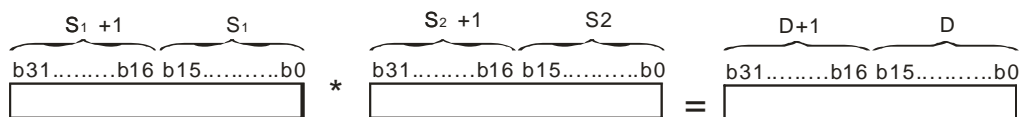
**Символ:**



- $S_1$  : Множимое
- $S_2$  : Множитель
- D** : Произведение

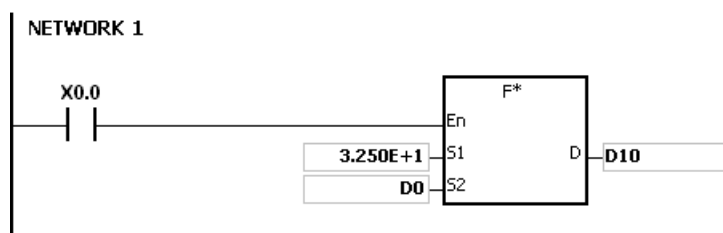
**Описание:**

7. Эта инструкция предназначена для перемножения 32-битных чисел одинарной точности с плавающей запятой.
8. Число с плавающей запятой в  $S_2$  перемножается с числом с плавающей запятой в  $S_1$ , произведение сохраняется в **D**.
9. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), SM602 (флаг переноса)
  - Если произведение равно 0, включается флаг нуля SM600.
  - Когда абсолютное значение результата меньше минимального значения для чисел с плавающей запятой, значение в **D** равно 16#FF800000 и включается флаг заимствования SM601.
  - Когда абсолютное значение результата больше максимального значения для чисел с плавающей запятой, значение в **D** равно 16#7F800000 включается флаг переноса SM602.



**Пример:**

Умножение чисел одинарной точности с плавающей запятой: когда X0.0 включен, множимое 32.5 перемножается со значением множителя в (D1, D0), произведение сохраняется в (D11, D10).



**Примечание:**

Если значения в S<sub>1</sub> или S<sub>2</sub> выходят за допустимые для чисел с плавающей запятой пределы, инструкция не выполняется, включается флаг SMO и в SR0 записывается код ошибки 16#2013.

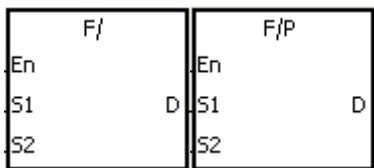
API	Код инструкции			Операнды							Функция					
0107		F/	P	$S_1 \cdot S_2 \cdot D$							Деление чисел с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							○
$S_2$	●	●			●	●	●	●	●							○
D		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$									●				
$S_2$									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

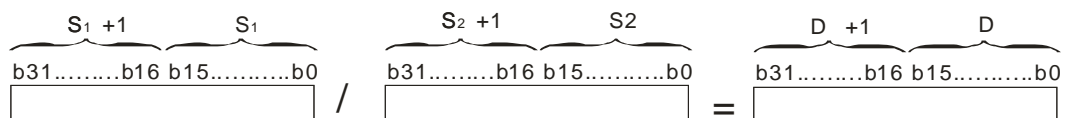
**Символ:**



- $S_1$  : Делимое
- $S_2$  : Делитель
- D : Частное

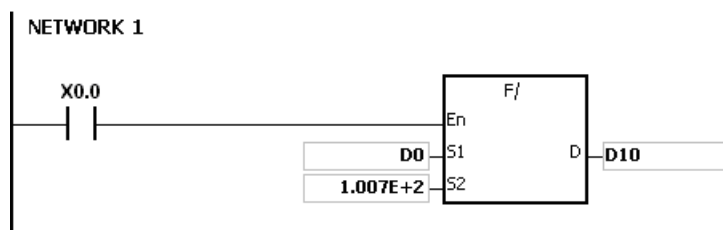
**Описание:**

10. Эта инструкция предназначена для деления 32-битных чисел одинарной точности с плавающей запятой.
11. Число одинарной точности с плавающей запятой в  $S_1$  делится на число одинарной точности с плавающей запятой в  $S_2$ , частное сохраняется в D.
12. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), SM602 (флаг переноса)
  - Если частное равно 0, включается флаг нуля SM600.
  - Когда абсолютное значение результата меньше минимального значения для чисел с плавающей запятой, значение в D равно 16#FF800000 и включается флаг заимствования SM601.
  - Когда абсолютное значение результата больше максимального значения для чисел с плавающей запятой, значение в D равно 16#7F800000 включается флаг переноса SM602.



**Пример:**

Деление чисел одинарной точности с плавающей запятой: когда X0.0 включен, делимое в (D1, D0) делится на значение делителя 100.7, частное сохраняется в (D11, D10).



**Примечание:**

1. Если значение делителя равно 0, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2012.
2. Если значения в S<sub>1</sub> или S<sub>2</sub> выходят за допустимые для чисел с плавающей запятой пределы, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2013.



API	Код инструкции			Операнды						Функция					
0112	D	BK+	P	$S_1 \cdot S_2 \cdot n \cdot D$						Сложение двоичных чисел в блоках					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							
$S_2$	●	●			●	●	●	●	●				○	○		
n	●	●			●	●	●	●	●				○	○		
D		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●					●	●	
$S_2$		●			●	●					●	●	
n		●			●	●					●	●	
D		●			●	●					●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**

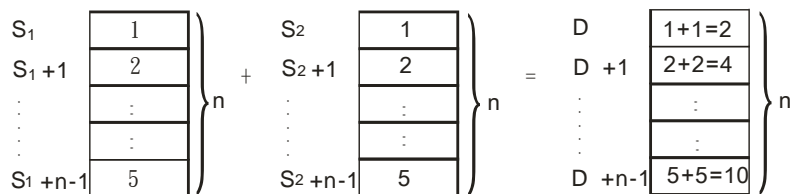
BK+		BK+P	
En		En	
S1	D	S1	D
S2		S2	
n		n	
DBK+		DBK+P	
En		En	
S1	D	S1	D
S2		S2	
n		n	

- $S_1$  : Слагаемое
- $S_2$  : Слагаемое
- n : Длина данных
- D : Сумма

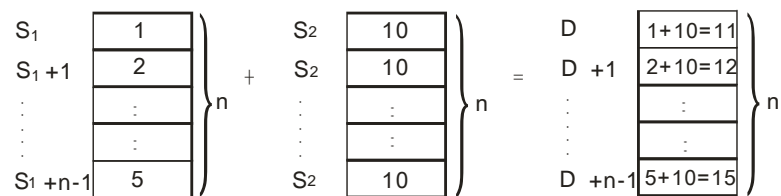
**Описание:**

1. n частей данных из  $S_2$  добавляются к данным в операндах, начиная с  $S_1$ . Слагаемые являются двоичными значениями, результат сохраняется в D.
2. Значение операнда n находится в диапазоне от 1 до 256.
3. 32-битный счетчик может использовать только 32-битная инструкция DBK+.
4. Если результат сложения равен 0, включается флаг нуля SM600.
5. Для 16-битной инструкции, если результат меньше -32 768, включается флаг заимствования SM601.
6. Для 16-битной инструкции, если результат больше 32 767, включается флаг переноса SM602.
7. Для 32-битной инструкции, если результат меньше -21 474 836 488, включается флаг заимствования SM601

8. Для 32-битной инструкции, если результат больше 2 147 483 647, включается флаг переноса SM602.
9. Пример 16-битной инструкции, когда операнд  $S_2$  является регистром (не константа и не шестнадцатеричное значение):

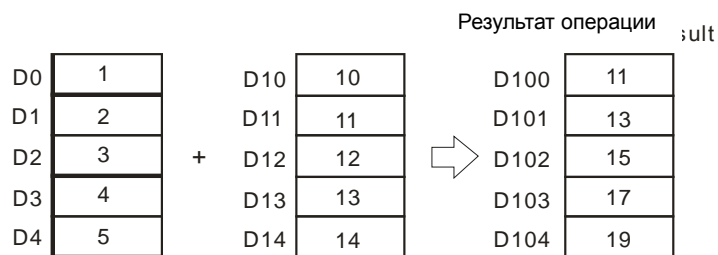
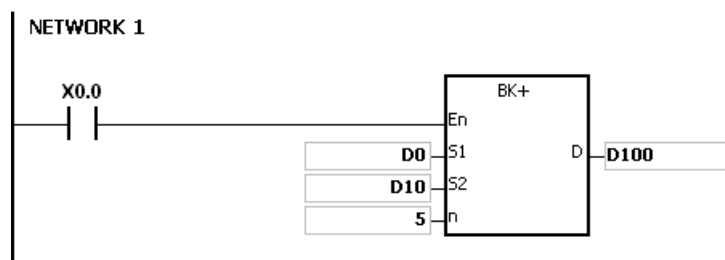


10. Пример 16-битной инструкции, когда операнд  $S_2$  является константой или шестнадцатеричным значением:



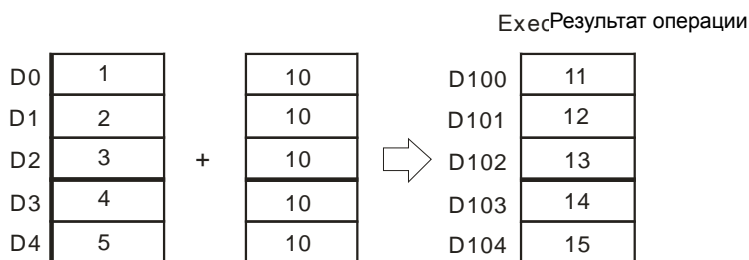
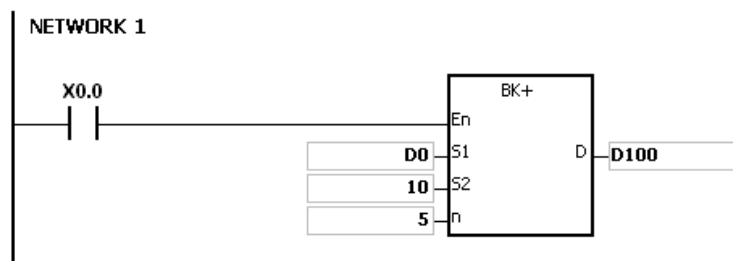
**Пример 1:**

Когда X0.0 включен, двоичные значения в D10~D14 складываются с двоичными значениями в D0~D4, сумма сохраняется в D100~D104.



**Пример 2:**

Когда X0.0 включен, слагаемое 10 складывается с двоичным значением D0~D4, сумма сохраняется в D100~D104.



**Примечание:**

1. Для 16-битной инструкции, если значения в  $S_1 \sim S_1+n-1$ ,  $S_2 \sim S_2+n-1$  или  $D \sim D+n-1$  выходят за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM и в SR0 записывается код ошибки 16#2003.
2. Для 32-битной инструкции, если значения в  $S_1 \sim S_1+2*n-1$ ,  $S_2 \sim S_2+2*n-1$  или  $D \sim D+2*n-1$  выходят за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM и в SR0 записывается код ошибки 16#2003.
3. Если  $n < 1$  или  $n > 256$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.
4. Для 16-битной инструкции, если  $S_1 \sim S_1+n-1$  перекрывает  $D \sim D+n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.
5. Для 32-битной инструкции, если  $S_1 \sim S_1+2*n-1$  перекрывает  $D \sim D+2*n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.
6. Для 16-битной инструкции, если  $S_2 \sim S_2+n-1$  перекрывает  $D \sim D+n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.
7. Для 32-битной инструкции, если  $S_2 \sim S_2+2*n-1$  перекрывает  $D \sim D+2*n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.

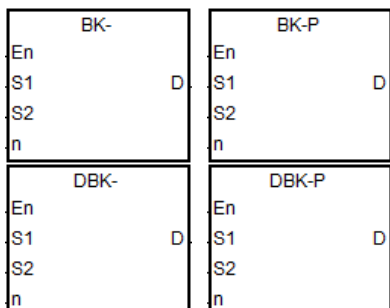
API	Код инструкции			Операнды							Функция					
0113	D	BK-	P	$S_1 \cdot S_2 \cdot n \cdot D$							Вычитание двоичных чисел в блоках					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							
$S_2$	●	●			●	●	●	●	●				○	○		
$n$	●	●			●	●	●	●	●				○	○		
$D$		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●					●	●	
$S_2$		●			●	●					●	●	
$n$		●			●	●					●	●	
$D$		●			●	●					●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**

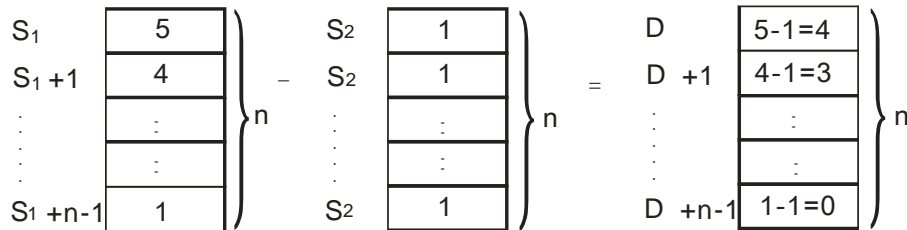


- $S_1$  : Уменьшаемое
- $S_2$  : Вычитаемое
- $n$  : Длина данных
- $D$  : Разность

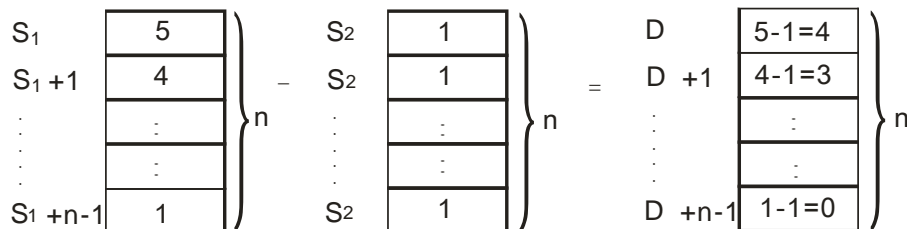
**Описание:**

- $n$  частей данных из  $S_2$  вычитаются из данных в операндах, начиная с  $S_1$ . Уменьшаемое и вычитаемое являются двоичными значениями, результат сохраняется в  $D$ .
- Значение операнда  $n$  находится в диапазоне от 1 до 256.
- 32-битный счетчик может использовать только 32-битная инструкция DBK-.
- Если результат вычитания равен 0, включается флаг нуля SM600.
- Для 16-битной инструкции, если результат меньше -32 768, включается флаг заимствования SM601.
- Для 16-битной инструкции, если результат больше 32 767, включается флаг переноса SM602.
- Для 32-битной инструкции, если результат меньше -21 474 836 488, включается флаг заимствования SM601.

- 8. Для 32-битной инструкции, если результат больше 2 147 483 647, включается флаг переноса SM602.
- 9. Пример 16-битной инструкции, когда операнд S<sub>2</sub> является регистром (не константа и не шестнадцатеричное значение):

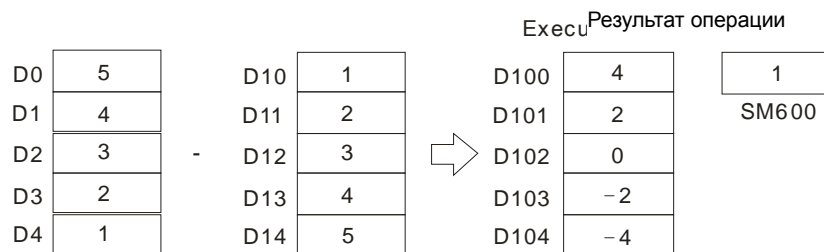
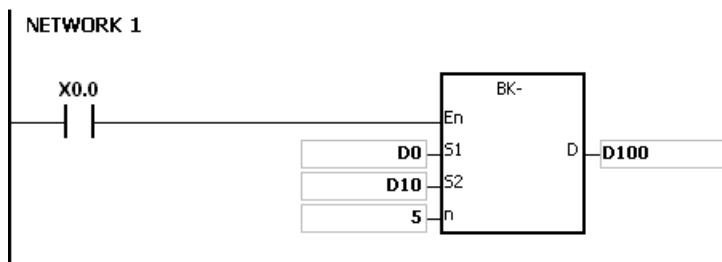


- 10. Пример 16-битной инструкции, когда операнд S<sub>2</sub> является константой или шестнадцатеричным значением:



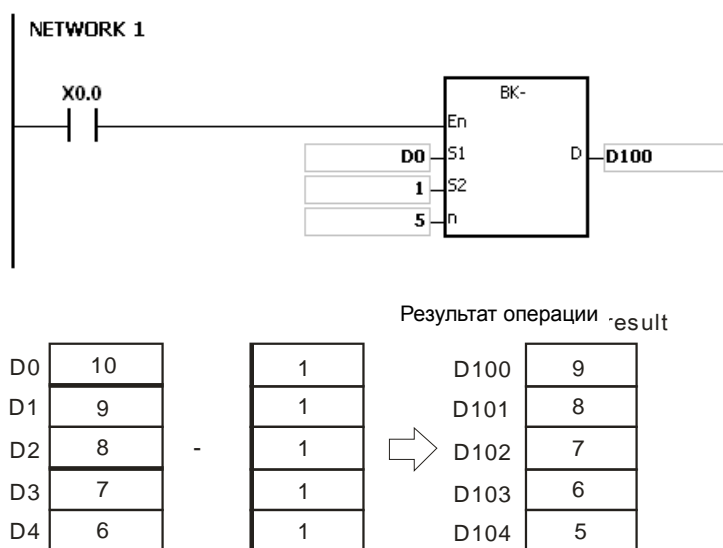
**Пример 1:**

Когда X0.0 включен, двоичные значения в D10~D14 вычитается из двоичных значений D0~D4, разность сохраняется в D100~D104.



**Пример 2:**

Когда X0.0 включен, вычитаемое 1 вычитается из двоичного значения D0~D4, разность сохраняется в D100~D104.

**Примечание:**

1. Для 16-битной инструкции, если значения в  $S_1 \sim S_1+n-1$ ,  $S_2 \sim S_2+n-1$  или  $D \sim D+n-1$  выходят за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM и в SR0 записывается код ошибки 16#2003.
2. Для 32-битной инструкции, если значения в  $S_1 \sim S_1+2*n-1$ ,  $S_2 \sim S_2+2*n-1$  или  $D \sim D+2*n-1$  выходят за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM и в SR0 записывается код ошибки 16#2003.
3. Если  $n < 1$  или  $n > 256$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.
4. Для 16-битной инструкции, если  $S_1 \sim S_1+n-1$  перекрывает  $D \sim D+n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.
5. Для 32-битной инструкции, если  $S_1 \sim S_1+2*n-1$  перекрывает  $D \sim D+2*n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.
6. Для 16-битной инструкции, если  $S_2 \sim S_2+n-1$  перекрывает  $D \sim D+n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.
7. Для 32-битной инструкции, если  $S_2 \sim S_2+2*n-1$  перекрывает  $D \sim D+2*n-1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200C.

API	Код инструкции			Операнды							Функция					
0114		\$+	P	<b>S<sub>1</sub> · S<sub>2</sub> · D</b>							Связывание строковых переменных					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●		●	●						○	
S <sub>2</sub>	●	●			●	●		●	●						○	
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>													●
S <sub>2</sub>													●
D													●

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

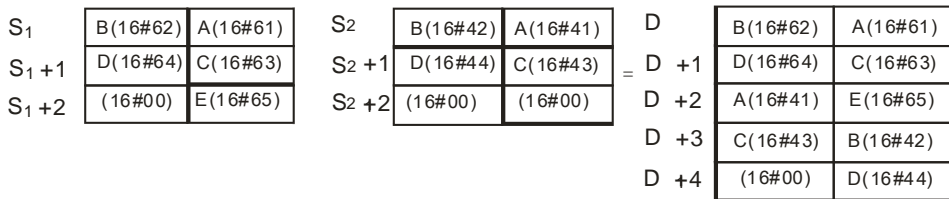
**Символ:**

\$+		\$+P	
En		En	
S1	D	S1	D
S2		S2	

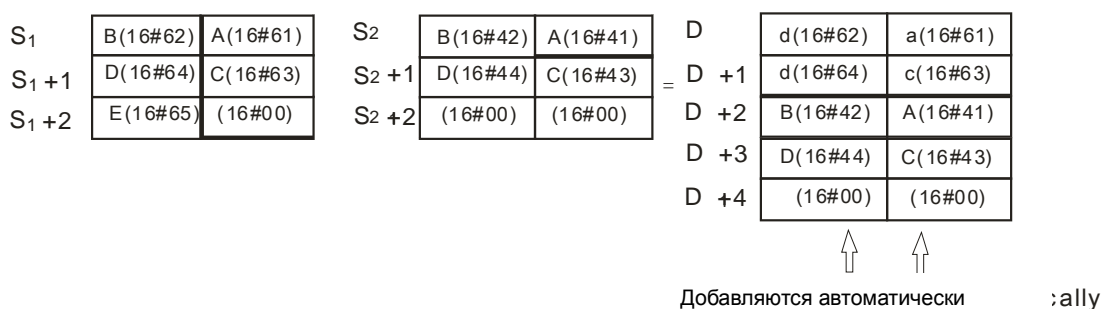
- S<sub>1</sub> : Строковая переменная 1
- S<sub>2</sub> : Строковая переменная 2
- D : Операнд сохранения результата связи

**Описание:**

- При выполнении инструкции, строковые переменные, начиная с данных в S<sub>1</sub> (исключая 16#00), и строковые переменные, начиная с данных в S<sub>2</sub> (исключая 16#00), связываются и перемещаются в D. Кроме того, код 16#00 добавляется в конец связанных строковых переменных в операнде D. Когда инструкция не выполняется, данные в D не изменяются.
- Если данные в источниках S<sub>1</sub>, S<sub>2</sub> или в операнде D не являются строковыми переменными (\$), содержание источника данных составляет до 256 символов (включая код окончания 16#00).
- Процесс связывания строковых переменных в S<sub>1</sub> и в S<sub>2</sub>, а также их перемещение в D, показаны ниже.



↑ Turnin добавляется автоматически

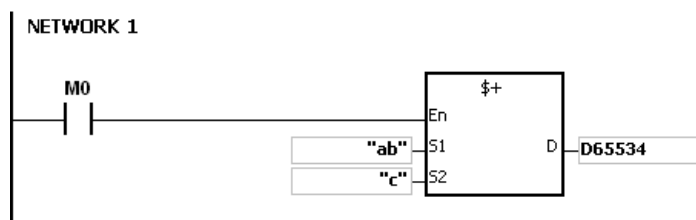


4. Если данные в источниках **S<sub>1</sub>**, **S<sub>2</sub>** или в операнде **D** не являются строковыми переменными (\$), код окончания 16#00 следует добавить в конец перемещенных данных.
5. Предположим, данные в источниках **S<sub>1</sub>** или **S<sub>2</sub>** не являются строковыми переменными (\$). Если инструкция выполняется и первым символом является код 16#00, 16#00 связывает и перемещает данные.
6. Строковая величина “abcde” в операнде **S<sub>1</sub>** показана ниже.

S <sub>1</sub>	b(16#62)	a(16#61)
S <sub>1</sub> +1	d(16#64)	c(16#63)
S <sub>1</sub> +2	(16#00)	e(16#65)

**Пример:**

Предположим, в **S<sub>1</sub>** содержится строковая величина “ab” , а в **S<sub>2</sub>** строковая величина “c”. После включения назначенного контакта M0 данные в D65534 будут 16#6261 и данные в D65535 будут 16#0063.



**Примечание:**

1. Если **S<sub>1</sub>** или **S<sub>2</sub>** содержат строковые переменные, возможно перемещение не более 31 символа.
2. Если объема **D** не хватает для сохранения связанных строковых переменных из **S<sub>1</sub>** и **S<sub>2</sub>**, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#2003.
3. Если строковая величина в **S<sub>1</sub>+S<sub>2</sub>** превышает 256 символов (включая код окончания 16#00), инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#2003.
4. Если **S<sub>1</sub>** или **S<sub>2</sub>** перекрывают **D**, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#200C.
5. Если строковые переменные в **S<sub>1</sub>** или **S<sub>2</sub>** не заканчиваются кодом 16#00, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#200E.



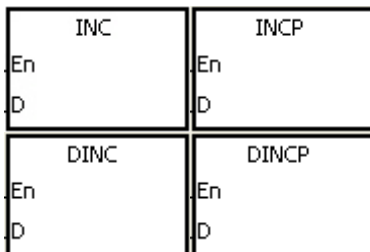
API	Код инструкции			Операнды				Функция			
0115	D	INC	P	D				Добавление единицы к двоичному числу			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



**D** : Операнд результата

**Описание:**

1. Добавление единицы к значению в D.
2. С 32-битными счетчиками может работать только 32-битная инструкция DINC.
3. Когда выполняется 16-битная инструкция, 32 767 плюс 1 равно -32 768. Когда выполняется 32-битная инструкция, 2 147 483 647 плюс 1 равно -2 147 483 648.

**Пример:**

Когда X0.0 включен, к значению в D0 прибавляется единица.



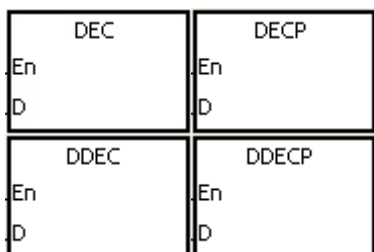
API	Код инструкции			Операнды	Функция
0116	D	DEC	P	D	Вычитание единицы из двоичного числа

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



**D** : Операнд результата

**Описание:**

1. вычитание единицы из значения в **D**.
2. С 32-битными счетчиками может работать только 32-битная инструкция DDEC.
3. Когда выполняется 16-битная инструкция, -32 767 минус 1 равно 32 768. Когда выполняется 32-битная инструкция, -2 147 483 647 минус 1 равно 2 147 483 648.

**Пример:**

Когда X0.0 включен, из значения в D0 вычитается единица.



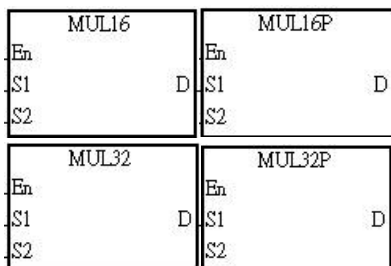
API	Код инструкции			Операнды							Функция					
0117		MUL16 MUL32	P	$S_1 \cdot S_2 \cdot D$							Умножение двоичных чисел для 16 бит Умножение двоичных чисел для 32 бит					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

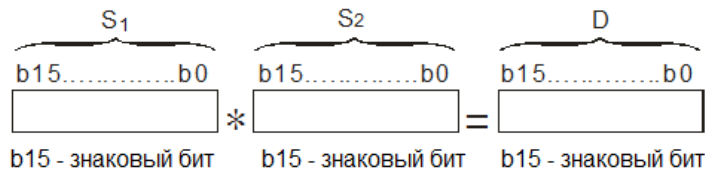
**Символ:**



- $S_1$  : Множимое
- $S_2$  : Множитель
- D : Произведение

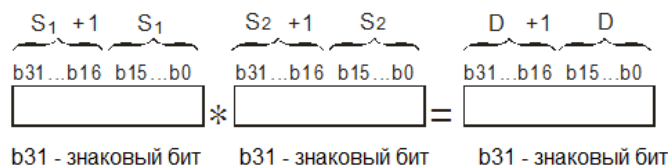
**Описание:**

1. Двоичное значение со знаком в  $S_1$  перемножается с двоичным значением со знаком в  $S_2$ , результат сохраняется в D.
2. Высокоскоростные счетчики HC может использовать только инструкция MUL32.
3. Умножение двоичных чисел для 16 бит:



Полученное произведение является 16-битным значением. Оно сохраняется в D, который является 16-битным регистром. Если значение бита b15 в D равно 0, значение произведения положительное. Если значение b15 в D равно 1, значение произведения отрицательное.

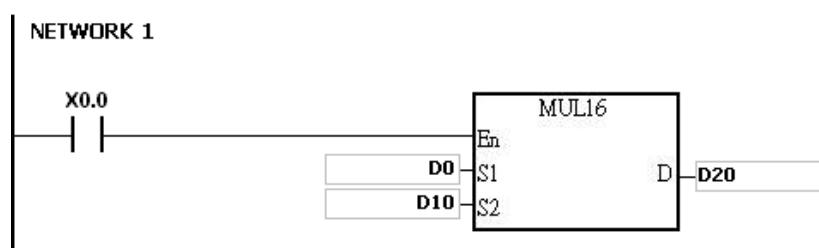
## 4. Умножение двоичных чисел для 32 бит:



Произведение является 32-битным значением. Оно сохраняется в  $(D, D+1)$ , который является 32-битным регистром. Если значение бита  $b_{31}$  в  $D$  равно 0, значение произведения положительное. Если значение  $b_{31}$  в  $D$  равно 1, значение произведения отрицательное.

**Пример:**

16-битное значение в  $D0$  перемножается с 16-битным значением в  $D10$ , произведение сохраняется в  $D20$ . Положительное или отрицательное значение определяется битом слева (bit 15) в  $D20$ . Если значение  $b_{15}$  в  $D$  равно 1, значение произведения отрицательное.



$D0 \times D10 = D20$

16-битное значение  $\times$  16-битное значение = 16-битное значение

**Примечание:**

1. Если произведение не является 16-разрядным значением со знаком и больше максимального 16-битного положительного числа  $K32767$  или меньше минимального отрицательного числа  $K-32768$ , включается флаг переноса  $SM602$ , и будут сохранены только младшие 16 бит значения произведения.
2. Если необходимо получить полный результат 16-битного умножения (32-битное значение), необходимо использовать инструкцию  $API0102 * / *P$ . См. описание инструкции  $API 0102 * / *P$ .
3. Если произведение не является 32-битным значением со знаком и больше максимального 32-битного положительного числа  $K2147483647$  или меньше минимального отрицательного числа  $K-2147483648$ , включается флаг переноса  $SM602$ , и будут сохранены только младшие 32 бита значения произведения.
4. Если необходимо получить полный результат 32-битного умножения (64-битное значение), необходимо использовать инструкцию  $API 0102 D* / D*P$ . См. описание инструкции  $API 0102 D* / D*P$ .

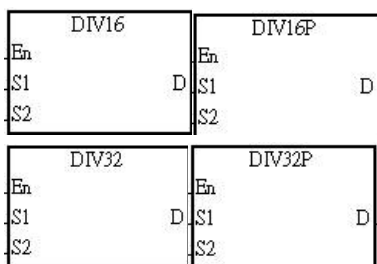
API	Код инструкции			Операнды								Функция				
0118		DIV16 DIV32	P	$S_1 \cdot S_2 \cdot D$								Деление двоичных чисел для 16 бит Деление двоичных чисел для 32 бит				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	●	●		
$S_2$	●	●			●	●	●	●	●		○	○	●	●		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

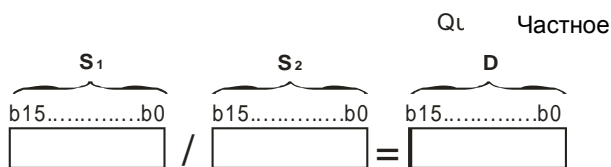
**Символ:**



- $S_1$  : Делимое
- $S_2$  : Делитель
- D : Частное, остаток

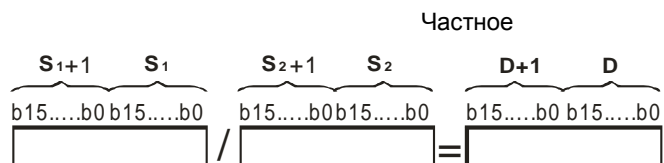
**Описание:**

1. Двоичное значение со знаком в  $S_1$  делится на двоичное значение со знаком в  $S_2$ . Частное сохраняется в D.
2. Высокоскоростные счетчики HC может использовать только инструкция DIV32.
3. Знаковый бит = 0 (положительное значение); знаковый бит = 1 (отрицательное значение)
4. Деление двоичных чисел для 16 бит:



Частное сохраняется в D.

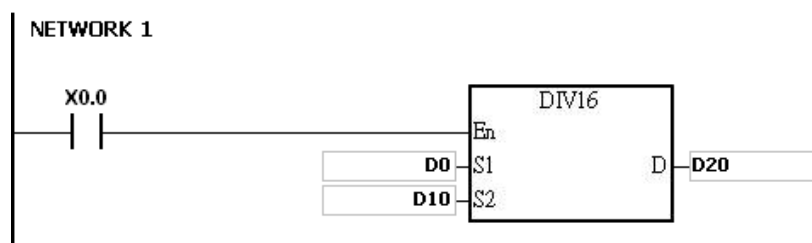
5. Деление двоичных чисел для 32 бит:



**D** занимает два последовательных регистра. Остаток сохраняется в (**D+1**, **D**).

**Пример:**

Когда X0.0 включен, делимое в D0 делится на делитель в D10, частное сохраняется в D20.  
 Положительное или отрицательное значение определяется битом слева (bit 15) в D20.



**Примечание:**

1. Если используемый объект недоступен, инструкция не выполняется, SM0 включен, в регистре SR0 will be 16#2003.
2. If the divisor used is 0, the instruction will not be executed, включается флаг SM0 и в SR0 записывается код ошибки 16#2012.
3. Если необходимо сохранить остаток деления, используйте инструкцию API 0103 “/” (Деление двоичных чисел). См. описание инструкции API 0103 “/”.

## 6.3 Инструкции преобразования данных

### 6.3.1 Описание инструкций преобразования данных

API	Код инструкции			Операнды								Функция				
0200	D	BCD	P	S · D								Преобразование двоичного числа в двоично-десятичное число				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○	○				
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символ:

BCD	BCDP
En	En
S	D
D	D

DBCD	DBCDP
En	En
S	D
D	D

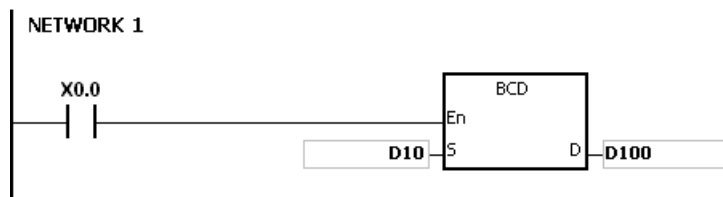
S : Источник данных  
 D : Результат преобразования

Описание:

1. Двоичное значение в **S** преобразуется в двоично-десятичное значение и результат преобразования сохраняется в **D**.
2. 32-битный счетчик может использовать только 32-битная инструкция DBCD.
3. Четыре инструкции основных арифметических операций, инструкция INC и инструкция DEC содержат двоичные числа. Чтобы показать десятичное значение на дисплее, можно использовать инструкцию BCD для преобразования двоичного значения в двоично-десятичное значение.

Пример:

1. Когда X0.0 включен, двоичное значение в D10 преобразуется в двоично-десятичное, результат сохраняется в D100.



2. Если  $D10=16\#04$   $D2=1234$ , результат преобразования будет  $D100=16\#1234$ .

**Примечание:**

1. Если результат преобразования выходит за пределы диапазона от 0 до 9 999, инструкция BCD не выполняется, включается флаг  $SM0$ , в  $SR0$  записывается код ошибки  $16\#200D$  (двоично-десятичное значение представлено шестнадцатеричным значением, но одна из цифр не находится в пределах от 0 до 9).
2. Если результат преобразования выходит за пределы диапазона от 0 до 99 999 999, инструкция DBCD не выполняется, включается флаг  $SM0$ , в  $SR0$  записывается код ошибки  $16\#200D$  (двоично-десятичное значение представлено шестнадцатеричным значением, но одна из цифр не находится в пределах от 0 до 9).



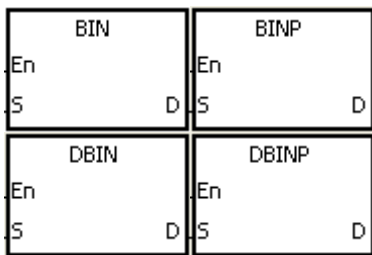
API	Код инструкции			Операнды							Функция					
0201	D	BIN	P	S · D							Преобразование двоично-десятичного числа в двоичное число					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○	○				
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



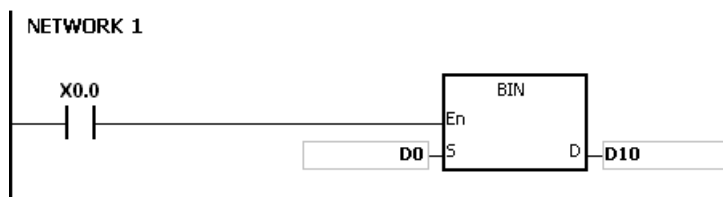
**S** : Источник данных  
**D** : Результат преобразования

**Описание:**

1. Двоично-десятичное значение в **S** преобразуется в двоичное значение, результат сохраняется в **D**.
2. 16-битное двоично-десятичное значение в **S** должно находиться в диапазоне между 0 и 9 999, 32-битное двоично-десятичное значение в **S** должно находиться в диапазоне между 0 и 99 999 999.
3. 32-битный счетчик может использовать только 32-битная инструкция DBIN.
4. Константы и шестнадцатеричные значения автоматически преобразуются в двоичные значения. Поэтому нет необходимости использовать эту инструкцию.

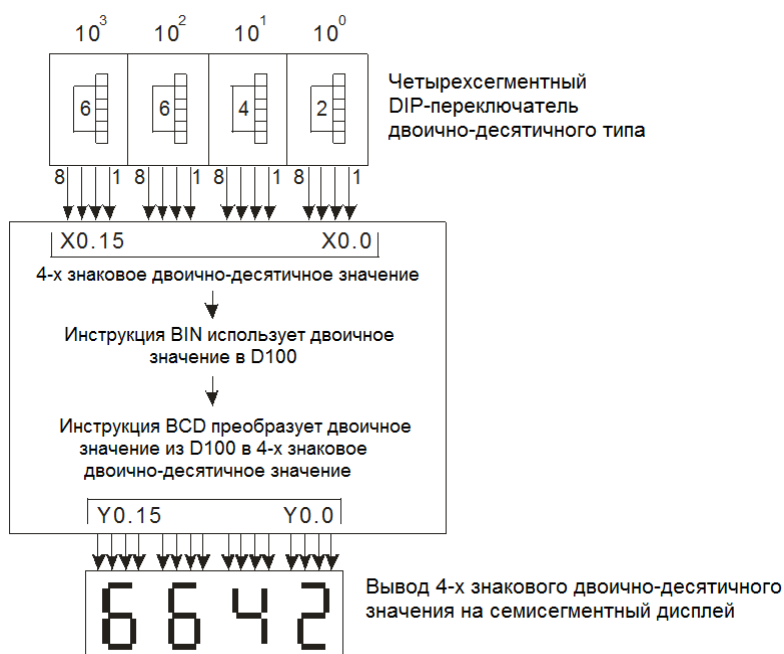
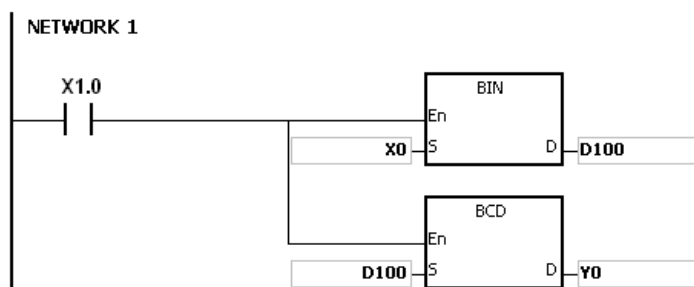
**Пример:**

Когда X0.0 включен, двоично-десятичное значение в D0 преобразуется в двоичное значение, которое сохраняется в D10.



**Примечание:**

1. Если значение в **S** не является двоично-десятичным, инструкция BIN не выполняется, включается флаг SMO, в SR0 записывается код ошибки 16#200D (двоично-десятичное значение представлено шестнадцатеричным значением, но одна из цифр не находится в пределах от 0 до 9).
2. Применение инструкций BCD и BIN:
  - До того, как двоично-десятичное значение DIP-переключателя будет считано в ПЛК, необходимо использовать инструкцию BIN для преобразования данных в двоичное значение и только тогда сохранить результат преобразования в ПЛК.
  - Если необходимо отображать данные, хранящиеся в ПЛК, на семисегментном дисплее как двоично-десятичный тип данных, необходимо использовать инструкцию BCD для преобразования данных в двоично-десятичное значение перед отправкой этих данных на семисегментный дисплей.
  - Когда X1.0 включен, двоично-десятичное значение в X0.0 ~ X0.15 преобразуется в двоичное значение, и результат преобразования сохраняется в D100. Затем двоичное значение в D100 преобразуется в двоично-десятичное значение, а результат преобразования сохраняется в Y0.0 ~ Y0.15.



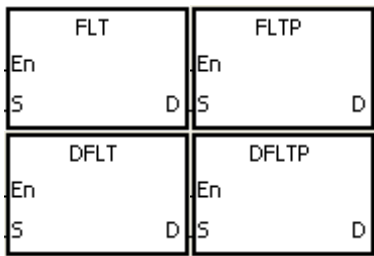
API	Код инструкции			Операнды						Функция					
0202	D	FLT	P	S · D						Преобразование двоичного целого числа в двоичное число с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S</b>	●	●			●	●	●	●	●		○	○				
<b>D</b>		●					●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●	●		●	●	●				●	●	
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



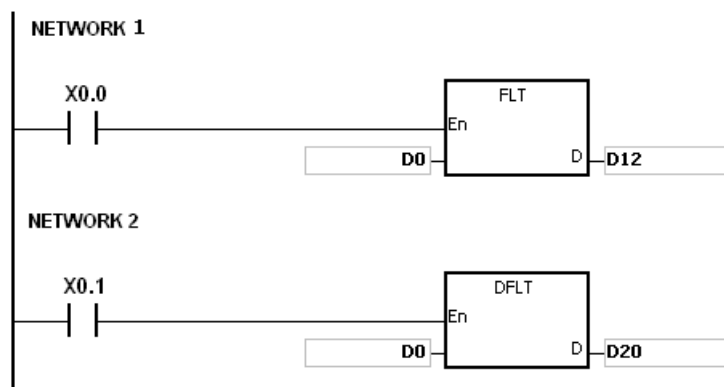
**S** : Источник данных  
**D** : Результат преобразования

**Описание:**

- Инструкция используется для преобразования двоичного целого числа в двоичное число с плавающей запятой.
- Операнд **S** в инструкции FLT не использует 32-битный счетчик, но не является объектом E.
- Операнд **S** в инструкции FLT занимает один регистр, операнд **D** занимает два регистра.
- Операнд **S** в инструкции DFLT занимает два регистра, операнд **D** в инструкции DFLT занимает два регистра.
  - Когда абсолютное значение результата преобразования больше, чем максимальное значение, которое может быть представлено числом с плавающей запятой, включается флаг SM602, а в **D** сохраняется максимальное возможное число с плавающей запятой.
  - Когда абсолютное значение результата преобразования меньше, чем минимальное значение, которое может быть представлено числом с плавающей запятой, включается флаг SM601, а в **D** сохраняется минимальное возможное число с плавающей запятой.
  - Если результат преобразования равен 0, включается флаг SM600.

**Пример 1:**

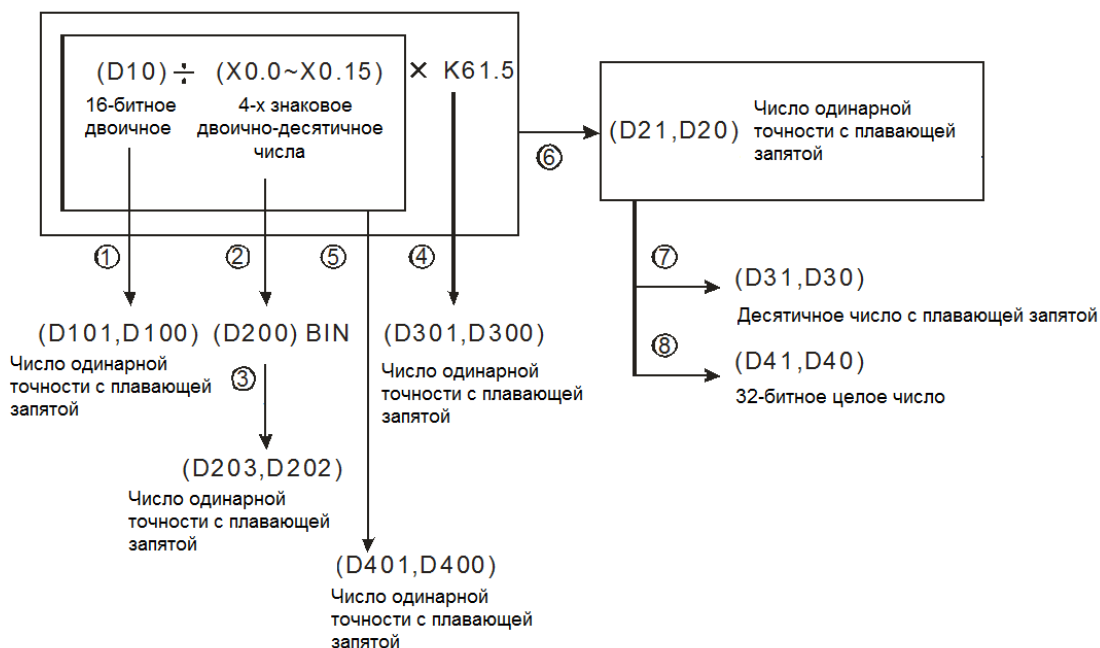
1. Когда X0.0 включен, двоичное целое значение в D0 преобразуется в число одинарной точности с плавающей запятой, результат сохраняется в (D13, D12).
2. Когда X0.1 включен, двоичное целое значение в (D1, D0) преобразуется в число одинарной точности с плавающей запятой, результат сохраняется в (D21, D20).
3. Предположим, что значение в 16-битном регистре D0 равно 10. Когда X0.0 включен, 10 преобразуется в число одинарной точности с плавающей запятой 16#41200000, и это значение сохраняется в 32-битный регистр (D13, D12).
4. Предположим, что значение в 32-битном регистре (D1, D0) равно 100,000. Когда X0.1 включен, 100,000 преобразуется в число одинарной точности с плавающей запятой 16#47C35000, и это значение сохраняется в 32-битный регистр (D21, D20).

**Пример 2:**

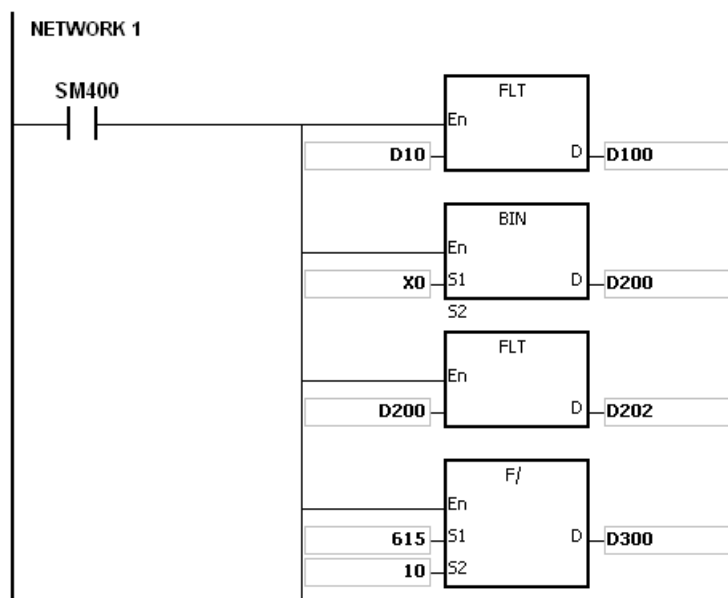
Пользователь может использовать прикладные инструкции для выполнения следующего расчета.

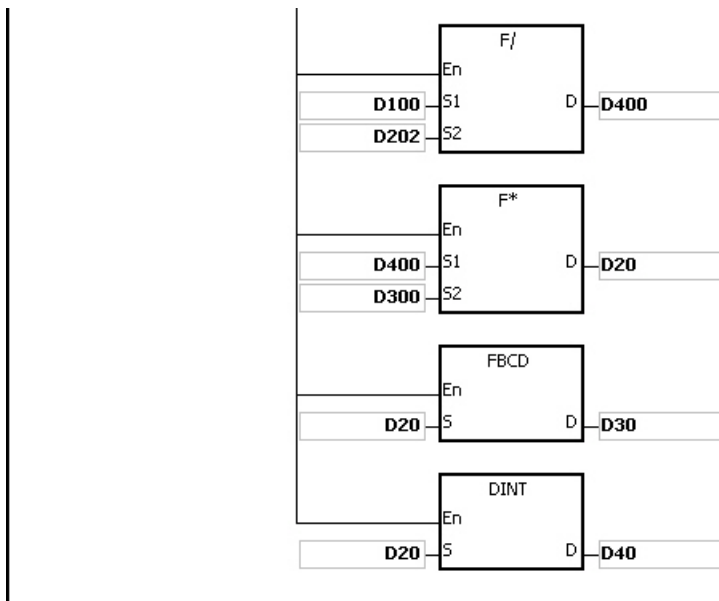
- Двоичное целое число в D10 преобразуется в число одинарной точности с плавающей запятой, и результат преобразования сохраняется в (D101, D100).
- Двоично-десятичное значение в X0.0 ~ X0.15 преобразуется в двоичное значение, а результат преобразования сохраняется в D200.
- Двоичное целое число в D200 преобразуется в число одинарной точности с плавающей запятой, и результат преобразования сохраняется в (D203, D202).
- Константа 615 делится на константу 10, а частное, которое является числом одинарной точности с плавающей запятой, сохраняется в (D301, D300).
- Число одинарной точности с плавающей запятой в (D101, D100) делится на число одинарной точности с плавающей запятой в (D203, D202), а частное, которое является числом одинарной точности с плавающей запятой, сохраняется в (D401, D400).

- Число одинарной точности с плавающей запятой в (D401, D400) умножается на число одинарной точности с плавающей запятой в (D301, D300), произведение, которое является числом одинарной точности с плавающей запятой, сохраняется в (D21, D20).
- Число одинарной точности с плавающей запятой в (D21, D20) преобразуется в десятичное число с плавающей запятой, а результат преобразования сохраняется в (D31, D30).
- Число одинарной точности с плавающей запятой в (D21, D20) преобразуется в двоичное целое число, а результат преобразования сохраняется в (D41, D40).



6





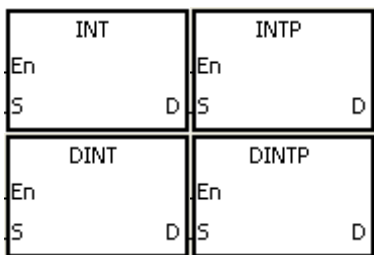
API	Код инструкции			Операнды						Функция					
0204	D	INT	P	S · D						Преобразование 32-битового числа с плавающей запятой в двоичное целое число					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○					
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



**S** : Источник данных  
**D** : Результат преобразования

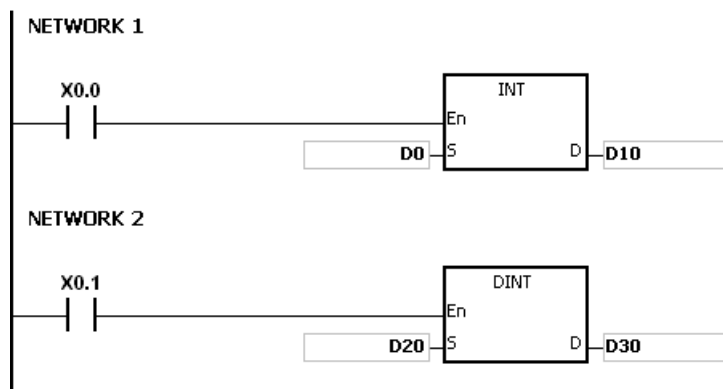
**Описание:**

1. Число одинарной точности с плавающей запятой в **S**, преобразуется в двоичное целое число. Двоичное число с плавающей запятой округляется до ближайшего целого и становится целым двоичным числом, которое сохраняется в **D**.
2. Операнд **S** в инструкции INT занимает один регистр, операнд **D** занимает два регистра.
3. Операнд **S** в инструкции DINT занимает два регистра, операнд **D** в инструкции DINT занимает два регистра.
4. Операнд **S** в инструкции INT не использует 32-битный счетчик, но не является объектом E.
5. Инструкции INT и FLT противоположны по действию.
6. Если результат преобразования равен 0, включается флаг нуля SM600.
7. Во время преобразования, если число с плавающей запятой округлено до ближайшего целого, включается флаг заимствования SM601.
8. Когда результат преобразования превышает допустимый диапазон, включается флаг переноса SM602.

9. Для инструкций INT/INTP, диапазон результата преобразования от -32 768 до 32 767.
10. Для инструкций DINT/DINTP, диапазон результата преобразования от -2 147 483 648 до 2 147 483 647.

**Пример:**

1. Когда X0.0 включен, число одинарной точности с плавающей запятой (D1, D0) преобразуется в двоичное целое, результат сохраняется в D10. Двоичное число с плавающей запятой округляется до ближайшего целого.
2. Когда X0.1 включен, число одинарной точности с плавающей запятой (D21, D20) преобразуется в двоичное целое, результат сохраняется в (D31, D30). Двоичное число с плавающей запятой округляется до ближайшего целого.

**Примечание:**

Если значение в **S** выходит за пределы допустимого для чисел с плавающей запятой значений, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#2013.



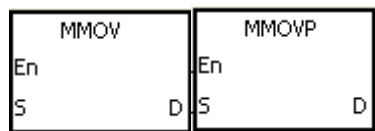
API	Код инструкции			Операнды								Функция				
0206		MMOV	P	S · D								Преобразование 16-битного в 32-битное значение				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●		○	○	○	○		
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●	●				●	●	
D			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



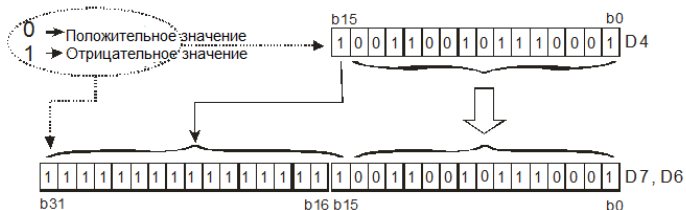
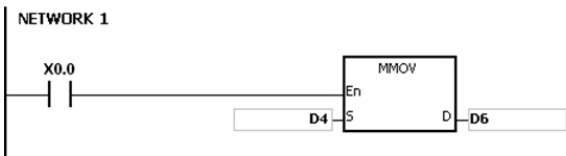
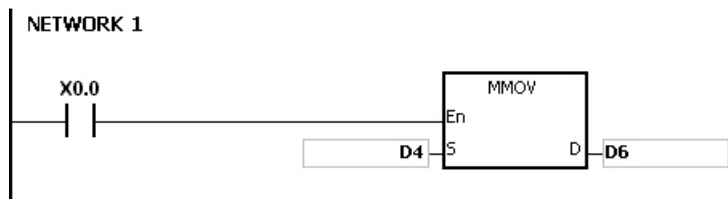
**S** : Источник данных  
**D** : Результат преобразования

**Описание:**

16-битные данные из **S** преобразуются в 32-битные данные в **D**. Знаковый бит копируется в операнд результата.

**Пример:**

Когда X0.0 включен, значение бита b15 в D4 передается в биты b15~b31 в (D7, D6). Данные в (D7, D6) имеют отрицательное значение.



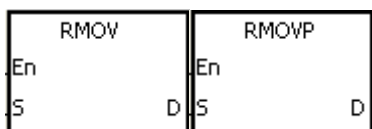
API	Код инструкции			Операнды								Функция					
0207		RMOV	P	S · D								Преобразование 32-битного в 16-битное значение					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○		○	○		
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S			●				●						
D		●			●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



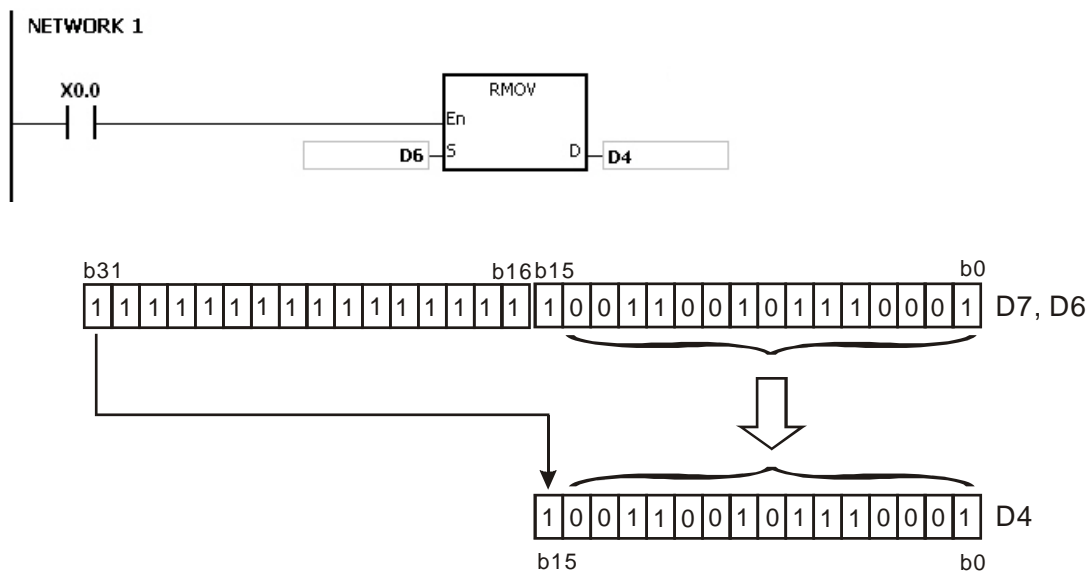
- S : Источник данных
- D : Результат преобразования

**Описание:**

32-битные данные из **S** преобразуются в 16-битные данные в **D**. Знаковый бит копируется в операнд результата.

**Пример:**

Когда X0.0 включен, значение бита b31 в D7 передается в бит b15 в D4, значения b0~b14 передаются соответствующим битам, а значения b15~b30 игнорируются.



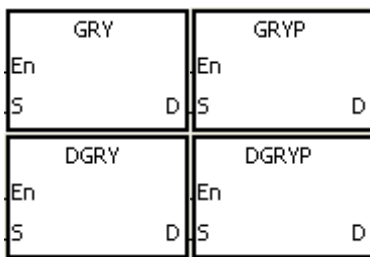
API	Код инструкции			Операнды							Функция					
0208	D	GRY	P	S · D							Преобразование двоичного числа в код Грея					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



**S** : Источник данных  
**D** : Результат преобразования

**Описание:**

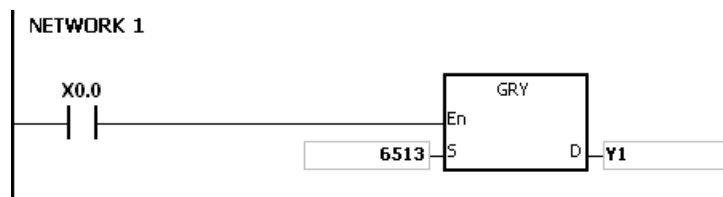
1. Двоичное значение в **S** преобразуется в код Грея и результат сохраняется в **D**.
2. Только 32-битная инструкция DGRY может использовать 32-битный счетчик.
3. Значение в **S** должно находиться в пределах допустимого диапазона.

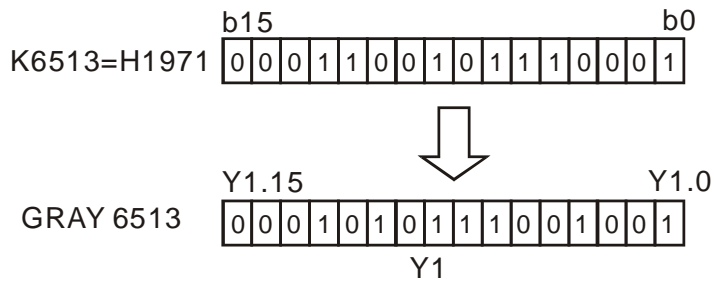
Значение в **S** для 16-битной инструкции должно находиться в диапазоне от 0 до 32 767.

Значение в **S** для 32-битной инструкции должно находиться в диапазоне от 0 до 2 147 483 647.

**Пример:**

Когда X0.0 включен, константа 6513 преобразуется в код Грея, результат сохраняется в Y1.0~Y1.15.



**Примечание:**

Если значение в **S** меньше 0, возникает ошибка операции, инструкция не выполняется, включается флаг **SM0** и в **SR0** записывается код ошибки 16#2003.

API	Код инструкции			Операнды							Функция					
0209	D	GBIN	P	S · D							Преобразование кода Грея в двоичное число					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●				●	●	
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



**S** : Источник данных  
**D** : Результат преобразования

**Описание:**

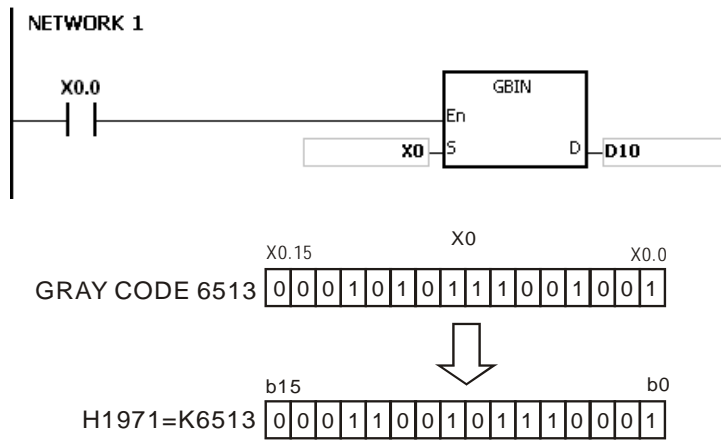
1. Код Грея в **S** преобразуется в двоичное значение и результат сохраняется в **D**.
2. Инструкция используется для преобразования кода Грея в абсолютном энкодере, который подключен к входу ПЛК, в двоичное значение и результат преобразования сохраняется в заданном регистре.
3. Только 32-битная инструкция DGBIN может использовать 32-битный счетчик.
4. Значение в **S** должно находиться в пределах допустимого диапазона.

Значение в **S** для 16-битной инструкции должно находиться в диапазоне от 0 до 32 767.

Значение в **S** для 32-битной инструкции должно находиться в диапазоне от 0 до 2 147 483 647.

**Пример:**

Когда X0.0 включен, код Грея от абсолютного энкодера, подключенного к входам X0.0~X0.15, преобразуется в двоичное значение, результат сохраняется в D10.



**Примечание:**

Если значение в **S** меньше 0, возникает ошибка операции, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнды							Функция			
0210	D	NEG	P	D							Второе дополнение			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



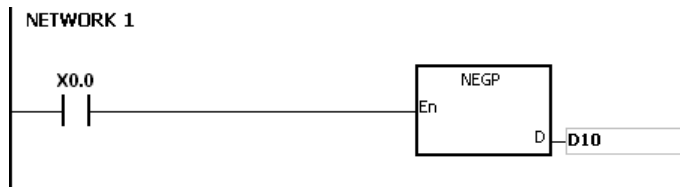
**D** : Операнд операции второго дополнения

**Описание:**

1. Инструкция используется для преобразования отрицательного двоичного значения в абсолютное значение.
2. 32-битный счетчик может использовать только 32-битная инструкция DNEG.
3. Как правило, используются импульсные инструкции NEGP и DNEGP.

**Пример 1:**

Когда X0.0 включается, все биты в D0 инвертируются (0 в 1 и 1 в 0) и к результату добавляется 1. Конечное значение сохраняется в исходном регистре D10.

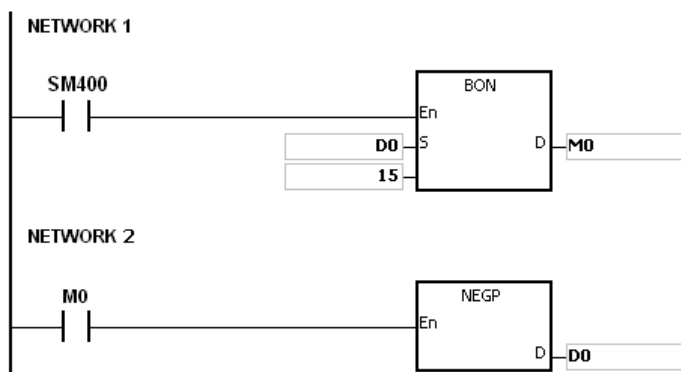


**Пример 2:**

Абсолютное значение отрицательного значения:

1. Когда значение 15<sup>го</sup> бита в D0 равно 1, включается M0. (значение в D0 отрицательно)
2. Когда M0 включен, команда NEG используется для получения второго дополнения отрицательного значения

в D0. (получено соответствующее положительное значение)

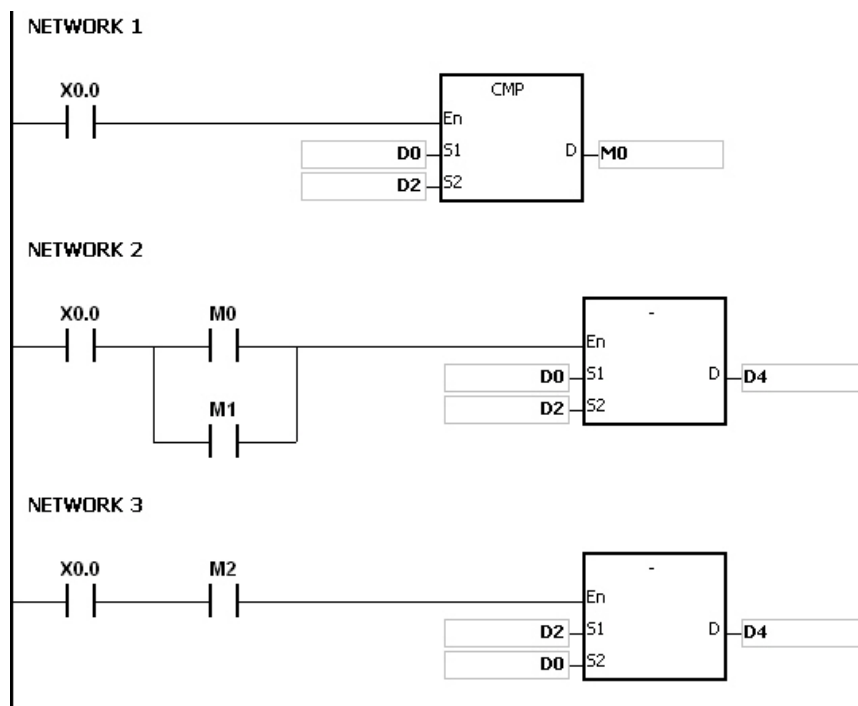


**Пример 3:**

Абсолютное значение разницы двух значений:

Предположим, X0.0 включен.

1. Когда значение в D0 больше значения в D2, включается M0.
2. Когда значение в D0 равно значению в D2, включается M1.
3. Когда значение в D0 меньше значения в D2, включается M2.
4. Значение в D4 положительное.





**Примечания:**

Представление значения и его абсолютная величина:

1. Положительное или отрицательное значение данных зависит от значения старшего бита в регистре. Если значение старшего бита в регистре равно 0, данные имеют положительное значение. Если оно равно 1, данные имеют отрицательное значение.
2. Отрицательное значение может быть преобразовано в абсолютное значение с помощью инструкции NEG.

(D0)=2

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0)=1

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

(D0)=0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

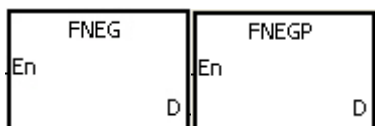
API	Код инструкции			Операнды								Функция					
0211		FNEG	P	D								Изменение знака 32-битного числа с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символ:**



**D** : Операнд изменения знака

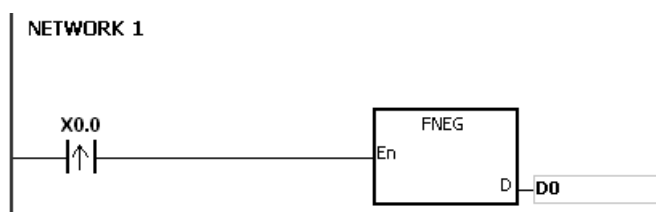
**Описание:**

Изменение знака числа одинарной точности с плавающей запятой в операнде **D**.

**Пример:**

Перед выполнением инструкции значение в (D1, D0) является отрицательным значением 16#AE0F9000. Когда X0.0 включен, знак числа одинарной точности с плавающей запятой в (D1, D0) изменяется на противоположный. Другими словами, после выполнения инструкции значение в (D1, D0) является положительным значением 16#2E0F9000.

Перед выполнением инструкции значение в (D1, D0) является положительным значением 16#2E0F9000. Когда X0.0 включен, знак числа одинарной точности с плавающей запятой в (D1, D0) изменяется на противоположный. Другими словами, после выполнения инструкции значение в (D1, D0) является отрицательным значением 16#AE0F9000.



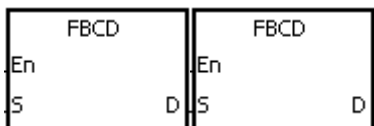
API	Код инструкции			Операнды							Функция					
0212		FBCD	P	S · D							Преобразование двоичного числа с плавающей запятой в десятичное число с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●		●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символ:**



- S** : Источник данных
- D** : Результат преобразования

**Описание:**

1. Число одинарной точности с плавающей запятой в **S** преобразуется в десятичное число с плавающей запятой, результат сохраняется в **D**.
2. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), and SM602 (флаг переноса)

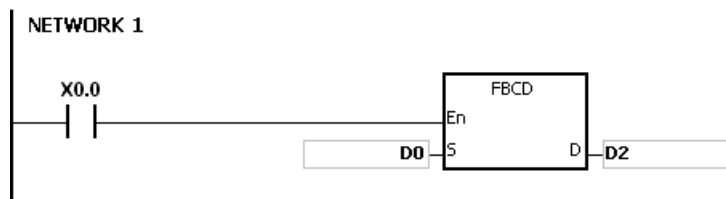
Когда абсолютное значение результата преобразования больше, чем максимальное значение, которое может быть представлено числом с плавающей запятой, включается флаг SM602.

Когда абсолютное значение результата преобразования меньше, чем минимальное значение, которое может быть представлено числом с плавающей запятой, включается флаг SM601.

Если результат преобразования равен 0, включается флаг нуля SM600.

**Пример:**

Когда X0.0 включен, число с плавающей запятой в (D1, D0) преобразуется в десятичное число с плавающей запятой, результат сохраняется в (D3, D2).



Двоичное число с плавающей запятой 

D 1	D 0
-----	-----

 Основание: 23 бита, экспонента: 8 бит, знак: 1 бит



Десятичное число с плавающей запятой 

D 3	D 2
-----	-----

 Экспонента      Основание  
Математическая запись  $\Rightarrow$   $[D2] \times 10^{[D3]}$

**Примечание:**

Если значение в **S** выходит за пределы допустимого для чисел с плавающей запятой диапазона, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2013.

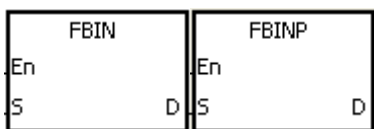
API	Код инструкции			Операнды							Функция					
0213		FBIN	P	S · D							Преобразование десятичного числа с плавающей запятой в двоичное число с плавающей запятой					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●		●	●		●					
D		●			●	●	●	●			●					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символ:**



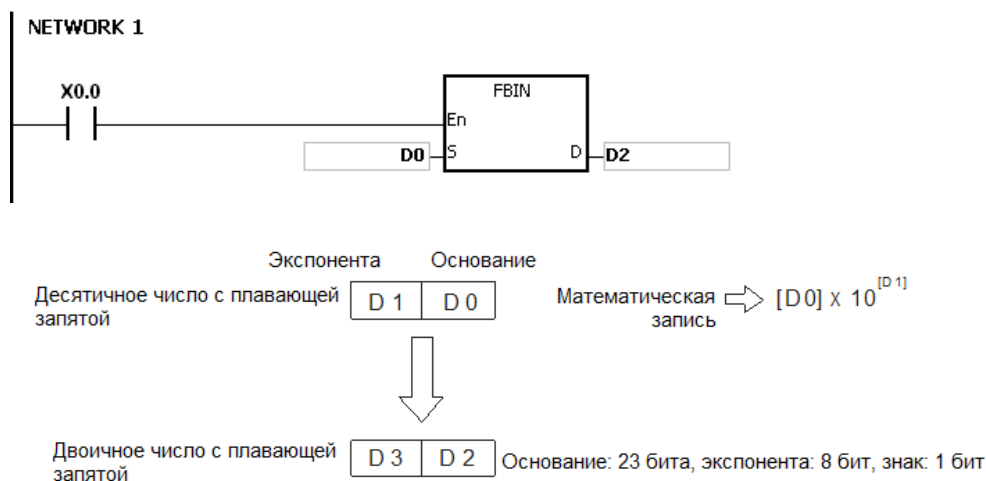
- S : Источник данных
- D : Результат преобразования

**Описание:**

- Десятичное число с плавающей запятой в S преобразуется в число одинарной точности с плавающей запятой, результат сохраняется в D.
- Предположим, что значение в S равно 1234, а в S+1 равно 3. Значение в S преобразуется в  $1,234 \times 10^6$ .
- Значение в D должно быть числом одинарной точности с плавающей запятой, а значения в S и S + 1 представляют собой десятичное основание и десятичный показатель степени соответственно.
- Инструкция FBIN используется для преобразования десятичного числа с плавающей запятой в число одинарной точности с плавающей запятой.
- Десятичное с плавающей запятой основание находится в пределах диапазона от -9 999 до +9 999, десятичная экспонента находится в пределах диапазона от -41 до +35, таким образом, реальный диапазон значений десятичных чисел с плавающей запятой в ПЛК:  $\pm 1175 \times 10^{-41}$  и  $\pm 3402 \times 10^{+35}$ . Если результат преобразования равен 0, включается флаг нуля SM600.

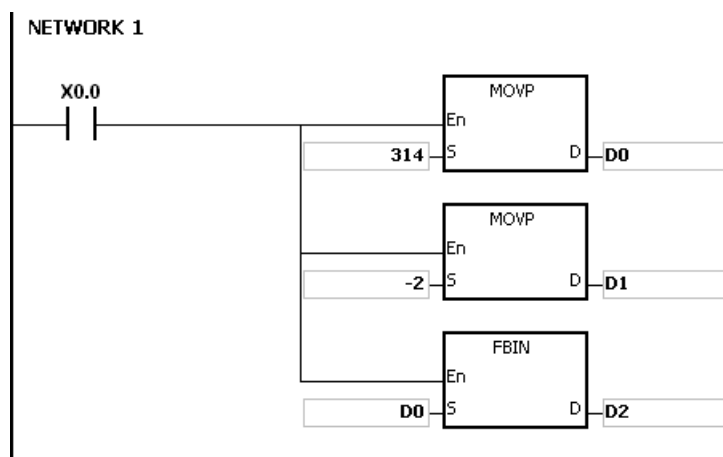
**Пример 1:**

- Когда X0.0 включен, десятичное с плавающей запятой значение в (D1, D0) преобразуется в число одинарной точности с плавающей запятой, результат преобразования сохраняется в (D3, D2).



**Пример 2:**

1. Перед выполнением операции с числами с плавающей запятой необходимо использовать инструкцию FLT для преобразования двоичного целого в число одинарной точности с плавающей запятой. Инструкция FBIN может использоваться для преобразования числа с плавающей запятой в число одинарной точности с плавающей запятой.
2. Когда X0.0 включен, K314 и K-2 перемещаются в D0 и D1 соответственно и объединяются в десятичное число с плавающей запятой ( $3,14=314 \times 10^{-2}$ ).



**Примечание:**

Если десятичное с плавающей запятой основание в **S** находится вне пределов допустимого диапазона -9,999...999, или десятичная с плавающей точкой экспоненты в **S** находится вне пределов допустимого диапазона -41...+35, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2013.

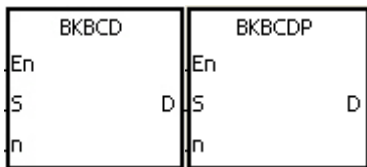
API	Код инструкции			Операнды							Функция					
0214		BKBCD	P	S · n · D							Преобразование двоичных чисел в блоке в десятичные числа в блоке					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
n	●	●			●	●		●	●				○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



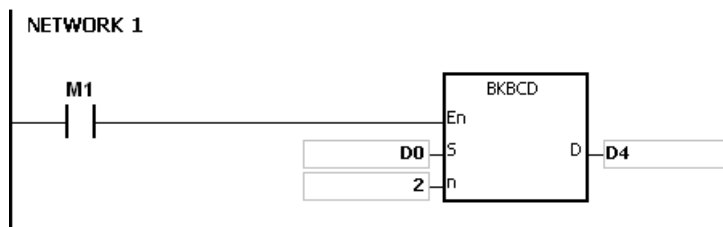
- S** : Источник данных
- n** : Длина данных
- D** : Результат преобразования

**Описание:**

1. **n** частей данных (двоичных) в операндах, начиная от **S** преобразуются в двоично-десятичные значения, результат сохраняется в **D**.
2. Значение **n** находится в диапазоне от 1 до 256.

**Пример:**

Когда M1 включается, двоичные значения в D0 и D1 преобразуются в двоично-десятичные, результат сохраняется в D4 и D5.



**Примечания:**

1. Если **n** меньше 1 или больше 256, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200B.

2. Если значения в  $S+n-1$  и  $D+n-1$  превышают допустимый диапазон, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
3. Если результат преобразования находится за пределами диапазона от 0 до 9 999, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200D (двоично- десятичное значение представлено шестнадцатеричным числом, и одна из цифр не находится в диапазоне от 0 до 9).
4. Если  $S \sim S+n-1$  перекрывает  $D \sim D+n-1$ , инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200C.

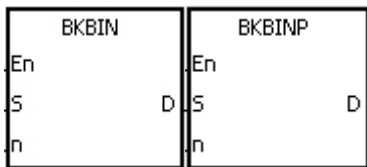


API	Код инструкции				Операнды							Функция				
0215		ВКВІN	P		S · n · D							Преобразование двоичных чисел в блоке в двоично-десятичные числа в блоке				
Объекты	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F	
S	●			●	●		●	●								
n	●			●	●		●	●				○	○			
D	●			●	●		●									

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



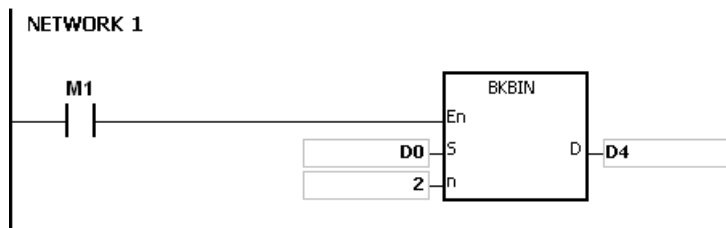
- S** : Источник данных
- n** : Длина данных
- D** : Результат преобразования

**Описание:**

1. n частей данных (двоично-десятичных) в операндах, начиная от S преобразуются в двоичные значения, результат сохраняется в D.
2. Значение n находится в диапазоне от 1 до 256.
3. Двоично-десятичные значения в S находятся в диапазоне от 0 до 9 999.

**Пример:**

Когда M1 включается, двоично-десятичные значения в D0 и D1 преобразуются в двоичные, результат сохраняется в D4 и D5.



**Примечания:**

1. Если  $n$  меньше 1 или больше 256, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200B.
2. Если значения в  $S+n-1$  и  $D+n-1$  превышают допустимый диапазон, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
3. Если результат преобразования находится за пределами диапазона от 0 до 9 999, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200D (двоично-десятичное значение представлено шестнадцатеричным числом, и одна из цифр не находится в диапазоне от 0 до 9).
4. Если  $S \sim S+n-1$  перекрывает  $D \sim D+n-1$ , инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200C.

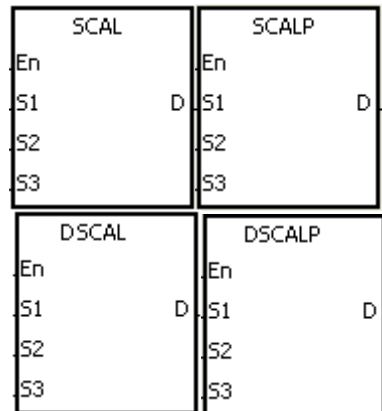
API	Код инструкции			Операнды							Функция					
0216		SCAL	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							Операция масштабирования значений					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●		○	○	○	○		
$S_2$	●	●			●	●		●	●		○	○	○	○		
$S_3$	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
$S_3$		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



- $S_1$  : Источник данных
- $S_2$  : Наклон
- $S_3$  : Смещение
- D : Операнд результата

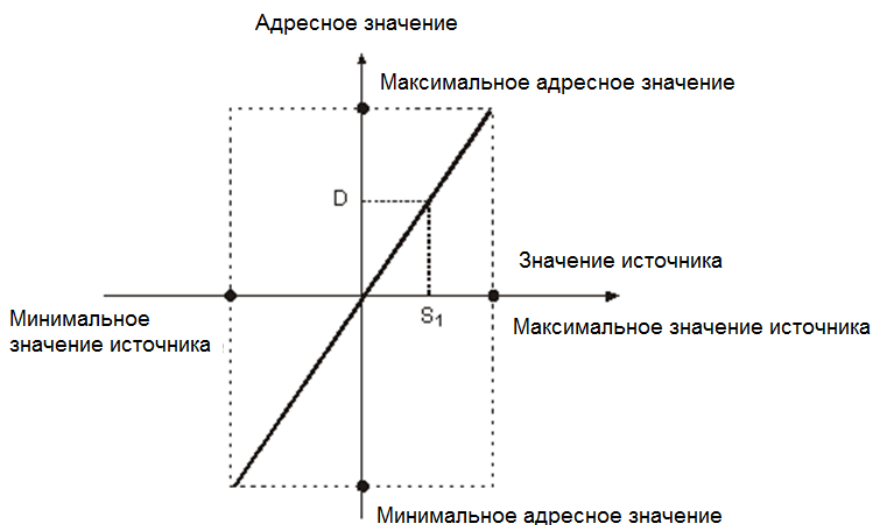
**Описание:**

1. Функционирование инструкции:  $D=(S_1 \times S_2) \div 1,000 + S_3$
2. Чтобы получить значения в  $S_2$  и  $S_3$ , сначала необходимо использовать формулы расчета наклона и смещения ниже, а затем округлить результаты до ближайшего целого. Окончательные 16-битные значения вводятся в  $S_2$  и  $S_3$ .

Формула наклона:  $S_2 = [(\text{Максимальное адресное значение} - \text{Минимальное адресное значение}) \div (\text{Максимальное значение источника} - \text{Минимальное значение источника})] \times 1,000$

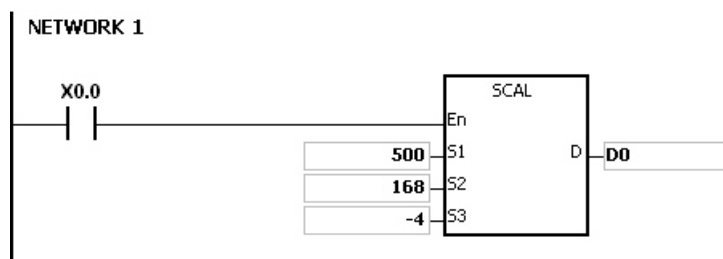
Формула смещения:  $S_3 = \text{Максимальное адресное значение} - \text{Минимальное адресное значение} \times S_2 \div 1,000$

Выходная кривая показана ниже:



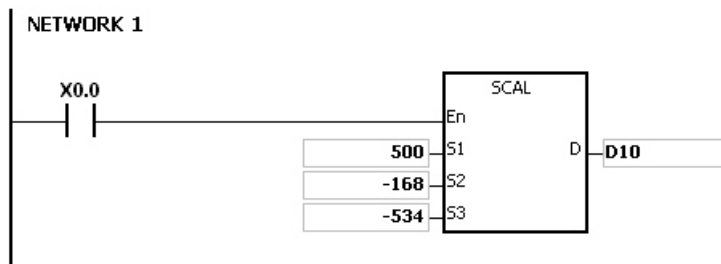
**Пример 1:**

1. Например, значения  $S_1$ ,  $S_2$  и  $S_3$  равны 500, 168 и -4 соответственно. Когда X0.0 включен, инструкция SCAL выполняется и масштабируемое значение сохраняется в D0.
2. Функционирование инструкции:  $D0 = (500 \times 168) \div 1,000 + (-4) = 80$



**Пример 2:**

1. Например, значения  $S_1$ ,  $S_2$  и  $S_3$  равны 500, 168 и 534 соответственно. Когда X0.0 включен, инструкция SCAL выполняется и масштабируемое значение сохраняется в D10.
2. Функционирование инструкции:  $D10 = (500 \times 168) \div 1,000 + 534 = 450$



**Примечание:**

1. Инструкцию SCAL можно использовать только, если известны наклон и смещение. Если их значения неизвестны, необходимо использовать инструкцию SCLP.
2. Когда выполняется 16-битная инструкция, значение, введенное в  $S_2$ , должно находиться в пределах от -32 768 до 32 767 (практическое значение находится в пределах от -32,768 до 32,767). Если значение в  $S_2$  превышает данный диапазон, для выполнения операции используйте инструкцию SCLP.
3. Когда выполняется 32-битная инструкция, значение, введенное в  $S_2$ , должно находиться в пределах от -2 147 483 648 до 2 147 483 647 (практическое значение находится в пределах от -2 147 483 648 до 2 147 483 647). Если значение в  $S_2$  превышает данный диапазон, для выполнения операции используйте инструкцию SCLP.
4. Когда используется уравнение наклона, максимальное значение источника должно быть больше минимального значения источника. Однако максимальное значение назначения не обязательно больше минимального адресного значения.

Когда выполняется 16-битная инструкция, если значение в  $D$  больше 32 767, значение, сохраняемое в  $D$ , будет равно 32 767. Если значение в  $D$  меньше -32 768, значение, сохраняемое в  $D$ , будет -32 768.

Когда выполняется 32-битная инструкция, если значение в  $D$  больше 2 147 483 647, значение, сохраняемое в  $D$ , будет равно 2 147 483 647. Если значение в  $D$  меньше -2 147 483 647, значение, сохраняемое в  $D$ , будет -2 147 483 647.

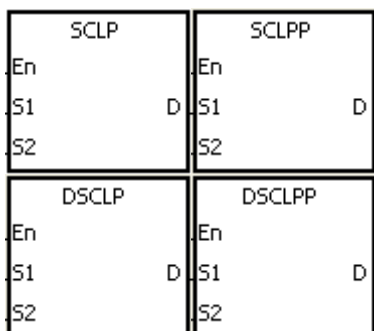
API	Код инструкции			Операнды							Функция						
0217	D	SCLP	P	$S_1 \cdot S_2 \cdot S_3 \cdot D$							Тип параметра операции масштабирования значений						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		○
$S_2$	●	●			●	●	●	●	●							
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●		●				
$S_2$		●	●		●	●	●		●				
D		●	●		●	●	●		●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



- $S_1$  : Источник данных
- $S_2$  : Параметр
- D : Операнд результата

**Описание:**

1. 32-битный счетчик использует только 32-битная инструкция DSLP.
2. Значения, используемые операндом  $S_1$

Значение	16-битная инструкция	32-битная инструкция	
		SM685 включен	SM685 выключен
Константа	○	X	○
Шестнадцатеричное число	○	X	○
Число с плавающей запятой	X	○	X

Флаг SM685 (флаг использования числа с плавающей запятой) работает только с 32-битной инструкцией.

3. Операнд  $S_2$ , используемый в 16-битной инструкции, задается следующим образом.

Номер операнда	Параметр	Диапазон задания
$S_2$	Максимальное значение источника	-32,768~32,767
$S_{2+1}$	Минимальное значение источника	-32,768~32,767
$S_{2+2}$	Максимальное адресное значение	-32,768~32,767
$S_{2+3}$	Минимальное адресное значение	-32,768~32,767

4. Операнд  $S_2$  для 16-битной инструкции занимает четыре регистра.  
 5. Операнд  $S_2$ , используемый в 32-битной инструкции, задается следующим образом.

Номер операнда	Параметр	Диапазон задания	
		Целое число	Число с плавающей запятой
$S_2 \setminus S_{2+1}$	Максимальное значение источника	-2,147,483,648~ 2,147,483,647	Диапазон 32-битных чисел с плавающей запятой
$S_{2+2} \setminus 3$	Минимальное значение источника		
$S_{2+4} \setminus 5$	Максимальное адресное значение		
$S_{2+6} \setminus 7$	Минимальное адресное значение		

6. Операнд  $S_2$  для 32-битной инструкции занимает восемь регистров.  
 7. Если в 32-битной инструкции используется число с плавающей запятой, включается флаг SM658. Если используется десятичное целое число, флаг SM685 отключен.  
 8. Формула расчета результата:  $D = [(S_1 - \text{Минимальное значение источника}) \times (\text{Максимальное адресное значение} - \text{Минимальное адресное значение})] \div (\text{Максимальное значение источника}) + \text{Минимальное адресное значение}$   
 9. Отношение между значением источника и адресным значением:

$$y = kx + b$$

$$y = \text{Адресное значение (D)}$$

$$k = \text{Наклон} = (\text{Максимальное адресное значение} - \text{Минимальное адресное значение}) \div (\text{Максимальное значение источника} - \text{Минимальное значение источника})$$

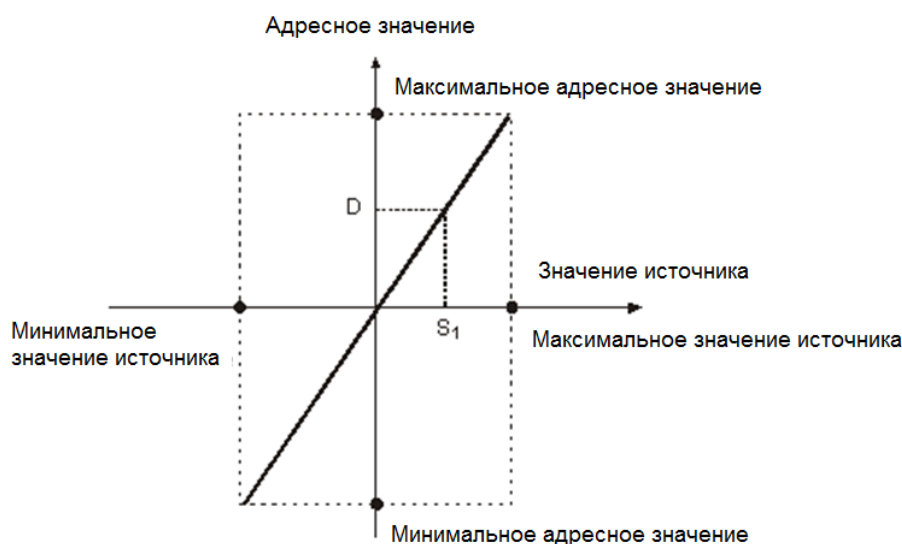
$x$  = Значение источника ( $S_1$ )

$b$  = Смещение = Минимальное адресное значение – Минимальное значение источника  $\times$  Наклон

Приведенные выше параметры заменяются на  $y$ ,  $k$ ,  $x$  и  $b$  в уравнении  $y = kx + b$ , и получается операционная формула инструкции.

$y = kx + b = D = kS_1 + b = \text{Наклон} \times S_1 + \text{Смещение} = \text{Наклон} \times S_1 + \text{Минимальное адресное значение} - \text{Минимальное значение источника} \times \text{Наклон} = \text{Slope} \times (S_1 - \text{Минимальное значение источника}) + \text{Минимальное адресное значение} = (S_1 - \text{Минимальное значение источника}) \times (\text{Максимальное адресное значение} - \text{Минимальное адресное значение}) \div (\text{Максимальное значение источника} - \text{Минимальное значение источника}) + \text{Минимальное адресное значение}.$

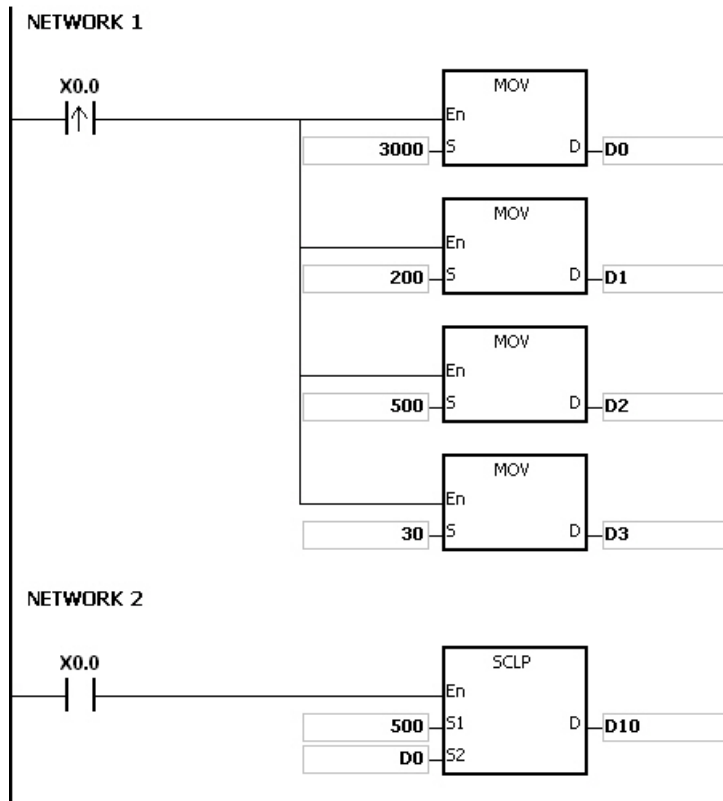
10. Если  $S_1$  больше максимального значения источника, максимальным значением источника будет значение  $S_1$ . Если  $S_1$  меньше минимального значения источника, минимальным значением источника будет значение  $S_1$ . После того, как значения и параметры заданы, выходная кривая выглядит так, как показано ниже.



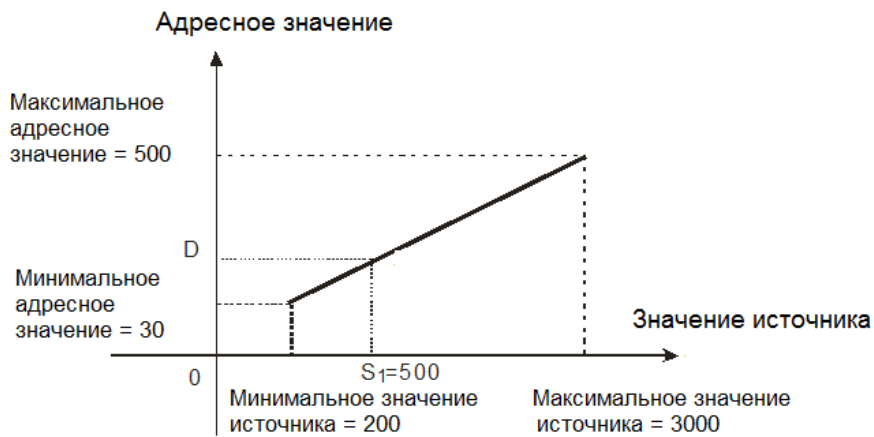
#### Пример 1:

- Предположим, что значение в  $S_1$  равно 500, максимальное значение источника в  $D_0$  равно 3000, минимальное значение источника в  $D_1$  равно 200, максимальное адресное значение в  $D_2$  равно 500, а минимальное адресное значение в  $D_3$  равно 30. Когда  $X0.0$  включен, выполняется инструкция  $SCLP$ , а значение масштабирования сохраняется в  $D10$ .
- Операционная формула:  $D10 = [(500 - 200) \times (500 - 30)] \div (3,000 - 200) + 30 = 80.35$   
80.35 округляется до ближайшего целого и будет равно 80. Значение 80 сохраняется в  $D10$ .



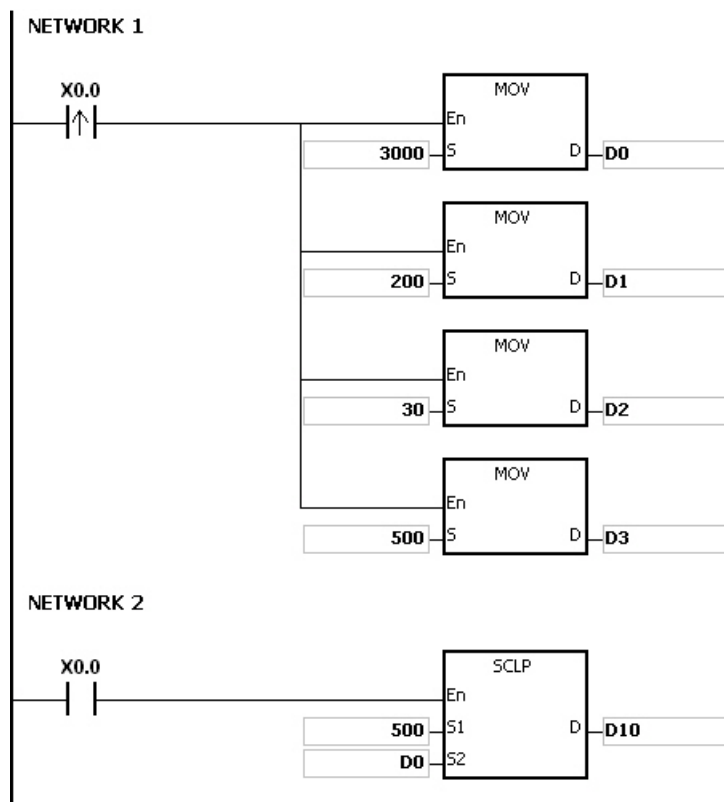


6



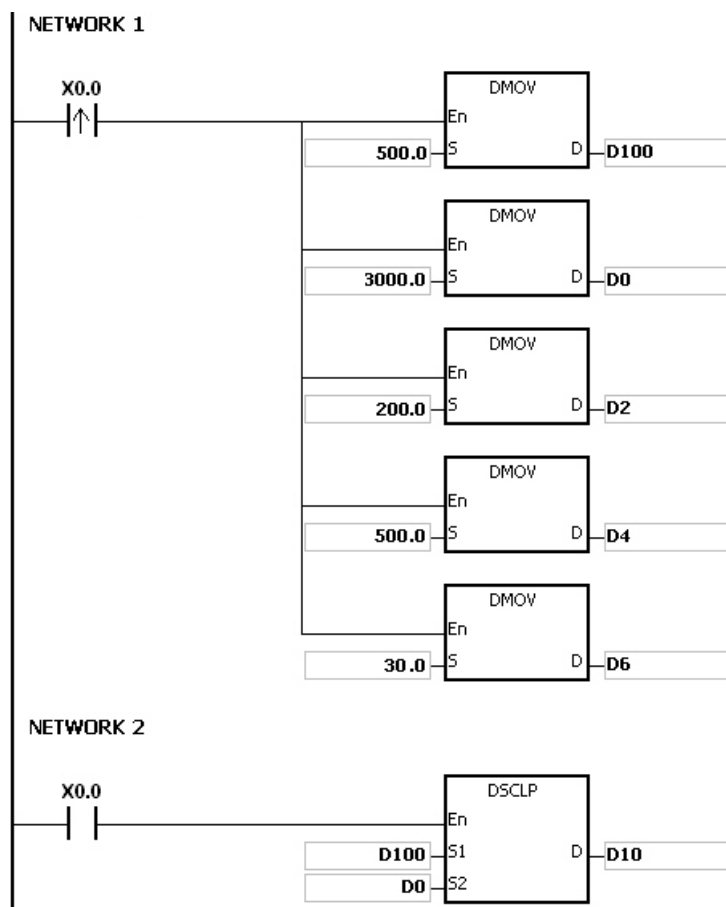
**Пример 2:**

1. Предположим, что значение в S1 равно 500, максимальное значение источника в D0 равно 3000, минимальное значение источника в D1 равно 200, максимальное адресное значение в D2 равно 30, а минимальное адресное значение в D3 равно 500. Когда X0.0 включен, выполняется инструкция SCLP, а значение масштабирования сохраняется в D10.
2. Операционная формула:  $D10 = [(500 - 200) \times (30 - 500)] \div (3,000 - 200) + 500 = 449.64$   
 449.64 округляется до ближайшего целого и будет равно 450. Значение 450 сохраняется в D10.

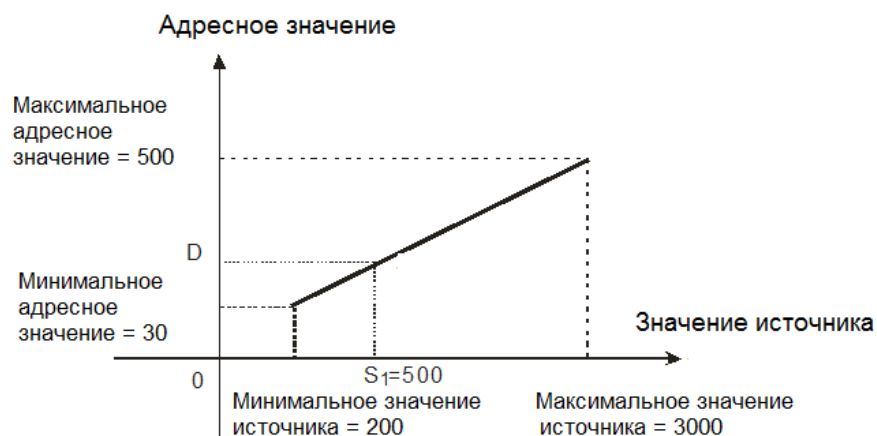


### Пример 3:

- Предположим, что значение в S1 равно 500, максимальное значение источника в D0 равно 3000, минимальное значение источника в D2 равно 200, максимальное адресное значение в D4 равно 500, а минимальное адресное значение в D6 равно 300. Когда X0.0 включен, выполняется инструкция SCLP, а значение масштабирования сохраняется в D10.
- Операционная формула:  $D10 = [(500.0 - 200.0) \times (500.0 - 30.0)] \div (3000.0 - 200.0) + 30.0 = 80.35$   
80.35 округляется до ближайшего целого и будет равно 80. Значение 80.0 сохраняется в D10.



6



**Примечание:**

1. Значение в  $S_1$ , которое используется в 16-битной инструкции, должно находиться в пределах диапазона между минимальным значением источника и максимальным значением источника, то есть между -32 768 и 32 767. Если значение превышает граничное значение, в операции используется граничное значение.
2. Целое значение в  $S_1$ , которое используется в 32-битной инструкции, должно находиться в пределах диапазона между минимальным значением источника и максимальным значением источника, то есть между

-2 147 483 648 и 2 147 483 647. Если целое значение превышает граничное значение, в операции используется граничное значение.

3. Когда используется уравнение наклона, максимальное значение источника должно быть больше минимального значения источника. Однако максимальное значение назначения не обязательно больше минимального адресного значения.
4. Если максимальное значение источника совпадает с минимальным значением источника, инструкция выполняться не будет; включится флаг SMO и в SR0 будет записан код ошибки 16#2012.
5. Если операнд **S<sub>2</sub>** используется в 16-битной инструкции и задается через ПО ISPSOft, тип данных должен быть ARRAY [4] для WORD.
6. Если операнд **S<sub>2</sub>** используется в 32-битной инструкции и задается через ПО ISPSOft, тип данных должен быть ARRAY [4] для DWORD.

API	Код инструкции			Операнды								Функция				
0222	D	SCLM	P	S <sub>1</sub> · S <sub>2</sub> · S <sub>3</sub> · S <sub>4</sub> · D								Операция масштабирования многоточечной области				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S <sub>1</sub>					●	●	●	●								
S <sub>2</sub>					●	●	●	●					○	○		
S <sub>3</sub>					●	●	●	●	●							
S <sub>4</sub>					●	●	●	●	●							
D					●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●		●				
S <sub>2</sub>		●	●		●	●	●						
S <sub>3</sub>		●	●		●	●	●						
S <sub>4</sub>		●	●		●	●	●						
D		●	●		●	●	●		●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**

SCLM	SCLMP
En	En
S1	S1
S2	S2
S3	S3
S4	S4
DSCLM	DSCLMP
En	En
S1	S1
S2	S2
S3	S3
S4	S4

- S<sub>1</sub> : Источник данных
- S<sub>2</sub> : Число многоточечных областей
- S<sub>3</sub> : Сравниваемое значение в многоточечной области
- S<sub>4</sub> : Исходное значение при преобразовании
- D : Операнд результата

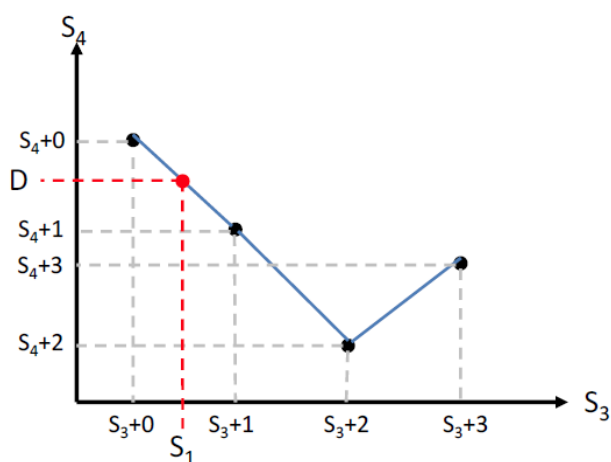
**Описание:**

- 32-битный счетчик может использоваться только 32-битной инструкцией, объекты E такие счетчики не поддерживают. Инструкция работает с версией прошивки V1.04.00 и более поздними для ПЛК AS300.
- В таблице ниже представлены типы данных, соответствующих операндам S<sub>1</sub>, S<sub>3</sub> и S<sub>4</sub> (○ – применимо, X – неприменимо).

Константа	16-битная инструкция	32-битная инструкция	
		SM685 ВКЛ	SM685 ВЫКЛ
K	○	X	○
16#	○	X	○
F	X	○	X

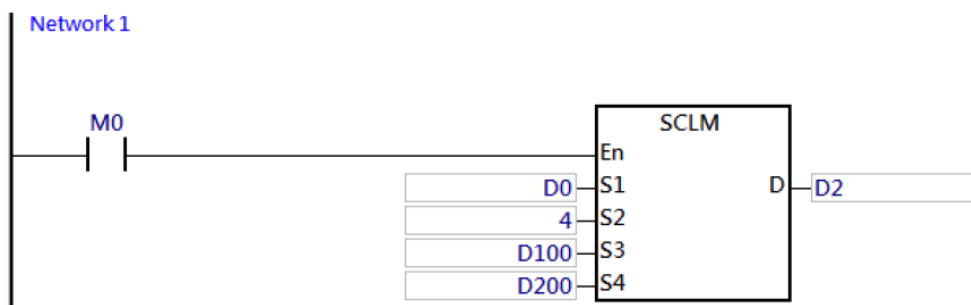
Примечание: При включенном SM685 (работа с числом с плавающей запятой) работает только 32-битная инструкция.

3.  $S_1$  – источник данных.  $S_2$  – число многоточечных областей, значение задается в диапазоне от 2 до 50. Если значение превышает диапазон, инструкция будет выполняться автоматически с минимальным значением или максимальным допустимым значением.  $S_3$  – значение для сравнения в многоточечной области.  $S_4$  является исходным значением преобразования, например, при числе многоточечных областей  $S_2$  равным 10,  $S_3 \sim S_3 + 9$  являются сравнительными значениями в 10 областях.  $S_4, \sim S_4 + 9$ , соответствующие 10 исходных значений преобразования.
4. Порядок сравнения для многоточечных областей определяется как 0, 1, 2 ...  $S_2 - 1$ . Условие сравнения:  $S_1 \geq S_3 + 0$  и  $S_1 < S_3 + 1$ . Если значение  $S_1$  не относится к данной области, сравнение будет сдвигаться к следующей области. Например,  $S_1 \geq S_3 + 1$  и  $S_1 < S_3 + 2$ , сравнение продолжается до тех пор, пока количество сравнений не достигнет  $S_2 - 1$ .
5. Инструкция сравнивает области от меньшей к большей, поэтому значения областей сравнения  $S_3$  необходимо конфигурировать также от меньших к большим.
6. См. рис. ниже как иллюстрацию преобразования значений многоточечной области. (заданное количество областей  $S_2 = 4$ ).



7. Если значение  $S_1$  находится в диапазоне между  $S_3 + 0$  и  $S_3 + 1$ , формула преобразования:  $D = ((S_1 - S_3 + 0) \times (S_4 + 1 - S_4 + 0) / (S_3 + 1 - S_3 + 0)) + S_4 + 0$ .
8. Если значение  $S_1$  не относится к какой-либо определенной области, расчет результата  $D$  показан ниже.  
 Если значение  $S_1 >$  последней указанной области,  $D$  будет сохранять последнее преобразование исходного значения  $S_4$ , например, если значение  $S_1 > S_3 + 3$  значение  $D = S_4 + 3$ .  
 Если значение  $S_1 <$  первой указанной области,  $D$  будет сохранять первое преобразование исходного значения  $S_4$ , например, если значение  $S_1 < S_3 + 0$  значение  $D = S_4 + 0$ .
9. Если  $S_3$  и  $S_4$  для 16-битной инструкции задаются в ПО ISPSOft, тип данных будет ARRAY [ $S_2$ ] типа WORD.
10. Если  $S_3$  и  $S_4$  для 32- битной инструкции задаются в ПО ISPSOft, тип данных будет ARRAY [ $S_2$ ] типа DWORD (SM685 выключен) или ARRAY [ $S_2$ ] типа REAL (SM685 включен).

Пример:



Значения сравнения  $S_3$  для многоточечной области приведены ниже:

Объект	D100	D101	D102	D103
Содержимое	100	200	300	400

Соответствующие исходные значения преобразования  $S_4$  представлены ниже:

Объект	D200	D201	D202	D203
Содержимое	4000	3000	1500	2000

Описание значений D2, полученных путем преобразования данных из источника D0.

Задание  $D0=10$ ,

Т.к.  $D0 < D100$  (в первой области),  $D2=D200=4000$  (первое преобразование исходного значения)

Задание  $D0=K150$ ,

Значение D0 находится в диапазоне (D100, D101) = (100, 200) и соответствующее исходное значение (D200, D201) = (4000, 3000)

Следовательно,

$$D2 = (150-100) * (3000-4000) / (200-100) + 4000 = 3500$$

Задание  $D0=450$

Т.к.  $D0 > D103$  (в последней области),  $D2=D203=2000$  (последнее преобразование исходного значения)

Задание  $D0=K250$

Значение D0 находится в диапазоне (D101, D102) = (200, 300) и соответствующее исходное значение (D201, D202) = (3000, 1500).

Следовательно,

$$D2 = (250-200) * (1500-3000) / (300-200) + 3000 = 2250$$

Задание  $D0=K350$

Значение D0 находится в диапазоне (D102, D103) = (300, 400) и соответствующее исходное значение (D202, D203) = (1500, 2000)

Следовательно,

$$D2 = (350-300) * (2000-1500) / (400-300) + 1500 = 1750$$

## 6.4 Инструкции передачи данных

### 6.4.1 Описание инструкций передачи данных

API	Код инструкции			Операнды								Функция				
0300	D	MOV	P	S · D								Передача данных				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○	○	○	○		○
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●		●		●	●	
D		●	●		●	●	●		●		●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символ:

MOV	MOVP
En	En
S	S
	D
	D
DMOV	DMOVP
En	En
S	S
	D
	D

**S** : Источник данных  
**D** : Операнд сохранения переданных данных

Описание:

1. Когда инструкция выполняется, данные из **S** передаются в **D**.
2. Если данные в **S** являются числом с плавающей запятой, должна применяться только 32-битная инструкция.
3. 32-битный счетчик может использовать только 32-битная инструкция DMOV, регистр E использовать не может.

Пример:

1. Инструкция MOV используется для передачи 16-битных данных.
  - Когда X0.0 отключен, данные в D0 не изменяются. Когда X0.0 включен, значение 10 передается в регистр данных D0.
  - Когда X0.1 отключен, данные в D10 не изменяются. Когда X0.1 включен, текущее значение T0 передается в регистр данных D10.



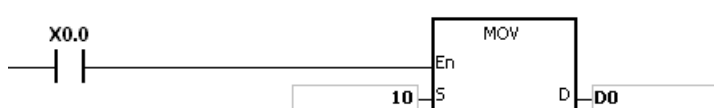
2. Инструкция DMOV используется для передачи 32-битных данных.

- Когда X0.2 отключен, данные в (D31, D30) и (D41, D40) не изменяются. Когда X0.2 включен, текущее значение в (D21, D20) передается в (D31, D30), текущее значение HC0 передается в (D41, D40).

3. Инструкция DMOV используется для передачи числа с плавающей запятой.

- Когда X0.3 отключен, данные в (D51, D50) не изменяются. Когда X0.3 включен, число с плавающей запятой 3.450 преобразуется в двоичное число с плавающей запятой, а затем результат преобразования передается в (D51, D50).

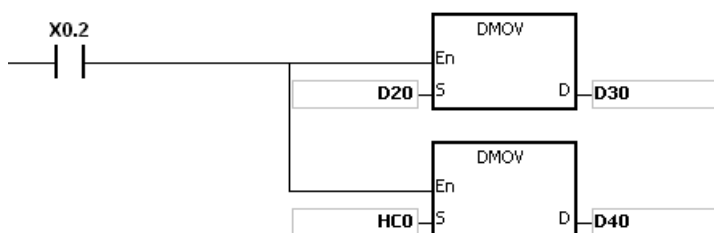
NETWORK 1



NETWORK 2



NETWORK 3



NETWORK 4



6

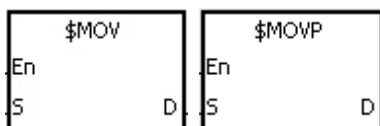
API	Код инструкции			Операнды							Функция						
0302		\$MOV	P	S · D							Передача строковых переменных						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●						○	
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S													●
D													●

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



**S** : Источник данных  
**D** : Операнд сохранения переданных данных

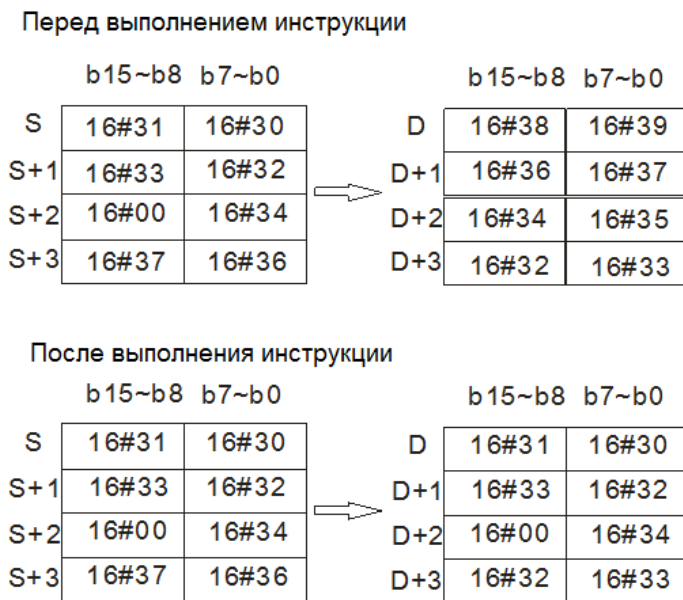
**Описание:**

1. Если в **S** находятся строковые данные и инструкция выполняется, строковые переменные передаются в **D**, и код окончания 16#00 добавляется в конец данных.
2. Если данные в операнде **S** данные не являются строковыми, в конец переданных данных должен быть добавлен код 16#00.
3. Если код 16#00 не найден в **S** в составе 256 символов в строке или выходит за пределы диапазона, инструкция не выполняется; включается флаг SM0 и в SR0 записывается код ошибки 16#200E.
4. Если операнд **S** не является строковым, а инструкция выполняется, строковые данные начиная с **S** (включая код 16#00) переносятся в **D**.
5. Если размер операнда **D** недостаточен, чтобы принять строковые данные из **S**, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
6. Если данные в **S** не являются строковыми, а инструкция выполняется, если первым символом является код 16#00, 16#00 продолжает передаваться в **D**.

7. Когда код 16#00 содержится в младшем байте, выполнение инструкции представлено ниже.



8. Когда код 16#00 содержится в старшем байте, выполнение инструкции представлено ниже.



9. Когда **S** перекрывает **D** и номер операнда **S** меньше номера операнда **D**, передача данных в **D** начинается с конечного кода 16#00.

Перед выполнением инструкции

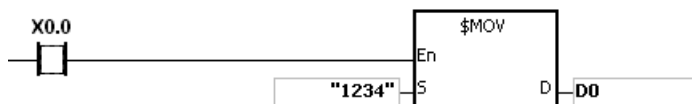
b15~b8		b7~b0	b15~b8		b7~b0
D0	16#31	16#30	D1	16#33	16#32
D1	16#33	16#32	D2	16#35	16#34
D2	16#35	16#34	D3	16#30	16#00
D3	16#30	16#00	D4	16#38	16#37

После выполнения инструкции

b15~b8		b7~b0	b15~b8		b7~b0
D0	16#31	16#30	D1	16#31	16#30
D1	16#33	16#32	D2	16#33	16#32
D2	16#35	16#34	D3	16#35	16#34
D3	16#30	16#00	D4	16#00	16#00

**Пример 1:**

Например, значение в **S** равно "1234". Когда включен соответствующий контакт X0.0, данные в D0~D3 показаны ниже.



Операнд **S**:

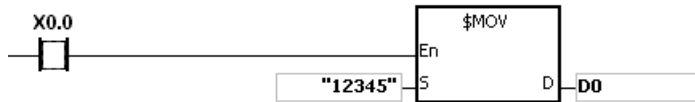
Строковая переменная	'1'	'2'	'3'	'4'
Шестнадцатеричное значение	16#31	16#32	16#33	16#34

Данные в **D** после выполнения инструкции.

Объект	Старший байт	Младший байт	Примечание
D0	16#32	16#31	'1'=16#31; '2'=16#32
D1	16#34	16#33	'3'=16#33; '4'=16#34
D2	16#00	16#00	Конечный код 16#00 в младшем байте. 16#00 автоматически добавляется в старший байт.
D3	Не изменяется	Не изменяется	

**Пример 2:**

Например, значение в **S** равно "1234". Когда включен соответствующий контакт X0.0, данные в D0~D3 показаны ниже.



Операнд **S**:

Строковая переменная	'1'	'2'	'3'	'4'	'5'
Шестнадцатеричное значение	16#31	16#32	16#33	16#34	16#35

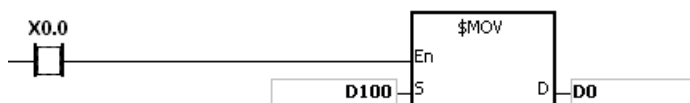
Данные в **D** после выполнения инструкции.

Объект	Старший байт	Младший байт	Примечание
D0	16#32	16#31	'1'=16#31; '2'=16#32
D1	16#34	16#33	'3'=16#33; '4'=16#34
D2	16#00	16#35	Конечный код 16#00 в старшем байте.
D3	Не изменяется	Не изменяется	

6

**Пример 3:**

Когда данные в **S** не являются строковыми, а код 16#00 находится в младшем байте, выполнение инструкции показано ниже.



Операнд **S**:

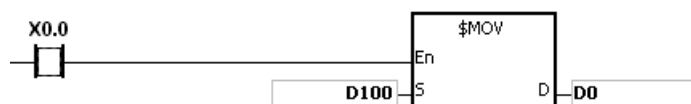
Объект	Старший байт	Младший байт	Примечание
D100	16#31	16#30	'1'=16#31; '0'=16#30
D101	16#33	16#32	'3'=16#33; '2'=16#32
D102	16#35	16#34	'5'=16#35; '4'=16#34
D103	16#30	16#00	'0'=16#30; 16#00 – конечный код.

Данные в **D** после выполнения инструкции.

Объект	Старший байт	Младший байт	Примечание
D0	16#31	16#30	'1'=16#31; '0'=16#30
D1	16#33	16#32	'3'=16#33; '2'=16#32
D2	16#35	16#34	'5'=16#35; '4'=16#34
D3	16#00	16#00	Конечный код 16#00 в младшем байте. 16#00 автоматически добавляется в старший байт.
D4	Не изменяется	Не изменяется	

#### Пример 4:

Когда данные в **S** не являются строковыми, а код 16#00 находится в старшем байте, выполнение инструкции показано ниже.



Операнд **S**:

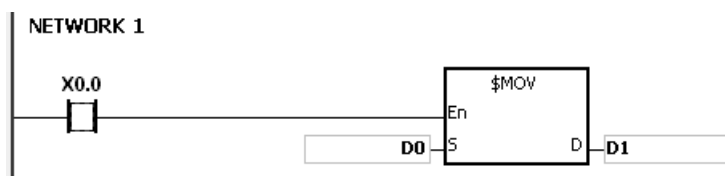
Объект	Старший байт	Младший байт	Примечание
D100	16#31	16#30	'1'=16#31; '0'=16#30
D101	16#33	16#32	'3'=16#33; '2'=16#32
D102	16#00	16#34	16#00 – конечный код. '4'=16#34
D103	16#37	16#36	'7'=16#37; '6'=16#36

Данные в **D** после выполнения инструкции.

Объект	Старший байт	Младший байт	Примечание
D0	16#31	16#30	'1'=16#31; '0'=16#30
D1	16#33	16#32	'3'=16#33; '2'=16#32
D2	16#00	16#34	16#00 – конечный код. '4'=16#34
D3	Не изменяется	Не изменяется	

**Пример 5:**

Когда **S** перекрывает **D** и номер операнда **S** меньше номера операнда **D**, передача данных в **D** начинается с конечного кода 16#00.



Операнд **S**:

Объект	Старший байт	Младший байт	Примечание
D0	16#31	16#30	'1'=16#31; '0'=16#30
D1	16#33	16#32	'3'=16#33; '2'=16#32
D2	16#35	16#34	'5'=16#35; '4'=16#34
D3	16#30	16#00	'0'=16#30; 16#00 – конечный код.
D4	16#38	16#37	'8'=16#38; '7'=16#37

**6**

Данные в **D** после выполнения инструкции.

Объект	Старший байт	Младший байт	Примечание
D1	16#31	16#30	'1'=16#31; '0'=16#30
D2	16#33	16#32	'3'=16#33; '2'=16#32
D3	16#35	16#34	'5'=16#35; '4'=16#34
D4	16#00	16#00	Конечный код 16#00 в младшем байте. 16#00 автоматически добавляется в старший байт.
D5	Не изменяется	Не изменяется	

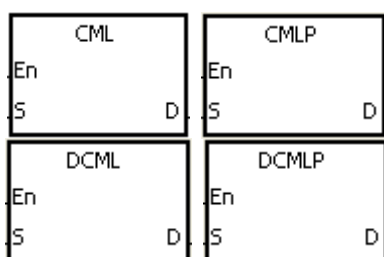
API	Код инструкции			Операнды							Функция						
0303	D	CML	P	S · D							Инвертирование данных						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



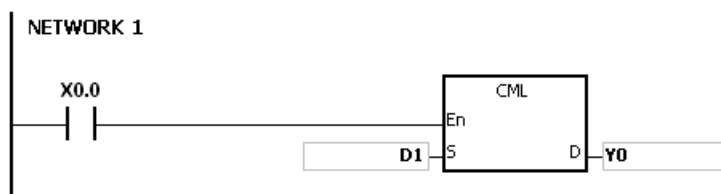
**S** : Источник данных  
**D** : Операнд сохранения результата

**Описание:**

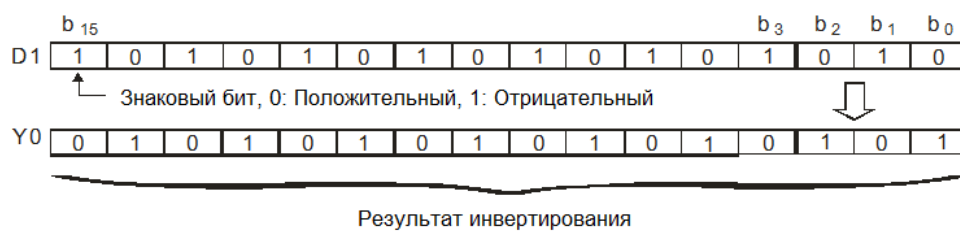
1. Инструкция используется для инвертирования битов в **S**, т.е. 1 на 0 и 0 на 1. Результат инвертирования сохраняется в **D**. Если данные в **S** являются константой, константа преобразуется в двоичное значение.
2. 32-битный счетчик может использовать только 32-битная инструкция, регистр E использовать такой счетчик не может.

**Пример 1:**

Когда X0.0 включен, все биты в D1 инвертируются и результат сохраняется в Y0.0~Y0.15.

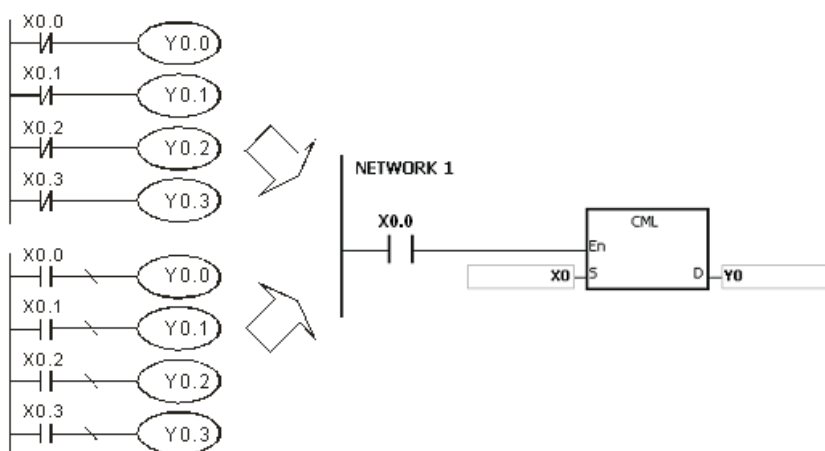






**Пример 2:**

Нижеприведенная операция может быть реализована с помощью инструкции CML.



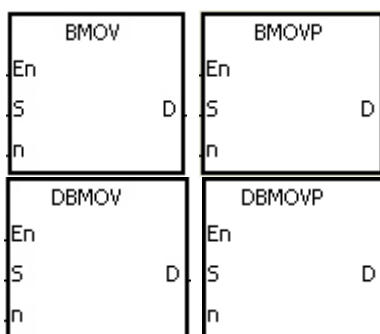
API	Код инструкции			Операнды							Функция						
0304	D	BMOV	P	S · D · n							Передача всех данных						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●							
D		●			●	●	●	●								
n	●	●			●	●	●	●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**

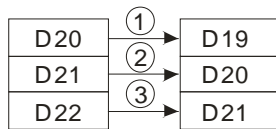


- S** : Источник данных
- D** : Операнд назначения
- n** : Длина данных

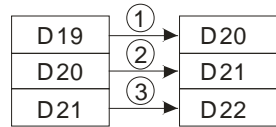
**Описание:**

1. n частей данных из операндов, начиная с S передаются в операнды, начиная с D.
2. Значение операнда n должно находиться в диапазоне от 1 до 256.
3. 32-битный счетчик может использовать только 32-битная инструкция, регистр E использовать такой счетчик не может.
4. Чтобы предотвратить ошибку, возникающую из-за перекрытия между операндом источника и операндом назначения, данные передаются следующим образом (с использованием 16-битной инструкции в качестве примера):

Если номер операнда S больше номера операнда D, данные передаются по порядку от ① до ③.

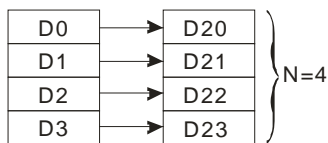
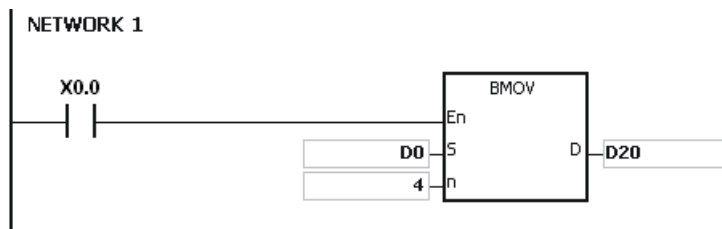


Если номер операнда **S** меньше номера операнда **D**, данные передаются в обратном порядке от ③ to ①.



**Пример 1:**

Когда X0.0 включен, данные из D0~D3 передаются в D20~D23.

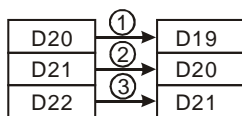
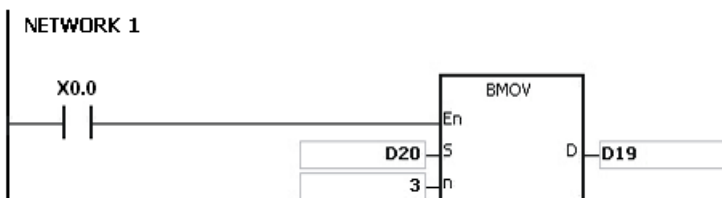


6

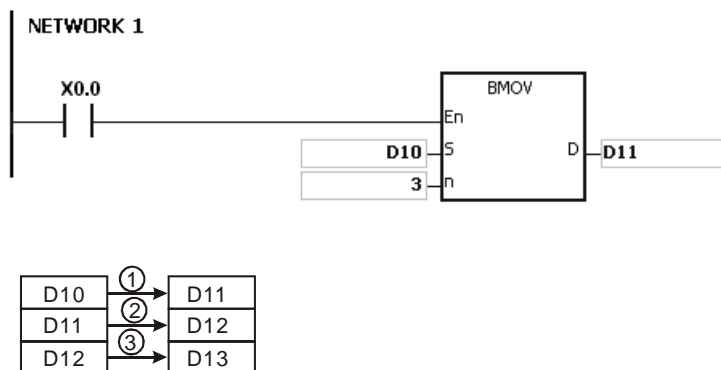
**Пример 2:**

Чтобы предотвратить ошибку, возникающую из-за перекрытия между исходными регистрами и целевыми регистрами, данные передаются следующим образом.

1. Если номер операнда **S** больше номера операнда **D**, данные передаются по порядку от ① до ③.



2. Если номер операнда **S** меньше номера операнда **D**, данные передаются в обратном порядке от ③ to ①.



**Примечание:**

1. Если значение **D+n-1** выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение **S+n-1** выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
3. Если значение **n** выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

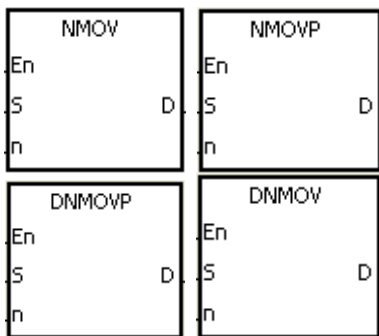
API	Код инструкции			Операнды								Функция				
0305	D	NMOV	P	S · D · n								Перенос данных на несколько операндов				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●				○	○		
D		●			●	●	●	●								
n	●	●			●	●	●	●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

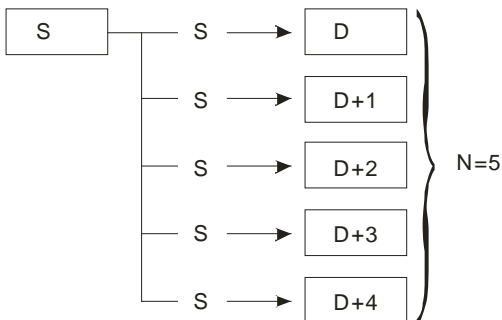
**Символ:**



- S** : Источник данных
- D** : Операнд назначения
- n** : Длина данных

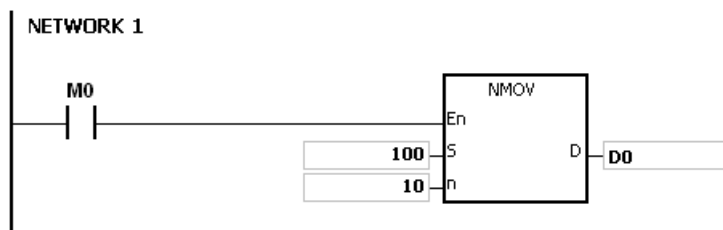
**Описание:**

1. При выполнении инструкции, данные из **S** передаются в **n** операндов, начиная с **D**. Если инструкция не работает, данные в **D** не изменяются.
2. 32-битный счетчик может использовать только 32-битная инструкция.
3. Значение операнда **n** должно находиться в диапазоне от 1 до 256.



**Пример:**

Когда M0 включен, значение 100 передается D0~D9.

**Примечание:**

1. Если значение  $D+n-1$  выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение  $n$  выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

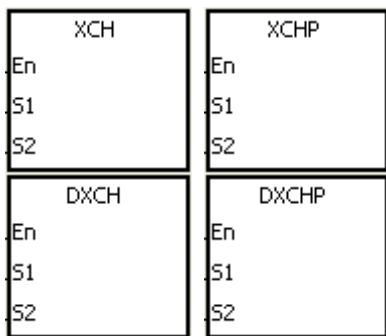
API	Код инструкции			Операнды								Функция				
0306	D	XCH	P	<b>S<sub>1</sub> · S<sub>2</sub></b>								Обмен данными				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>		●			●	●	●	●				○				
S <sub>2</sub>		●			●	●	●	●				○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●						
S <sub>2</sub>		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



S<sub>1</sub> : Данные для обмена

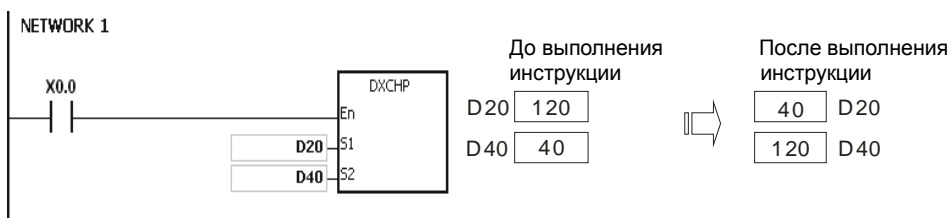
S<sub>2</sub> : Данные для обмена

**Описание:**

1. Данные из операндов, начиная с S<sub>1</sub>, обмениваются с данными из операндов, начиная с S<sub>2</sub>.
2. 32-битный счетчик может использовать только 32-битная инструкция, регистр E использовать такой счетчик не может.

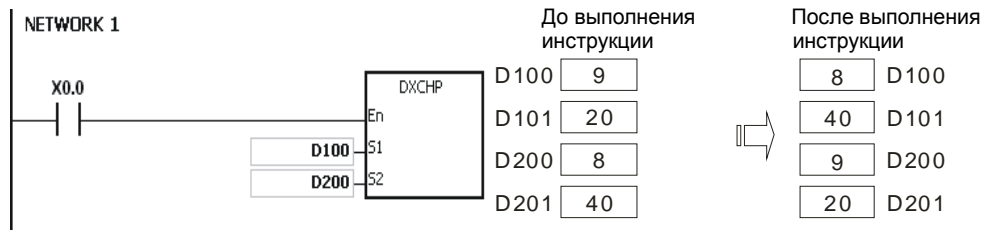
**Пример 1:**

Если X0.0 включен, операнды D20 и D40 обмениваются данными.



**Пример 2:**

Если X0.0 включен, операнды D100 и D200 обмениваются данными.





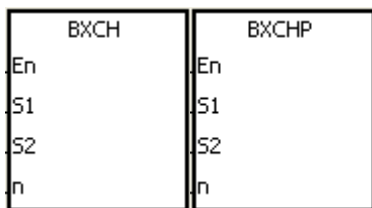
API	Код инструкции			Операнды							Функция				
0307		VXCH	P	$S_1 \cdot S_2 \cdot n$							Обмен всеми данными				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$		●			●	●		●								
$S_2$		●			●	●		●								
$n$	●	●			●	●		●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$n$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

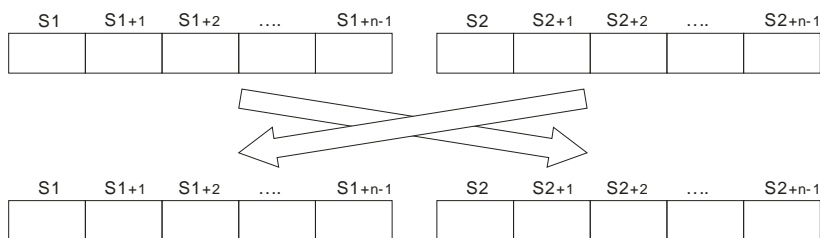
**Символ:**



- $S_1$  : Данные для обмена
- $S_2$  : Данные для обмена
- $n$  : Длина данных

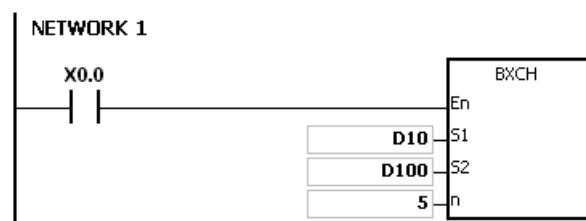
**Описание:**

- Операнды  $S_1 \sim S_{1+n-1}$  обмениваются данными с операндами  $S_2 \sim S_{2+n-1}$ .
- Значение операнда  $n$  должно находиться в диапазоне от 1 до 256.



**Пример:**

Когда X0.0 включен, операнды D10~D14 и D100~D104 обмениваются данными.

**Примечание:**

1. Если  $S_1+n-1$  выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если  $S_2+n-1$  выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
3. Если значение  $n$  выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

API	Код инструкции			Операнды								Функция				
0308	D	SWAP	P	S								Замена старшего байта младшим байтом				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



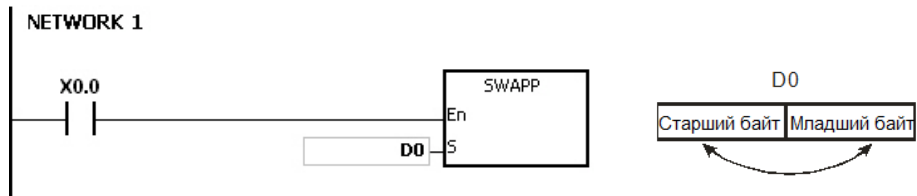
**S** : Источник данных

**Описание:**

1. При выполнении 16-битной инструкции, младший и старший байты в **S** обмениваются данными.
2. При выполнении 32-битной инструкции данные младшего байта старшего слова в **S** обмениваются с данными старшего байта старшего слова **S**, аналогично обмениваются данными младший и старший байты младшего слова в **S**.
3. 32-битный счетчик может использовать только 32-битная инструкция, регистр E использовать такой счетчик не может.

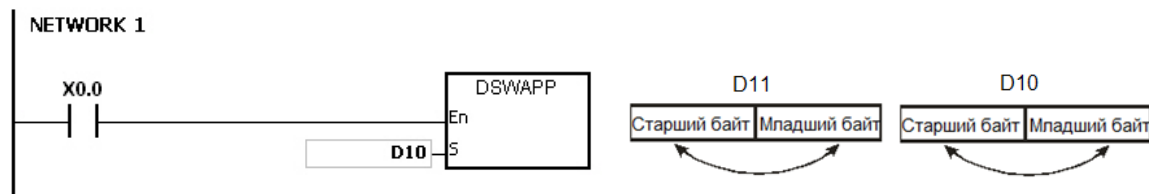
**Пример 1:**

Когда X0.0 включен, данные младшего байта D0 обмениваются с данными старшего байта D0.



**Пример 2:**

Когда X0.0 включен, данные младшего байта D11 обмениваются с данными старшего байта D11, а данные младшего байта D10 обмениваются с данными старшего байта D10.



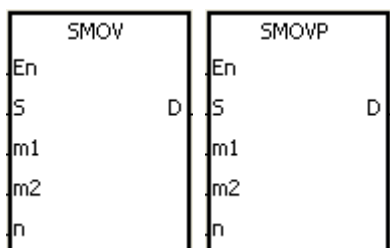
API	Код инструкции			Операнды							Функция					
0309		SMOV	P	<b>S · m<sub>1</sub> · m<sub>2</sub> · D · n</b>							Передача цифр					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S</b>	●	●			●	●		●	●				○	○		
<b>m<sub>1</sub></b>	●	●			●	●		●	●				○	○		
<b>m<sub>2</sub></b>	●	●			●	●		●	●				○	○		
<b>D</b>		●			●	●		●								
<b>n</b>	●	●			●	●		●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●			●	●							
<b>m<sub>1</sub></b>		●			●	●							
<b>m<sub>2</sub></b>		●			●	●							
<b>D</b>		●			●	●							
<b>n</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

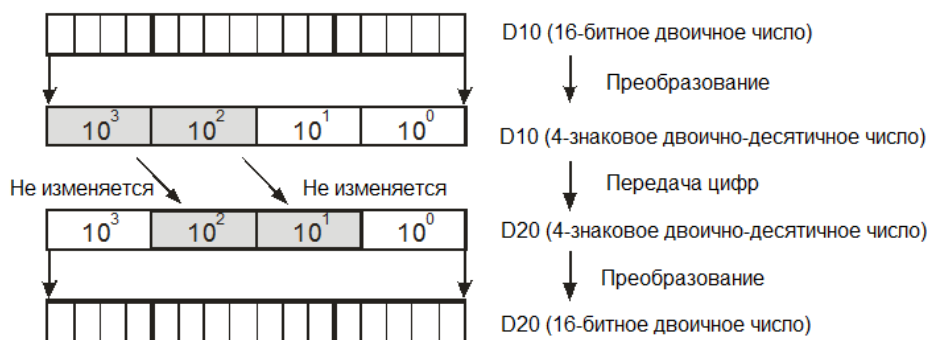
**Символ:**



- S** : Источник данных
- m<sub>1</sub>** : Начальная цифра, которая будет передана из источника данных
- m<sub>2</sub>** : Число цифр для передачи
- D** : Целевой операнд
- n** : Начальная цифра данных, сохраненных в целевом операнде

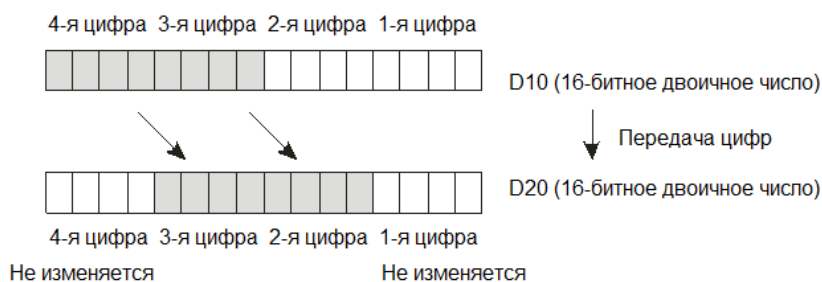
**Описание:**

1. Инструкция может использоваться для распределения и объединения данных. Когда инструкция выполняется, количество цифр **m<sub>2</sub>**, начиная с **m<sub>1</sub>**-й цифры в **S**, переносится в **D**, начиная с **n**-й цифры.
2. Значение операнда **m<sub>1</sub>** должно быть от 1 до 4. Значение операнда **m<sub>2</sub>** должно быть от 1 до **m<sub>1</sub>**. Значение операнда **n** должно быть от **m<sub>2</sub>** до 4 (четыре бита рассматриваются как единица).
3. Когда флаг SM605 отключен, данные, участвующие в выполнении инструкции, представляют собой двоично-десятичные числа.



Предположим, значение в **S** равно K1234, значение в **D** равно K5678. После выполнения инструкции, значение в **S** равно 1234, значение в **D** равно 5128.

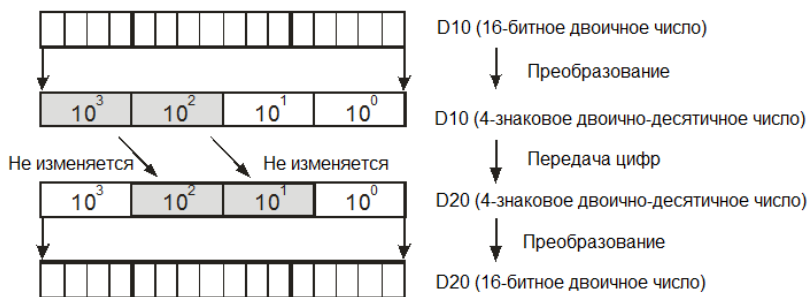
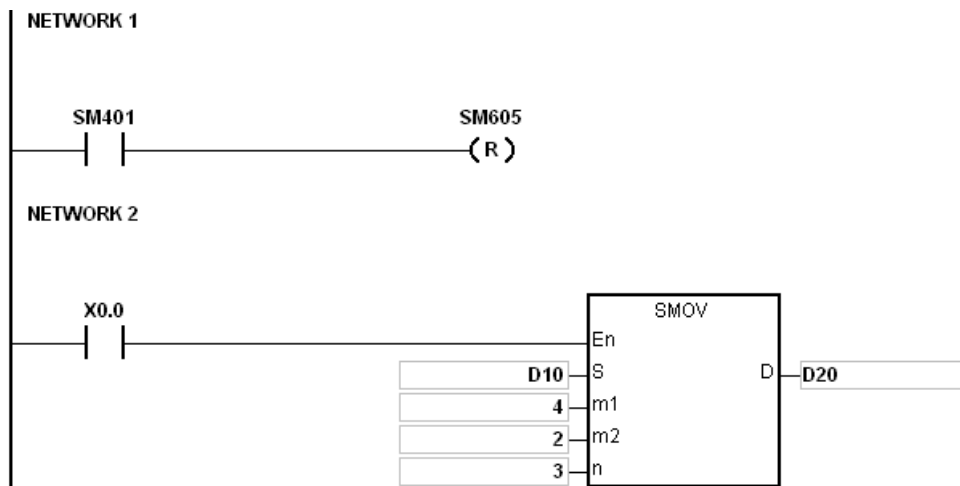
4. Когда флаг SM605 включен, данные, участвующие в выполнении инструкции, представляют собой двоичные числа.



Предположим, значение в **S** равно 16#1234, значение в **D** равно 16#5678. После выполнения инструкции, значение в **S** равно 16#1234, значение в **D** равно 16#5128.

**Пример 1:**

1. Когда SM605 выключен, данные, участвующие в выполнении инструкции, представляют собой двоично-десятичные числа. Когда X0.0 включен, две цифры десятичного числа, начинающиеся с четвертой цифры десятичного числа (тысячные десятичного числа) в D10, переносятся на две цифры десятичного числа, которые начинаются от третьей цифры десятичного числа (сотни десятичного числа) в D20. После выполнения команды тысячи десятичного числа (103) и единицы десятичного числа (100) в D20 не изменяются.
2. Если значение двоично-десятичного числа выходит за пределы диапазона от 0 до 9 999, возникает ошибка, инструкция не выполняется, включается флаг M0 и в SR0 записывается код ошибки 16#200D.

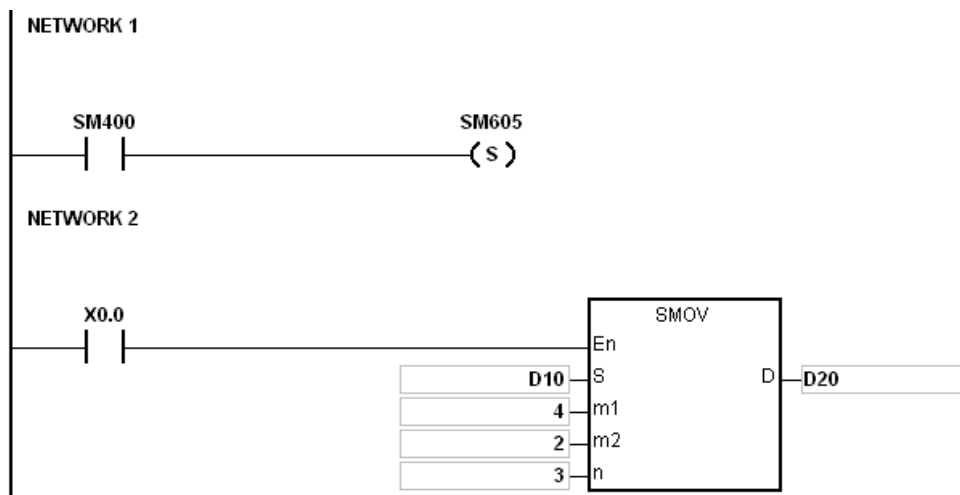


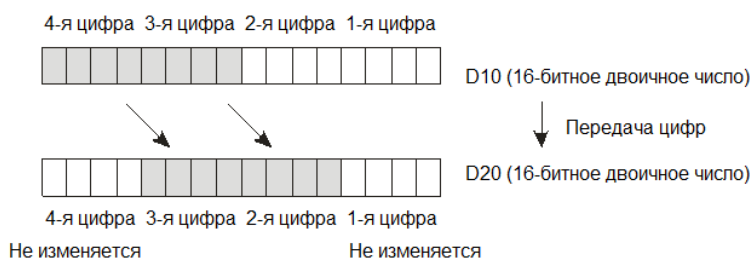
Предположим, значение в S равно 1234, значение в D равно 5678. После выполнения инструкции, значение в S равно 1234, значение в D равно 5128.

6

**Пример 2:**

Когда флаг SM605 включен, данные, участвующие в выполнении инструкции, представляют собой двоичные числа. Когда инструкция SMOV выполняется, двоичные числа в D10 и D20 не преобразуются в двоично-десятичные числа, а передаваемая цифра состоит из четырех бит.

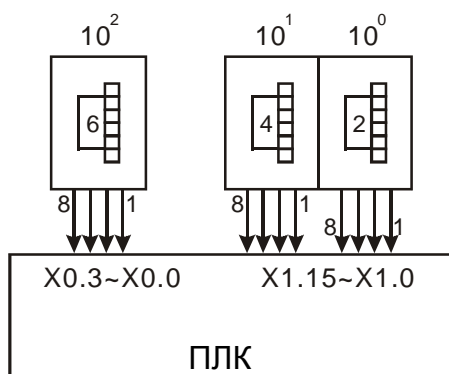




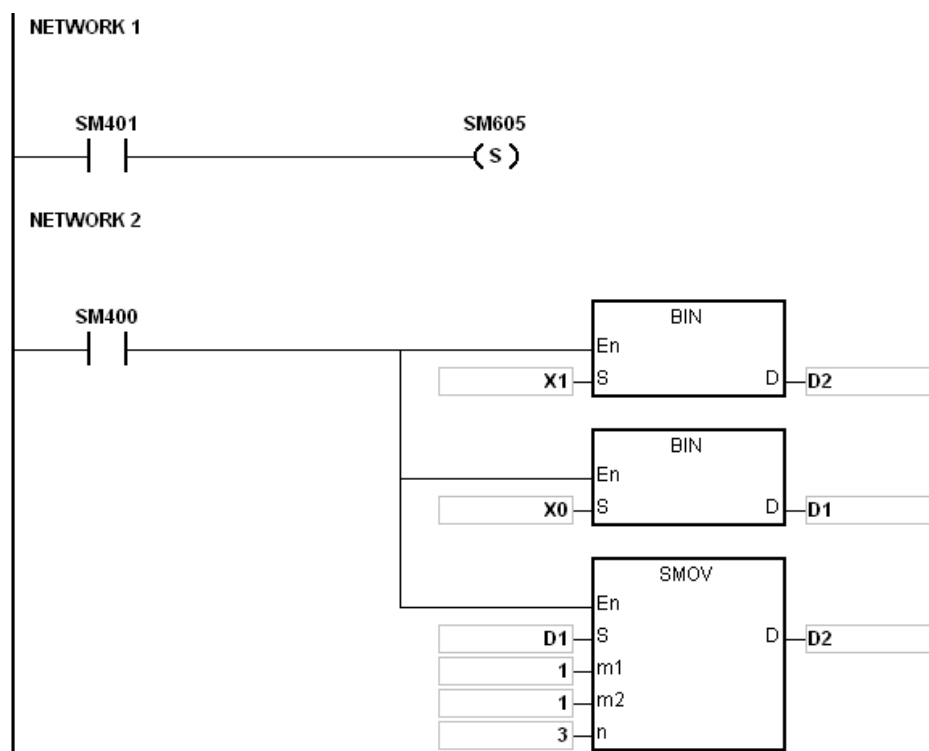
Предположим, значение в S равно 16#1234, значение в D равно 16#5678. После выполнения инструкции, значение в S равно 16#1234, значение в D равно 16#5128.

**Пример 3:**

1. Инструкция может использоваться для объединения значений DIP-переключателей, которые подключены к входным клеммам, номера которых не являются последовательными.
2. Две цифры значения DIP-переключателя справа передаются на две цифры, начинающиеся со второй цифры значения в D2, а одна цифра значения DIP-переключателя слева переносится на первую цифру числа в D1.
3. Инструкция SMOV может использоваться для передачи первой цифры значения в D1 на третью цифру значения в D2. Другими словами, с помощью данной инструкции, два DIP-переключателя могут быть объединены в один DIP-переключатель.







**Примечание:**

1. Предположим, в выполнении инструкции участвуют двоично-десятичные данные. Если число в **S** выходит за пределы диапазона от 0 до 9 999 или число в **D** выходит за пределы диапазона от 0 до 9999, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200D.
2. Если **m<sub>1</sub>** меньше 1 или больше 4, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.
3. Если **m<sub>2</sub>** меньше 1 или больше **m<sub>1</sub>**, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.
4. Если **n** меньше **m<sub>2</sub>** или больше 4, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

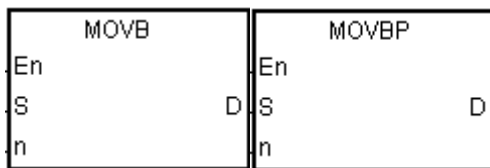
API	Код инструкции			Операнды							Функция					
0310		MOVB	P	S · n · D							Передача нескольких бит					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●	●	●	●	●	●	●		●						
n	●	●			●	●		●	●		○	○	○	○		
D		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												
n		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



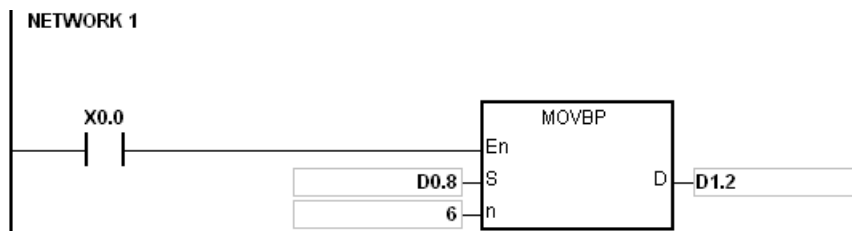
- S** : Источник данных
- n** : Длина данных
- D** : Целевой операнд

**Описание:**

1. Когда инструкция выполняется, **n** единиц данных в операндах начиная с **S**, передается операндам начиная с **D**.
2. Когда операндом **S** являются объекты T, C или HC, передается только состояние объекта, а текущее значение объекта не передается.
3. Значение операнда **n** должно находиться в диапазоне от 1 до 256. Если **n** меньше 1 или больше 256, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

**Пример:**

Когда X0.0 включен, данные в D0.8~D0.13 передаются в D1.2~D1.7.



**Примечание:**

1. Если значение в **D+n-1** превышает допустимый диапазон, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение в **S+n-1** превышает допустимый диапазон, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.

## 6.5 Инструкции перехода

### 6.5.1 Описание инструкций перехода

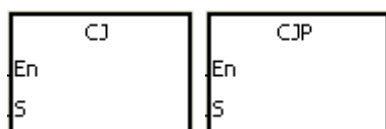
API	Код инструкции			Операнд								Функция				
0400		CJ	P	S								Условный переход				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S																

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

#### Символ:



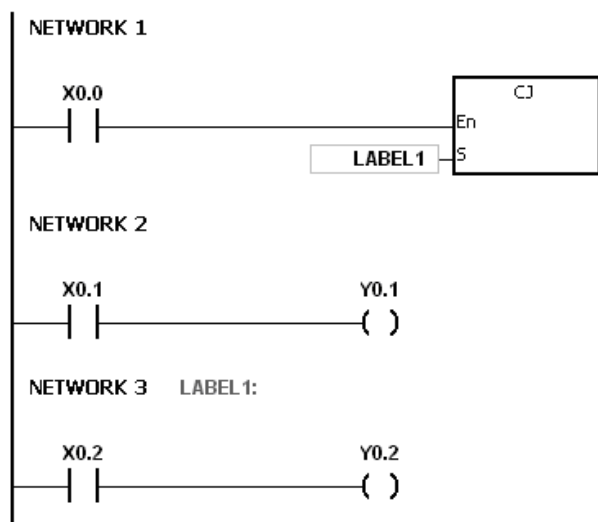
**S** : Целевой операнд перехода

#### Описание:

- Если часть программы в ПЛК не требуется выполнять, можно использовать инструкции CJ или CJP для сокращения времени сканирования. Кроме того, когда используется двойной выход, также можно использовать инструкции CJ или CJP.
- Если метка программы располагается перед инструкцией CJ, произойдет ошибка сторожевого таймера и ПЛК прекратит работу. Пожалуйста, внимательно применяйте данную инструкцию.
- Команда CJ может указывать одну и ту же метку несколько раз.
- Когда инструкция выполняется, действия объектов следующие:
  - Состояние Y, M и S остается таким же, как и до перехода.
  - Таймер продолжает отсчет и при достижении заданного значения катушка T будет активна.
  - Для дополнительной информации по инструкциям MC и MCR см. пример 2 ниже.
  - Основные прикладные инструкции выполняться не будут.

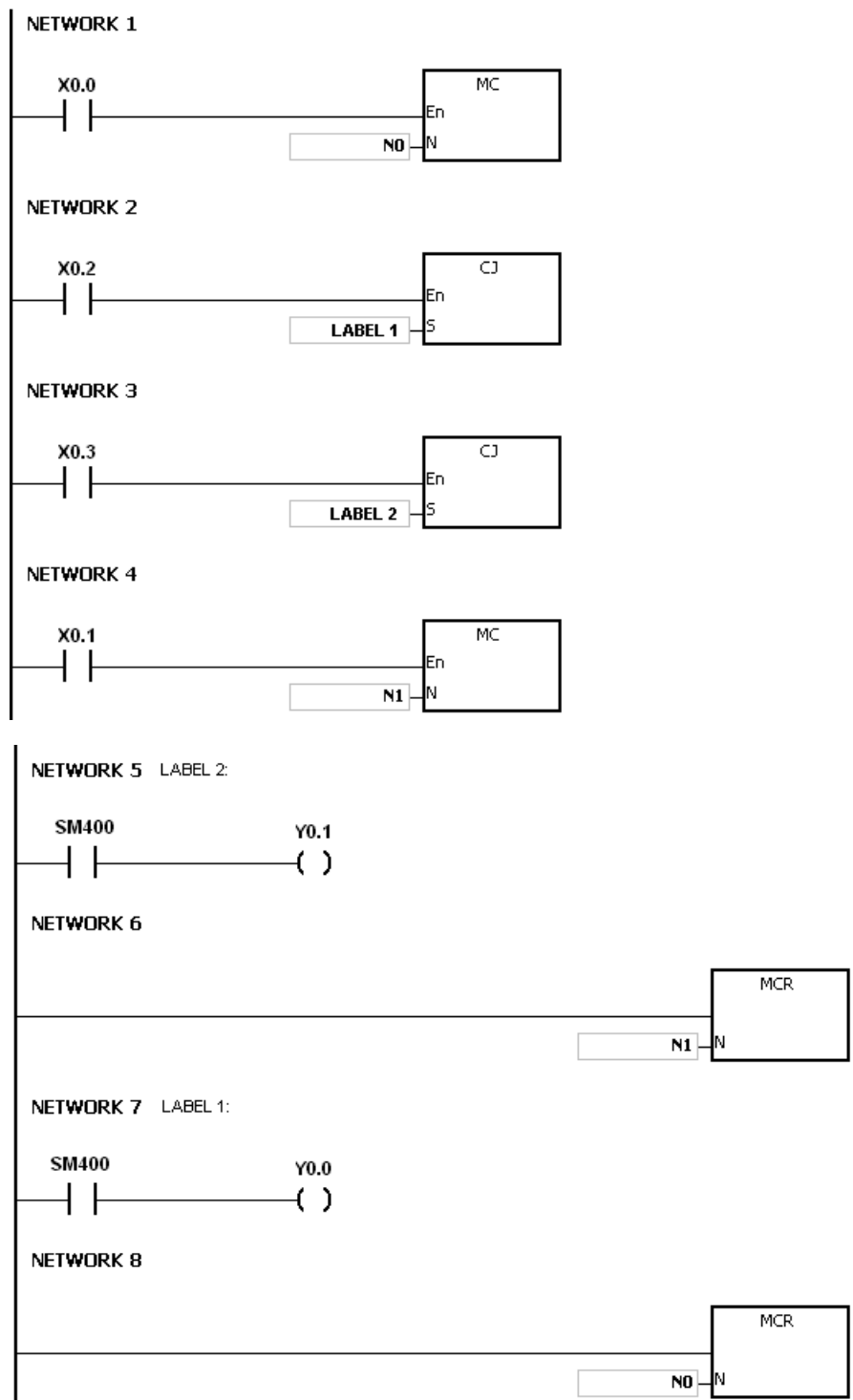
**Пример 1:**

1. Когда X0.0 включен, происходит переход от NETWORK1 к LABEL1 (NETWORK3), а NETWORK2 пропускается.
2. Когда X0.0 выключен, выполнение программы начинается с NETWORK1 до NETWORK3 последовательно, а инструкция CJ не выполняется.



**Пример 2:**

1. Инструкция CJ между инструкцией MC и инструкцией MCR может быть использована в пяти случаях:
  - (a) Выполнение перехода из-за пределов одного цикла MC/MCR за пределы другого цикла MC/MCR.
  - (b) Выполнение перехода из-за пределов цикла MC/MCR внутрь цикла MC/MCR.
  - (c) Выполнение перехода из пределов цикла MC/MCR внутрь цикла MC/MCR.
  - (d) Выполнение перехода из пределов цикла MC/MCR за пределы цикла MC/MCR.
  - (e) Выполнение перехода из пределов одного цикла MC/MCR внутрь другого цикла MC/MCR.
2. После выполнения команды MC, предыдущее состояние контакта переключателя помещается в верхнюю часть стека внутри ПЛК. Стек управляется ПЛК и не может быть изменен пользователем. После выполнения инструкции MCR, предыдущее состояние контакта переключателя удаляется из верхней части стека. При условиях, перечисленных в подпунктах (b), (d) и (e), количество раз, когда элементы помещаются в стек, может отличаться от количества удалений элементов из стека. Когда возникает такая ситуация, в стек может быть добавлено не более 32 элементов, а удаляться из стека элементы могут быть до тех пор, пока стек не станет пустым. Поэтому, когда инструкции CJ или CJP используются с инструкциями MC и MCR, необходимо с максимальной осторожностью следить за добавлением и удалением элементов в стеке.

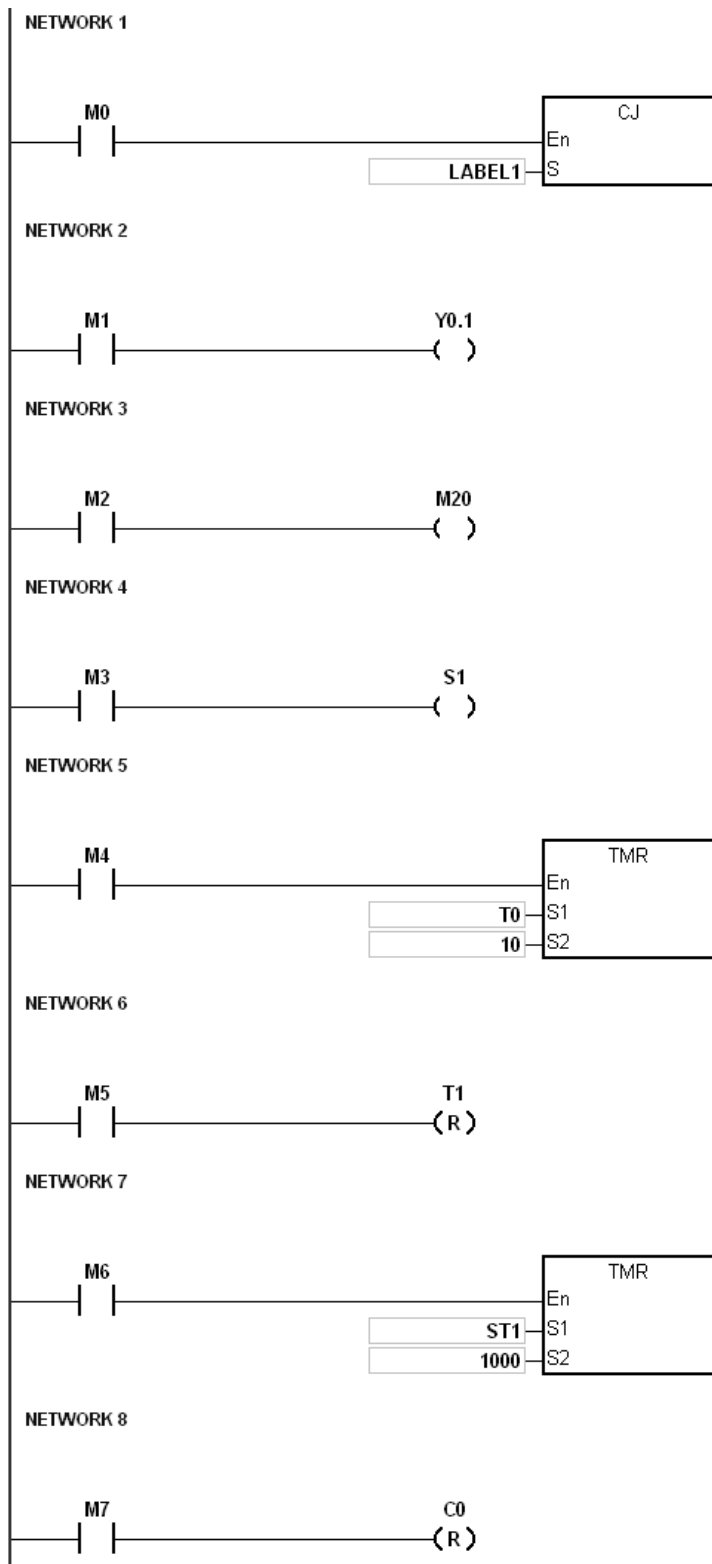


**Пример 3:**

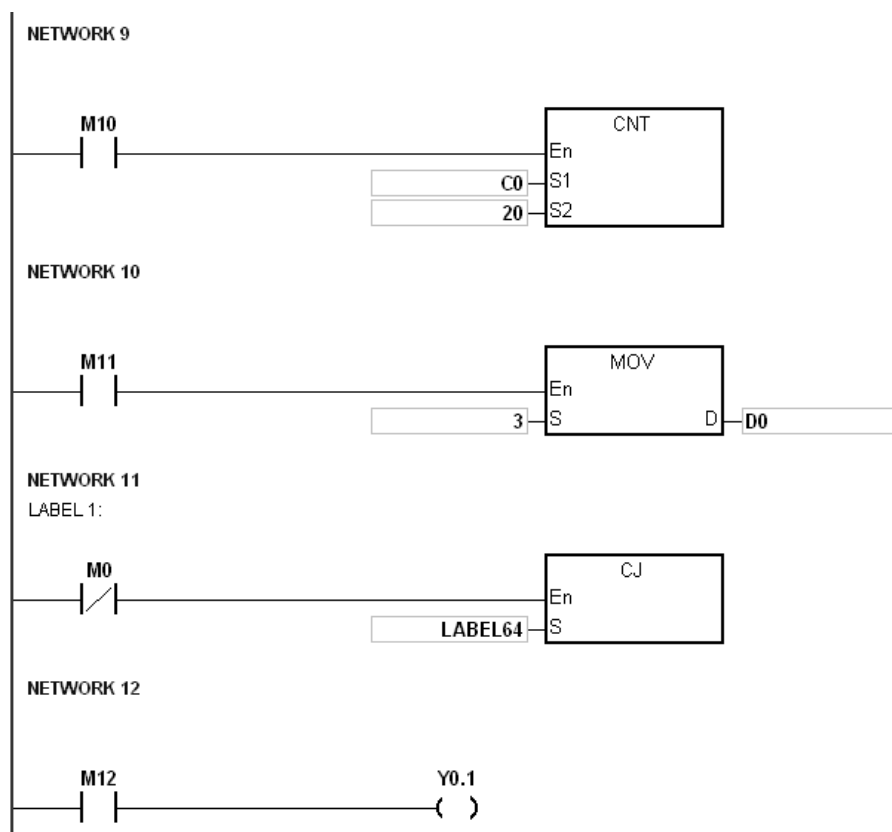
Состояния различных объектов представлены ниже.

Объект	Состояние контактов до выполнения инструкции CJ M0 ВЫКЛ	Состояние контактов при выполнении инструкции CJ M0 ВКЛ	Состояние выходных катушек при выполнении инструкции CJ M0 ВКЛ
Y, M и S	M1, M2 и M3 выключены.	M1, M2 и M3 включаются.	Y0.1 <sup>*1</sup> , M20 и S1 выключены.
	M1, M2 и M3 включены.	M1, M2 and M3 выключаются.	Y0.1 <sup>*1</sup> , M20 и S1 включены.
Таймер	M4 выключен.	M4 включается.	Таймер не включен.
	M4 включен.	M4 выключается	Таймер продолжает отсчет, и когда достигнуто заданное время, будет активирована выходная катушка T.
Аккумулятивный таймер	M6 выключен.	M6 включается.	ST1 не включен.
	M6 включен.	M6 выключается.	Аккумулятивный таймер продолжает отсчет, и когда достигнуто заданное время, будет активирована выходная катушка T.
Счетчик	M7 и M10 выключены.	M10 включен/выключен.	Счетчик не включен.
	M7 выключен OFF. M10 включен/выключен.	M10 включен/выключен.	C0 останавливает отсчет. Когда M0 выключается, C0 сохраняет отсчет.
Прикладная инструкция	M11 выключен.	M11 включается	Прикладная инструкция не выполняется.
	M11 включен.	M11 выключается	Пропущенная прикладная инструкция не выполняется.

\*1:Y0.1 – двойной выход. Когда M0 выключен, Y0.1 управляется M1. Когда M0 включен, Y0.1 управляется M12.







**Примечание:**

1. См. Руководство по работе с ПО ISPSOft для дополнительной информации по меткам (Указателям).

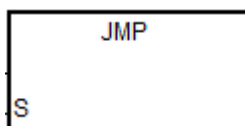
API	Код инструкции			Операнд								Функция					
0401		JMP		S								Безусловный переход					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S																

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



S : Целевой операнд

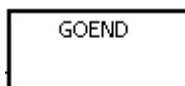
Описание:

1. Процесс выполнения программы переходит к той части программы, которая указана меткой (Указателем) без каких-либо условий.
2. Если программа, указанная меткой, находится перед инструкцией JMP, произойдет ошибка сторожевого таймера, и ПЛК прекратит работу. Пожалуйста, внимательно используйте инструкцию.
3. См. описание инструкции CJ для получения дополнительной информации о состояниях объектов при выполнении инструкции.
4. См. Руководство по работе с ПО ISPSOFT для дополнительной информации по меткам (Указателям).

API	Код инструкции		Операнд	Функция
0402		GOEND	—	Переход к END

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



Описание:

1. Когда условие выполнено, программа переходит к инструкции END.
2. Функциональные блоки и задачи прерывания не поддерживают данную инструкцию. Кроме того, инструкция не может находиться в программе между инструкцией FOR и инструкцией NEXT.
3. Когда выполняется команда GOEND, пропущенные инструкции не выполняются, данные во всех объектах не изменяются, состояние всех регистров также не изменяется.

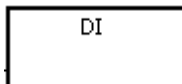
## 6.6 Исполняемые программой инструкции

### 6.6.1 Описание исполняемых программой инструкций

API	Код инструкции		Операнд	Функция
0500		DI	-	Запрет прерываний

Импульсная	16-битная	32-битная
-	AS	-

Символьное обозначение:

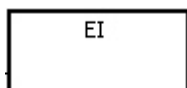


Описание:

1. Подробнее см. описание инструкции EI в параграфе, посвященном API0501.

API	Код инструкции		Операнд	Функция
0501		EI	-	Разрешение прерываний
		Импульсная инструкция	16-битная инструкция	32-битная инструкция
		-	AS	-

Символьное обозначение:



Описание:

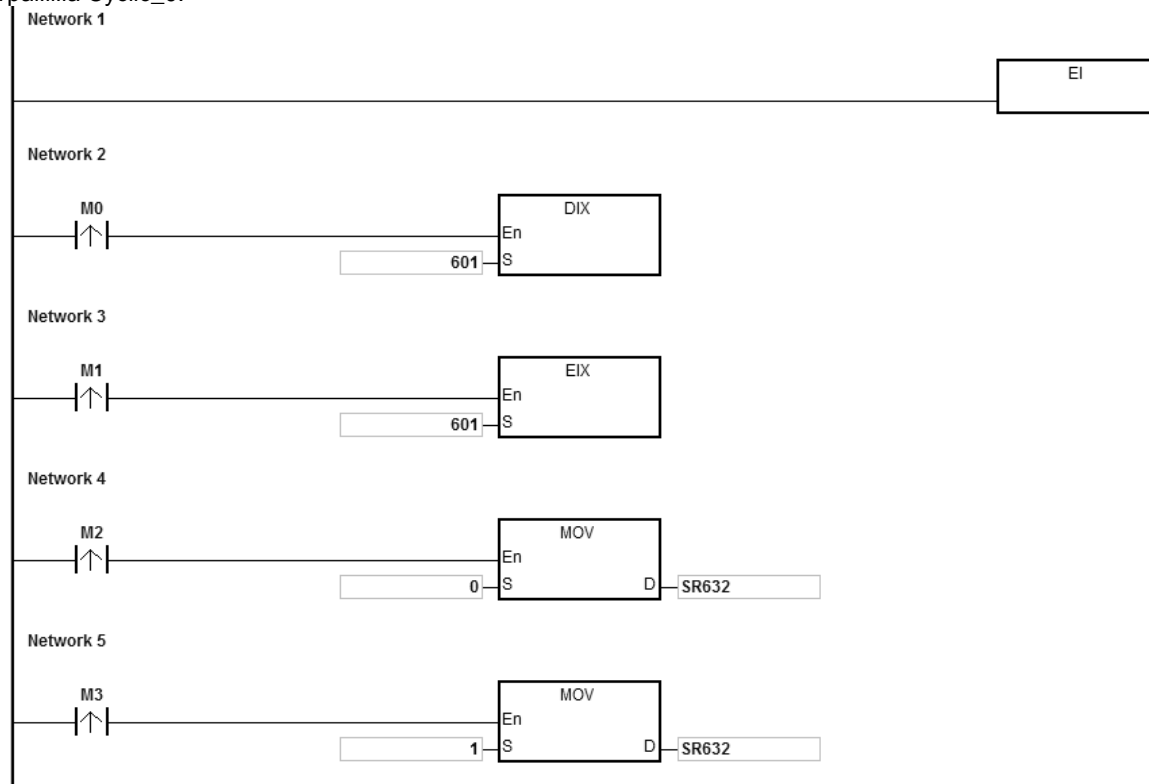
1. Инструкция EI предназначена для ввода разрешения подпрограммы прерывания в программе. (Подробнее по данной подпрограмме см. следующую страницу.)
2. Подпрограмма прерывания может находиться в любом месте программы между инструкцией EI и инструкцией DI. Если подпрограмма прерывания разрешена по всей программе, инструкцию DI можно не вставлять.
3. Во время исполнения одного прерывания запрос следующего прерывания не выполняется, а просто запоминается. И следующее прерывание будет исполнено только по завершении текущего прерывания. Например, во время исполнения прерывания I0 (№1) поступило 2 новых запроса прерывания I0 (№2 и №3), но запомнится только запрос прерывания №2.
4. Если поступает несколько запросов прерываний, первым будет исполняться прерывание, имеющее наиболее высокий приоритет. Если в отдельный момент времени одновременно поступает несколько запросов прерываний, первым будет исполняться прерывание с наименьшим номером.
5. Если подпрограмма прерывания оказывается между инструкциями DI и EI, она не может быть исполнена и запрос прерывания игнорируется. Не рекомендуется вводить запрет прерываний инструкцией DI прямо во время работы ПЛК.
6. Если подпрограмма прерывания выполняется по текущему входящему/выходящему сигналу, в программу можно вставить инструкцию REF или регистр DX/DY, чтобы обновлять состояние входа/выхода.
7. Номер каждого прерывания можно временно маскировать. Подробнее см. номера прерываний на следующей странице.

Пример:

- ◆ Откройте файл HWCONFIG в редакторе ISPSOFT и выставьте прерывание по таймеру I601 = 500 мс.

- ◆ Когда во время работы ПЛК программа Cyclic\_0 дойдет до инструкции EI, прерывание I601 будет разрешено и исполнено. По завершении исполнения прерывания контроллер продолжит исполнение основной программы.
- ◆ Если контакт M0 = ON, прерывание по таймеру I601 запрещено.
- ◆ Если контакт M1 = ON, прерывание по таймеру I601 разрешено
- ◆ Если контакт M2 = ON, SR623 = 0 и прерывание по таймеру I601 запрещено.
- ◆ Если контакт M3 = ON, SR623 = 1 и прерывание по таймеру I601 разрешено

Программа Cyclic\_0:



Подпрограмма прерывания:



График работы прерывания по таймеру:



**Дополнительные замечания:**

Всего существует 7 типов прерываний:

1. Прерывания по внешним сигналам (I000~I115)

Инструкция I000: срабатывание входа X0.0 по заднему фронту импульса. Инструкция I100: срабатывание входа X0.1 по переднему фронту импульса. Инструкция I101: срабатывание входа X0.1 по заднему фронту импульса. Все другие инструкции по такому же принципу.

2. Прерывания по аппаратным высокоскоростным компараторам (I200~I253)

Прерывания этого типа делятся на 6 групп. Каждая группа соответствует определенному аппаратному высокоскоростному счетчику (подробнее см. описание инструкции DCNT). У каждой группы прерываний есть по 4 номера (подробнее см. описание инструкции DHSCS). Например, у первой группы прерываний номера I200~I203, у второй группы номера I210~I213 и т. д.

3. Прерывания по программным высокоскоростным компараторам (I260~I267)

Всего существует 8 прерываний по программным высокоскоростным компараторам, и все 8 прерываний могут использоваться всеми 8 высокоскоростными счетчиками.

4. Прерывания передачи данных

Прерывание передачи данных можно использовать как инструкцию RS, иначе говоря, прерывание будет исполняться при получении определенного символа, или его можно использовать как общее прерывание передачи данных. Подробнее см. инструкцию AP11812 COMRS.

COM1: I300

COM2: I302

Плата 1: I304

Плата 2: I306

5. Прерывания по дополнительным модулям (I400~I431)

Каждому модулю выдается по одному прерыванию. Каждому дополнительному модулю можно назначить 1 службу прерывания.

6. Прерывания по быстродействующим выходам (I500~I519)

Запрос прерывания передается по завершении формирования выходного импульса; прерывания по завершении исполнения инструкций позиционирования (I500~I505) должны использоваться со специальными регистрами (SM) для активации службы прерываний. Например, когда контроллер по инструкции DDRVI завершает позиционирование по первой оси, передается запрос прерывания I500; чтобы включить службу прерываний, необходимо выставить SM471 = ON. Прерывания по завершении исполнения инструкции таблицы планирования позиционирования (I510~I519) должны исполняться вместе с инструкцией TPO. Запрос прерывания передается по завершении формирования выходного импульса;

7. Прерывания по таймерам (I601~I604)

Прерывания по таймеру прописываются в файле HWCONFIG.

Прерывания по таймеру I601~I603: по умолчанию 10 мс (шаг: 1 мс) (1~2000 мс).

Прерывания по таймеру I604: по умолчанию 1 мс (шаг: 0,1 мс) (0,1~200 мс).

Ниже в таблице приведены все номера прерываний вместе с описаниями и маскируемыми прерываниями (SR).

Номер прерывания	Описание	Маркируемые прерывания SR	Номер бита
I000	Прерывание по внешнему сигналу: срабатывание входа X0.0 по заднему фронту импульса.	SR623	0
I001	Прерывание по внешнему сигналу: срабатывание входа X0.1 по заднему фронту импульса.		1
I002	Прерывание по внешнему сигналу: срабатывание входа X0.2 по заднему фронту импульса.		2
I003	Прерывание по внешнему сигналу: срабатывание входа X0.3 по заднему фронту импульса.		3
I004	Прерывание по внешнему сигналу: срабатывание входа X0.4 по заднему фронту импульса.		4
I005	Прерывание по внешнему сигналу: срабатывание входа X0.5 по заднему фронту импульса.		5
I006	Прерывание по внешнему сигналу: срабатывание входа X0.6 по заднему фронту импульса.		6



Номер прерывания	Описание	Маркируемые прерывания SR	Номер бита	
	X0.6 по заднему фронту импульса.			
I007	Прерывание по внешнему сигналу: срабатывание входа X0.7 по заднему фронту импульса.		7	
I008	Прерывание по внешнему сигналу: срабатывание входа X0.8 по заднему фронту импульса.		8	
I009	Прерывание по внешнему сигналу: срабатывание входа X0.9 по заднему фронту импульса.		9	
I010	Прерывание по внешнему сигналу: срабатывание входа X0.10 по заднему фронту импульса.		10	
I011	Прерывание по внешнему сигналу: срабатывание входа X0.11 по заднему фронту импульса.		11	
I012	Прерывание по внешнему сигналу: срабатывание входа X0.12 по заднему фронту импульса.		12	
I013	Прерывание по внешнему сигналу: срабатывание входа X0.13 по заднему фронту импульса.		13	
I014	Прерывание по внешнему сигналу: срабатывание входа X0.14 по заднему фронту импульса.		14	
I015	Прерывание по внешнему сигналу: срабатывание входа X0.15 по заднему фронту импульса.		15	
I100	Прерывание по внешнему сигналу: срабатывание входа X0.0 по переднему фронту импульса.		SR624	0
I101	Прерывание по внешнему сигналу: срабатывание входа X0.1 по переднему фронту импульса.			1
I102	Прерывание по внешнему сигналу: срабатывание входа X0.2 по переднему фронту импульса.			2
I103	Прерывание по внешнему сигналу: срабатывание входа X0.3 по переднему фронту импульса.			3
I104	Прерывание по внешнему сигналу: срабатывание входа X0.4 по переднему фронту импульса.			4
I105	Прерывание по внешнему сигналу: срабатывание входа X0.5 по переднему фронту импульса.	5		
I106	Прерывание по внешнему сигналу: срабатывание входа X0.6 по переднему фронту импульса.	6		
I107	Прерывание по внешнему сигналу: срабатывание входа X0.7 по переднему фронту импульса.	7		
I108	Прерывание по внешнему сигналу: срабатывание входа X0.8 по переднему фронту импульса.	8		
I109	Прерывание по внешнему сигналу: срабатывание входа X0.9 по переднему фронту импульса.	9		

Номер прерывания	Описание	Маркируемые прерывания SR	Номер бита	
I110	Прерывание по внешнему сигналу: срабатывание входа X0.10 по переднему фронту импульса.		10	
I111	Прерывание по внешнему сигналу: срабатывание входа X0.11 по переднему фронту импульса.		11	
I112	Прерывание по внешнему сигналу: срабатывание входа X0.12 по переднему фронту импульса.		12	
I113	Прерывание по внешнему сигналу: срабатывание входа X0.13 по переднему фронту импульса.		13	
I114	Прерывание по внешнему сигналу: срабатывание входа X0.14 по переднему фронту импульса.		14	
I115	Прерывание по внешнему сигналу: срабатывание входа X0.15 по переднему фронту импульса.		15	
I200	Прерывание высокоскоростного компаратора 1 по аппаратному высокоскоростному счетчику 1		SR625	0
I201	Прерывание высокоскоростного компаратора 2 по аппаратному высокоскоростному счетчику 1			1
I202	Прерывание высокоскоростного компаратора 3 по аппаратному высокоскоростному счетчику 1			2
I203	Прерывание высокоскоростного компаратора 4 по аппаратному высокоскоростному счетчику 1			3
I210	Прерывание высокоскоростного компаратора 1 по аппаратному высокоскоростному счетчику 2			4
I211	Прерывание высокоскоростного компаратора 2 по аппаратному высокоскоростному счетчику 2			5
I212	Прерывание высокоскоростного компаратора 3 по аппаратному высокоскоростному счетчику 2			6
I213	Прерывание высокоскоростного компаратора 4 по аппаратному высокоскоростному счетчику 2			7
I220	Прерывание высокоскоростного компаратора 1 по аппаратному высокоскоростному счетчику 3			8
I221	Прерывание высокоскоростного компаратора 2 по аппаратному высокоскоростному счетчику 3	9		
I222	Прерывание высокоскоростного компаратора 3 по аппаратному высокоскоростному счетчику 3	10		
I223	Прерывание высокоскоростного компаратора 4 по аппаратному высокоскоростному счетчику 3	11		
I230	Прерывание высокоскоростного компаратора 1 по аппаратному высокоскоростному счетчику 4	12		
I231	Прерывание высокоскоростного компаратора 2 по	13		

Номер прерывания	Описание	Маркируемые прерывания SR	Номер бита	
	аппаратному высокоскоростному счетчику 4			
I232	Прерывание высокоскоростного компаратора 3 по аппаратному высокоскоростному счетчику 4		14	
I233	Прерывание высокоскоростного компаратора 4 по аппаратному высокоскоростному счетчику 4		15	
I240	Прерывание высокоскоростного компаратора 1 по аппаратному высокоскоростному счетчику 5	SR626	0	
I241	Прерывание высокоскоростного компаратора 2 по аппаратному высокоскоростному счетчику 5		1	
I242	Прерывание высокоскоростного компаратора 3 по аппаратному высокоскоростному счетчику 5		2	
I243	Прерывание высокоскоростного компаратора 4 по аппаратному высокоскоростному счетчику 5		3	
I250	Прерывание высокоскоростного компаратора 1 по аппаратному высокоскоростному счетчику 6		4	
I251	Прерывание высокоскоростного компаратора 2 по аппаратному высокоскоростному счетчику 6		5	
I252	Прерывание высокоскоростного компаратора 3 по аппаратному высокоскоростному счетчику 6		6	
I253	Прерывание высокоскоростного компаратора 4 по аппаратному высокоскоростному счетчику 6		7	
I260	Прерывание высокоскоростного компаратора 1 по программному высокоскоростному счетчику		SR627	0
I261	Прерывание высокоскоростного компаратора 2 по программному высокоскоростному счетчику			1
I262	Прерывание высокоскоростного компаратора 3 по программному высокоскоростному счетчику	2		
I263	Прерывание высокоскоростного компаратора 4 по программному высокоскоростному счетчику	3		
I264	Прерывание высокоскоростного компаратора 5 по программному высокоскоростному счетчику	4		
I265	Прерывание высокоскоростного компаратора 6 по программному высокоскоростному счетчику	5		
I266	Прерывание высокоскоростного компаратора 7 по программному высокоскоростному счетчику	6		
I267	Прерывание высокоскоростного компаратора 8 по программному высокоскоростному счетчику	7		
I300	Получено определенное слово, по которому исполняется прерывание передачи данных по порту COM1	SR628	0	

Номер прерывания	Описание	Маркируемые прерывания SR	Номер бита
I301	Резерв		1
I302	Получено определенное слово, по которому исполняется прерывание передачи данных по порту COM1		2
I303	Резерв		3
I304	Получено определенное слово, по которому исполняется прерывание передачи данных платой 1		4
I305	Резерв		5
I306	Получено определенное слово, по которому исполняется прерывание передачи данных платой 2		6
I307	Резерв		7
I500	Прерывание по быстродействующему выходу: позиционирование по 1-й оси по инструкции позиционирования завершено	SR629	0
I501	Прерывание по быстродействующему выходу: позиционирование по 2-й оси по инструкции позиционирования завершено		1
I502	Прерывание по быстродействующему выходу: позиционирование по 3-й оси по инструкции позиционирования завершено		2
I503	Прерывание по быстродействующему выходу: позиционирование по 4-й оси по инструкции позиционирования завершено		3
I504	Прерывание по быстродействующему выходу: позиционирование по 5-й оси по инструкции позиционирования завершено		4
I505	Прерывание по быстродействующему выходу: позиционирование по 6ой оси по инструкции позиционирования завершено		5
I510	Прерывание по быстродействующему выходу 1: исполнение по инструкции таблицы планирования позиционирования завершено		SR630
I511	Прерывание по быстродействующему выходу 2: исполнение по инструкции таблицы планирования позиционирования завершено	1	
I512	Прерывание по быстродействующему выходу 3: исполнение по инструкции таблицы планирования позиционирования завершено	2	
I513	Прерывание по быстродействующему выходу 4: исполнение по инструкции таблицы планирования	3	

Номер прерывания	Описание	Маркируемые прерывания SR	Номер бита
	позиционирования завершено		
I514	Прерывание по быстродействующему выходу 5: исполнение по инструкции таблицы планирования позиционирования завершено		4
I515	Прерывание по быстродействующему выходу 6: исполнение по инструкции таблицы планирования позиционирования завершено		5
I516	Прерывание по быстродействующему выходу 7: исполнение по инструкции таблицы планирования позиционирования завершено		6
I517	Прерывание по быстродействующему выходу 8: исполнение по инструкции таблицы планирования позиционирования завершено		7
I518	Прерывание по быстродействующему выходу 9: исполнение по инструкции таблицы планирования позиционирования завершено		8
I519	Прерывание по быстродействующему выходу 10: исполнение по инструкции таблицы планирования позиционирования завершено		9
I601	Прерывание по таймеру 1 (шаг 1 мс)	SR632	0
I602	Прерывание по таймеру 1 (шаг 1 мс)		1
I603	Прерывание по таймеру 1 (шаг 1 мс)		2
I604	Прерывание по таймеру 1 (шаг 0,1 мс)		3

Примечание: Если в отдельный момент времени одновременно поступает несколько запросов прерываний, первым будет исполняться прерывание с наименьшим номером. ПЛК завершает исполнение текущего прерывания и потом переходит к прерыванию по следующему номеру. Например, если во время исполнения прерывания I400 одновременно приходят запросы прерываний I500 и I300, по завершении выполнения прерывания I400 контроллер перейдет к выполнению прерывания I300 (потому что у него меньше номер).

API	Код инструкции			Операнд							Функция						
0503		EIX		<b>S</b>							Разрешение определенного прерывания						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S</b>													○			

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**

EIX
En
S

**S** : Источник данных

**Описание:**

1. В качестве вводных данных (S) инструкции могут использоваться только десятичные числа, а в качестве номера – номер прерывания. Если вместо номера прерывания указать другой номер, предупреждение выдаваться не будет, и такая инструкция просто не будет выполняться. Чтобы разрешить прерывание I500, нужно ввести инструкцию как EIX500. Подробнее см. список номеров прерывания в разделе, посвященном описанию инструкции EI.
2. В контроллерах серии AS прерывания по умолчанию разрешены. Если прерывания в программе запрещены инструкцией DIX, чтобы разрешить их, нужно добавить инструкцию EIX.
3. При помощи инструкции можно разрешить прерывания в SR623~SR634.
4. Если инструкция не выполнена, решение о выполнении или невыполнении прерывания принимается в зависимости от значения SR623~SR634.
5. Подробнее см. примеры инструкции EI в параграфе, посвященном API0501.

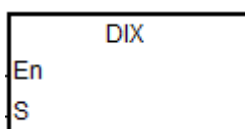
API	Код инструкции			Операнд							Функция				
0504		DIX		S							Запрет определенного прерывания				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S													○			

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



**S** : источник данных

**Описание:**

1. В качестве вводных данных (S) инструкции могут использоваться только десятичные числа, а в качестве номера – номер прерывания. Если вместо номера прерывания указать другой номер, предупреждение выдаваться не будет, и такая инструкция просто не будет выполняться. Чтобы запретить прерывание I500, нужно ввести инструкцию как DIX500. Подробнее см. список номеров прерывания в разделе, посвященном описанию инструкции EI.
2. Контроллеры серии AS поддерживают прерывания по умолчанию. Запрет на прерывания вводится инструкцией DIX.
3. При помощи инструкции можно запретить прерывания в SR623~SR634.
4. Если инструкция не выполнена, решение о выполнении или невыполнении прерывания принимается в зависимости от значения SR623~SR634.
5. Подробнее см. примеры инструкции EI в параграфе, посвященном API0501.

## 6.7 Инструкции обновления входов/выходов

### 6.7.1 Описание инструкций обновления входов/выходов

API	Код инструкции			Операнды								Функция				
0600		REF	P	D · n								Обновление входов/выходов				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D	○	○														
n								●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●												
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

REF		REFP	
En		En	
D		D	
n		n	

D : Обновляемый вход/выход

n : Количество входов/выходов, состояние которых обновляется

Описание:

- Состояние входа/выхода не обновляется до тех пор, пока не будет выполнена инструкция END. Когда начинается сканирование программы, состояния внешних входов считываются и сохраняются в памяти. После выполнения команды END состояния выходов из памяти отправляются на выходные клеммы. Поэтому, когда пользователю нужны последние данные состояния входов/выходов в процессе работы, можно использовать эту инструкцию или использовать регистры DX / DY.
- Значение операнда n должно быть кратным восьми, например 8, 16, 24 и до 256. Если значение меньше кратного восьми, будет рассматриваться следующее значение, кратное восьми. Например, значение 20 будет исправлено на 24.
- Количество высокоскоростных выходов сохраняется в D (для версии прошивки 1.04.00 и более поздних версий). Если n равно 1, это означает немедленное обновление высокоскоростного выхода и значения соответствующего SR. Если n равно 0, это означает отключение высокоскоростного выхода и обновление текущего значения SR. Например, во время выполнения этой инструкции, если n равно 0, а внешнее прерывание осуществляется через вход X0.0, то на X0.0 происходит внешнее прерывание, а



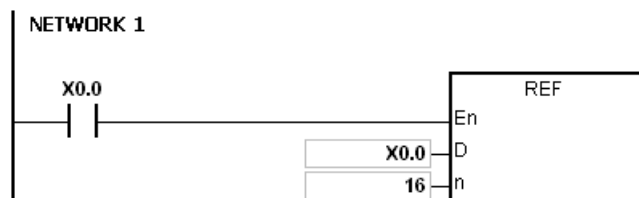
высокоскоростной выход Y0.0 немедленно отключается. ПЛК включает флаг остановки SM463 и обновляет текущее значение SR, соответствующего выходу.

**Примечание:** если флаг автоматического сброса остановки выхода включен, ПЛК выключает этот флаг и обновляет текущее значение SR, соответствующего выходу. Но при этом ПЛК не включает флаг остановки SM463.

Значение n	Объект D	Действие
n = кратное 8 значение	Y0.0 или X0.0	Немедленное обновление входов/выходов, см. примеры 1 и 2
n = 1	Высокоскоростной выход	Обновление новой импульсной позиции, см. пример 3
n = 0	Высокоскоростной выход Без флага автоматического сброса выхода	Отключение высокоскоростного выхода, включение флага SM463 и обновление текущей выходной позиции. См. пример 3
n = 0	Высокоскоростной выход С флагом автоматического сброса выхода	Отключение флага автоматического сброса выхода и обновление текущей выходной позиции. См. пример 3

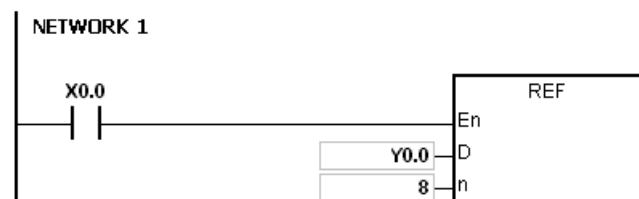
**Пример 1:**

1. Когда X0.0 включен, ПЛК немедленно считывает состояния входов X0.0~X0.15. Обновление входных сигналов происходят без задержки.



**Пример 2:**

Когда X0.0 включен, выходные сигналы от Y0.0~Y0.7 выдаются на клеммы выходов. Выходные сигналы обновляются немедленно, без необходимости ожидания выполнения команды END.



**Примечание:**

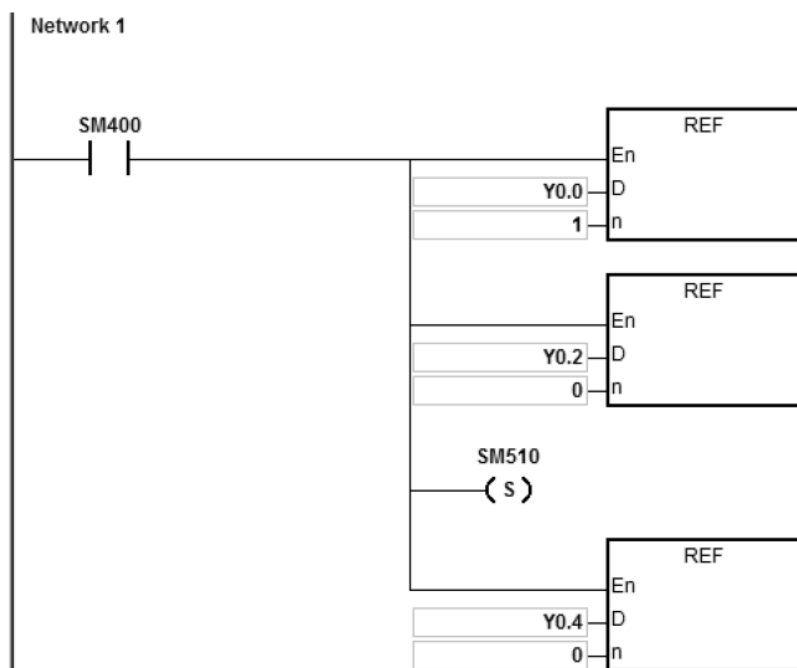
1. Если значение в D+n-1 выходит за пределы допустимого диапазона, инструкция не выполняется, включается флаг SM0, в SR0 фиксируется код ошибки 16#2003.

**Пример 3:**

1. При выполнении данной инструкции сигнал прерывания получен на вход X0.0.

Значение n	Объект D	Действие
n = 1	Высокоскоростной выход Y0.0	Немедленное обновление новой импульсной позиции Y0.0 (SR460)
n = 0	Высокоскоростной выход Y0.2 Без флага автоматического сброса выхода	<ol style="list-style-type: none"> <li>1. Отключение высокоскоростного выхода</li> <li>2. Включение флага SM483</li> <li>3. Обновление соответствующей выходной позиции SR460</li> </ol>
n = 0	Высокоскоростной выход Y0.4 С флагом автоматического сброса выхода	<ol style="list-style-type: none"> <li>1. Отключение флага автоматического сброса выхода SM510</li> <li>2. Обновление соответствующей выходной позиции SR500</li> </ol>

Программа внешнего прерывания X0.0:



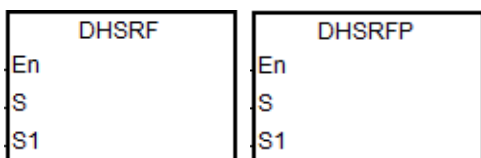
API	Код инструкции			Операнды								Функция				
0601	D	HSRF	P	<b>S · S<sub>1</sub></b>								Обновление значений высокоскоростного сравнения				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S</b>							○									
<b>S<sub>1</sub></b>								○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>												●	
<b>S<sub>1</sub></b>			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символ:**



- S** : Обновляемый высокоскоростной счетчик
- S<sub>1</sub>** : Количество входов/выходов, состояние которых обновляется

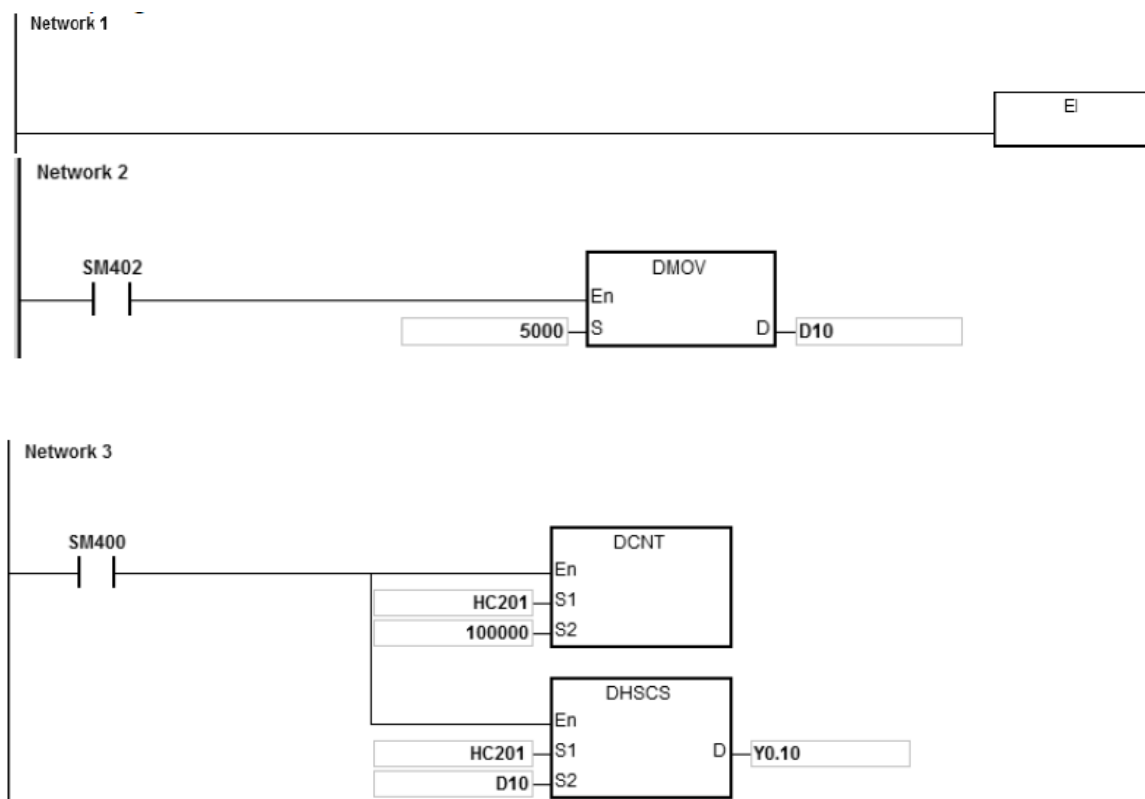
**Описание:**

1. Обновление сравнительного значения для ПЛК в его компараторе – это результат удачного сканирования программой инструкций DHSCS или DHSCR. Однако обновление может завершиться неудачно, если время сканирования слишком велико или входной сигнал проходит слишком быстро. В этом случае пользователи могут использовать инструкцию для задания нового сравнительного значения аппаратному компаратору в ПЛК и достижения результата сравнения в реальном времени.
2. **S** сохраняет номер указанного высокоскоростного счетчика, который необходимо обновить. **S<sub>1</sub>** имеет то же имя компонента или переменной, что и в инструкции DHSCS или DHSCR, и непосредственное значение для **S<sub>1</sub>** задано быть не может. Если **S<sub>1</sub>** отличается от операнда в включенной высокоскоростной инструкции сравнения, данная инструкция выполняться не будет.
3. Данная инструкция работает только в прошивке версии V1.04.00 или выше.

**Пример:**

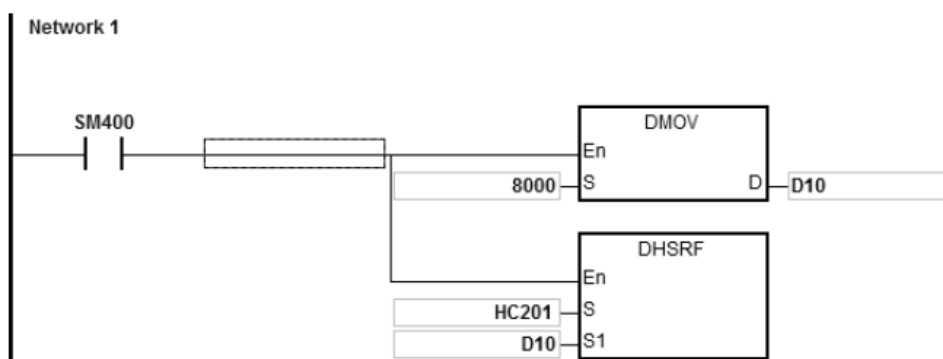
1. Когда ПЛК работает, сравнительное значение в инструкции DHSCS равно 5000.
2. Когда происходит внешнее прерывание на X0.0, сравнительное значение в DHSCS устанавливается сразу на 8000.

Основная программа:



Программа внешнего прерывания X0.0:

Сначала присвойте новое сравнительное значение той же переменной (например, в примере D10), а затем выполните инструкцию DHSRF для обновления



## 6.8 Вспомогательные инструкции

### 6.8.1 Описание вспомогательных инструкций

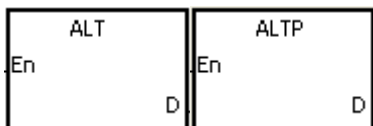
API	Код инструкции			Операнд								Функция				
0700		ALT	P	<b>D</b>								Смена состояния ON – OFF				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>D</b>	●	●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



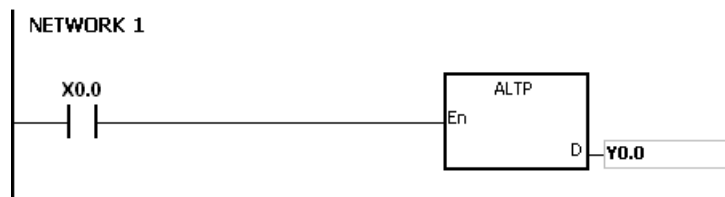
**D** : Управляемый регистр

**6** Описание:

1. При выполнении инструкции ALT состояние регистра, обозначенного параметром **D**, меняется с ON на OFF или наоборот.
2. Как правило, используется импульсная инструкция ALTP.

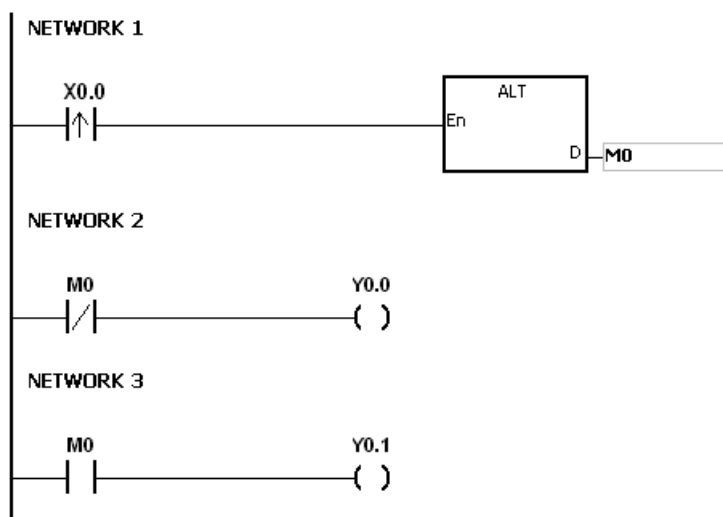
Пример 1:

Когда состояние входа X0.0 первый раз меняется с OFF на ON, выход Y0.0 переходит в состояние ON. Когда состояние входа X0.0 второй раз меняется с OFF на ON, выход Y0.0 переходит в состояние OFF.



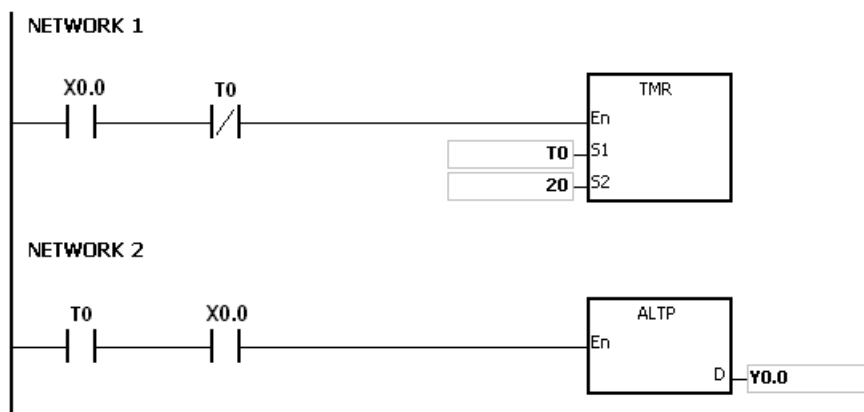
**Пример 2:**

Сначала контакт M0 = OFF. Поэтому выход Y0.0 = ON, а выход Y0.1 = OFF. Когда состояние входа X0.0 первый раз меняется с OFF на ON, контакт M0 = ON. Поэтому выход Y0.1 = ON, а выход Y0.0 = OFF. Когда состояние входа X0.0 второй раз меняется с OFF на ON, контакт M0 = OFF. Поэтому выход Y0.0 = ON, а выход Y0.1 = OFF.



**Пример 3:**

Когда вход X0.0 = ON, таймер T0 выдает импульс каждые две секунды. Состояние выхода Y0.0 меняется между ON и OFF в зависимости от очередного импульса, выданного таймером T0.



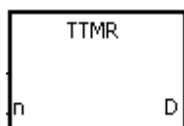
API	Код инструкции			Операнд							Функция						
0701		TTMR		D, n							Обучающийся таймер						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
D		●						●			○					
n	●	●						●	●		○		○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- D** : регистр, в котором записывается время
- n** : Множитель

**Описание:**

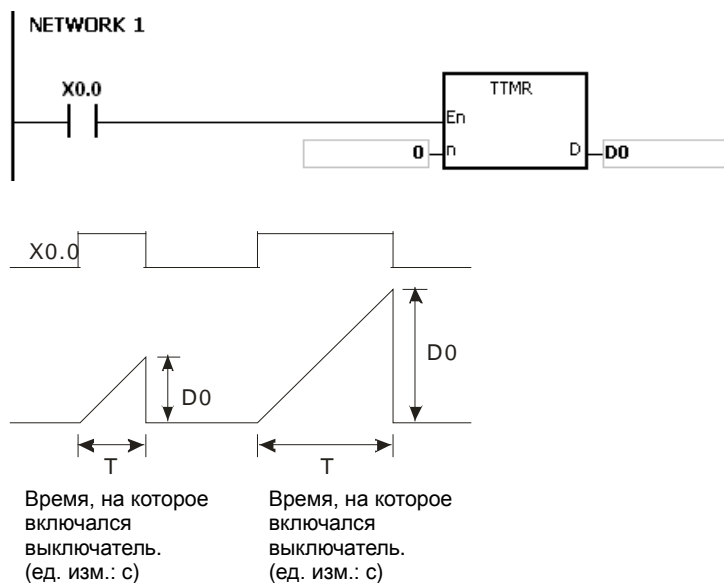
1. За единицу измерения времени берется секунда. Время, в течение которого выключатель находился в состоянии ON, умножается на множитель **n** и произведение записывается в регистр **D**. **D+1** используется только системой. После выполнения инструкции значение в **D+1** изменить нельзя, потому что иначе отсчет времени будет вестись неправильно.
2. Когда условный контакт переходит в состояние ON, значение в регистре **D** обнуляется.
3. Настройка множителя: если **n = 0**, регистр **D** берет за единицу измерения времени секунду. Если **n = 1**, время, в течение которого выключатель находился в состоянии ON, умножается на 10, и регистр **D** измеряет время единицей, равной 100 мс. Если **n = 2**, время, в течение которого выключатель находился в состоянии ON, умножается на 100, и регистр **D** измеряет время единицей, равной 10 мс.

n	D
K0 (ед. изм.: 1 с)	1×T
K1 (ед. изм.: 100 мс)	10×T
K2 (ед. изм.: 10 мс)	100×T

4. При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.
5. Операнд **n** может иметь значение в диапазоне от 0 до 2.

**Пример 1:**

1. Время, в течение которого выключатель, подсоединенный к входу X0.0, находился в состоянии ON, умножается на  $n$ , и произведение записывается в регистр D0. Параметры можно настроить выключателем.
2. Когда вход X0.0 переходит в состояние OFF, записанное в регистре D0 значение остается без изменений.



**Дополнительные замечания:**

1. Если значение  $D+1$  вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение  $n$  меньше 0 или значение  $n$  больше 2, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Если операнд  $D$  объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.



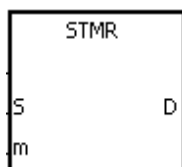
API	Код инструкции			Операнд								Функция				
0702		STMR		S, m, D								Специальный таймер				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S					○											
m	●	●						●	●		○		○	○		
D		●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S											●		
m		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



**S** : Номер таймера (T0~T511)

**m** : Настройка таймера

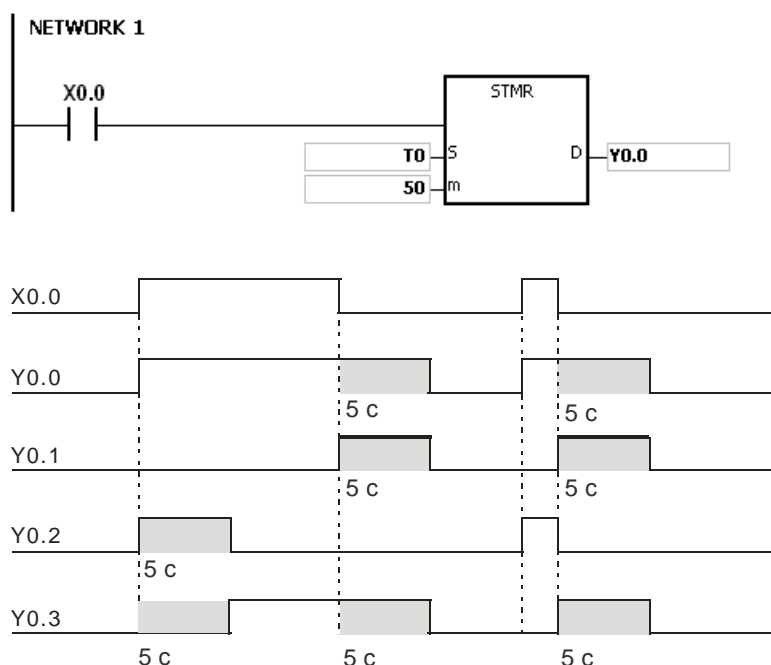
**D** : Регистр вывода

**Описание:**

- Инструкция STMR предназначена для задержки выключения реле, формирования импульса по включению/выключению, создания импульсных схем с периодом выходного сигнала, задаваемым таймером.
- Таймер, заданный инструкцией TMR, берет за единицу измерения времени 100 мс.
- Таймер, заданный инструкцией STMR, не может использоваться многократно.
- Параметр **D** представляет собой четыре последовательных регистра.
- Перед исполнением инструкции необходимо сбросить **D~D+3**.
- Если условный контакт не активирован и значение регистра удовлетворяет одному из двух условий, указанных ниже, тогда **D**, **D+1** и **D+3** переходят в состояние ON на **m** с, а потом переходят в состояние OFF. Если условный контакт не активирован и значение регистра не удовлетворяет ни одному из двух приведенных ниже условий, тогда **D~D+3** остается в состоянии OFF.
  - Значение таймера меньше или равно **m**, **D** = ON и **D+1** = OFF.
  - Значение таймера меньше **m**, **D +2** = OFF, а **D**, **D+1** и **D+3** = ON.
- При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.
- Операнд **m** может иметь значение в диапазоне от 1 до 32767.
-

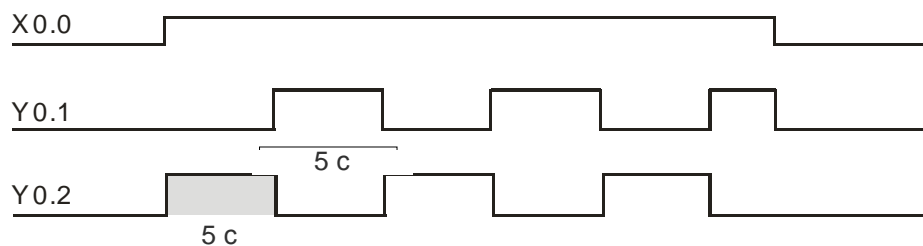
**Пример:**

1. Если вход X0.0 = ON, инструкция STMR запускает таймер T0 и выставляет его равным 5 с.
2. Выход Y0.0 – это контакт задержки выключения. Когда состояние входа X0.0 меняется с OFF на ON, выход Y0.0 переходит в состояние ON. Через 5 мин после того, как состояние входа X0.0 сменилось с ON на OFF, выход Y0.0 переходит в состояние OFF.
3. Когда состояние входа X0.0 меняется с ON на OFF, выход Y0.0 переходит в состояние ON на 5 с.
4. Когда состояние входа X0.0 меняется с OFF на ON, выход Y0.2 переходит в состояние ON на 5 с.
5. Через 5 с после того, как состояние входа X0.0 поменялось с OFF на ON, выход Y0.3 становится ON. Через 5 с после того, как состояние входа X0.0 поменялось с ON на OFF, выход Y0.3 становится OFF.



6. Если в цепи после условного контакта X0.0 стоит контакт Y0.3, через контакты Y0.1 и Y0.2 проходят пульсирующие токи. Когда вход X10 переходит в состояние OFF, выходы Y0.0, Y0.1 и Y0.3 тоже переходят в состояние OFF, а таймер T10 обнуляется.





**Дополнительные замечания:**

1. Если значение **D+3** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **m** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Если операнд **D** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа BOOL.

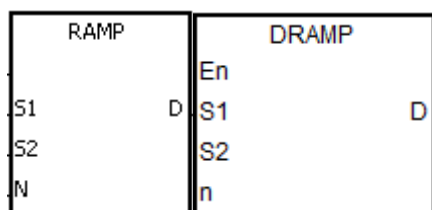
API	Код инструкции			Операнд						Функция					
0703	D	RAMP		$S_1, S_2, D, n$						Цикличное линейное изменение сигнала					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$	●	●					●	●	●		○	○	○	○		
$S_2$	●	●					●	●	●		○	○	○	○		
D		●					●	●								
n	●	●					●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

**Символьное обозначение:**



- $S_1$  : Начальный уровень сигнала на выходе
- $S_2$  : Конечный уровень сигнала на выходе
- D : Длительность сигнала на выходе
- n : Количество циклов опроса

**Описание:**

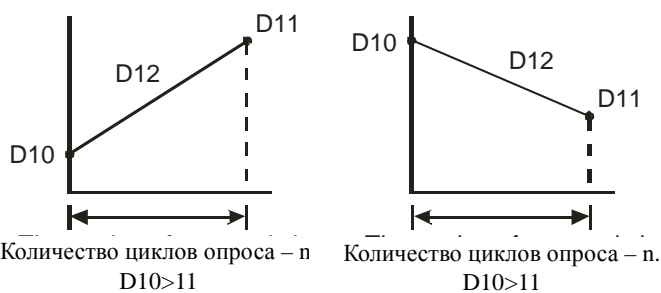
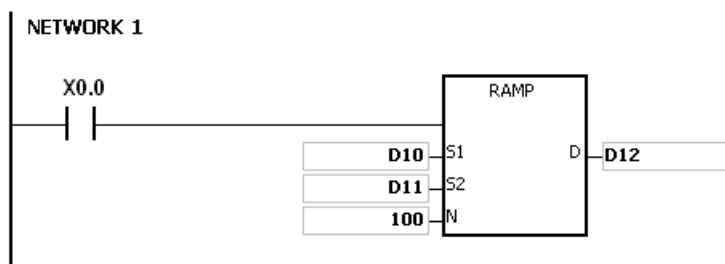
1. Данная инструкция предназначена для изменения сигнала. Характер изменения сигнала линейный и имеет абсолютную зависимость от времени опроса. Поэтому предлагается указывать фиксированное время опроса или вписывать данную инструкцию в прерывание по таймеру.
2. Начальный и конечный уровни сигнала на выходе блока заранее указываются в параметрах  $S_1$  и  $S_2$  соответственно. Когда вход  $X0.0 = ON$ , регистр D увеличивается от уровня, заданного в параметре  $S_1$ , до уровня, заданного в параметре  $S_2$ . Количество циклов опроса записывается в регистр D+1. Когда значение регистра D становится равным значению  $S_2$  или когда значение регистра D+1 становится равным n, задающему количеству циклов опроса, тогда флаг SM687 = ON.
3. Когда условный контакт не активирован, значения регистров D и D+1 равны 0, а флаг SM687 = OFF.
4. При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.
5. Подробнее см. фиксированное время опроса в руководстве пользователя редактора ISPSOft.

6. Операнд  $n$  может иметь значение в диапазоне от 1 до 32767. Если значение операнда  $n$  вне указанного диапазона, данная инструкция не выполняется.
7. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.
8. При установке флага SM686 заданное значение в регистре D обнуляется. Подробнее см. примеры ниже.

**Пример:**

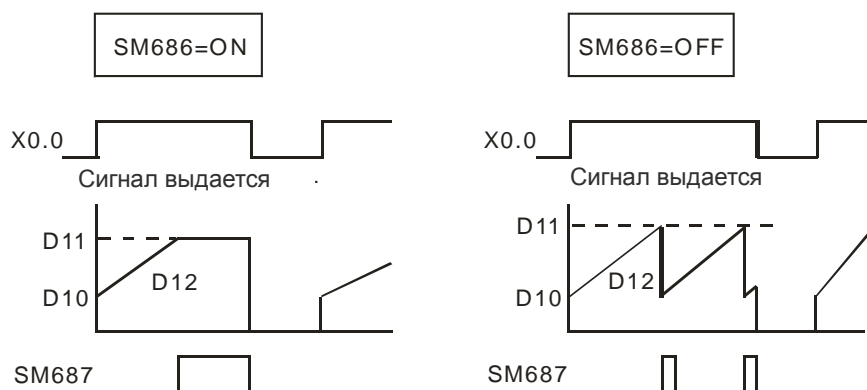
При помощи обработки данной инструкцией выходного аналогового сигнала можно добиться плавного пуска/останова оборудования.

1. Предположим, что инструкция выполняется. Когда вход X0.0 переходит в состояние OFF, исполнение инструкции прекращается. Когда вход X0.0 снова возвращается в состояние ON, флаг SM687 = OFF, значение регистра D12 сбрасываются до значения, заданного регистром D10, значение регистра D13 обнуляется и вычисление выполняется заново.
2. Флаг SM686 = OFF, и когда значение регистра D12 станет равным значению регистра D11, флаг SM687 = ON, показывая завершения цикла опроса. И когда регистр D12 будет сброшен до значения, заданного регистром D10, тогда регистр D13 сбросится в ноль.



Количество циклов опроса хранится в регистре D13

3. Если флаг SM686 = ON и значение регистра D12 стало равным значению, заданному регистром D11, значение регистра D12 не сбрасывается в ноль, а флаг SM687 = ON. Как только условный контакт замыкается, значение регистра D12 сбрасывается в ноль, а флаг SM687 = OFF. В зависимости от состояния флага SM686 значение регистра D12 меняется следующим образом.



**Дополнительные замечания:**

1. Если значение **D+1** вне пределов допустимого диапазона регистра, инструкция не выполняется, **SM0 = ON** и у **SR0** код ошибки 16#2003.
2. Если значение **n** меньше нуля, инструкция не выполняется, **SM0 = ON** и у **SR0** код ошибки 16#200B.
3. Если выполняется 16-битная инструкция, и операнд **D** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.
4. Если выполняется 32-битная инструкция, и операнд **D** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа DWORD/DINT.

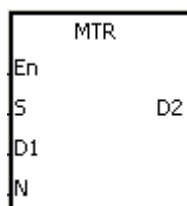
API	Код инструкции			Операнд								Функция			
0704		MTR		<b>S, D<sub>1</sub>, D<sub>2</sub>, n</b>								Матрица входов			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S</b>	○															
<b>D<sub>1</sub></b>		○														
<b>D<sub>2</sub></b>		○	○	○				○								
<b>n</b>	●	●						●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>	●												
<b>D<sub>1</sub></b>	●												
<b>D<sub>2</sub></b>	●												
<b>n</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- S** : Начальный регистр ввода при опросе матрицы
- D<sub>1</sub>** : Начальный регистр вывода при опросе матрицы
- D<sub>2</sub>** : Начальный соответствующий регистр при опросе матрицы
- n** : Количество опрашиваемых рядов

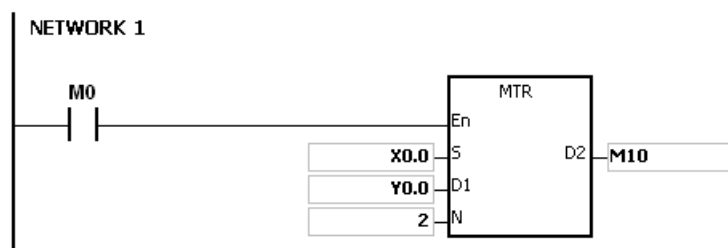
**Описание:**

1. Операнд **S** указывает начальный регистр ввода при опросе матрицы. 8 регистров, начиная с первого, указанного в операнде **S**, считаются регистрами ввода при опросе матрицы.
2. Операнд **D<sub>1</sub>** указывает транзисторный выход Y в качестве начального регистра при опросе матрицы. Когда условный контакт находится в состоянии OFF, состояние **n** регистров, начиная с регистра, заданного операндом **D<sub>1</sub>**, будет OFF.
3. За каждый цикл опроса обновляется состояние одного ряда входов. Всего в каждом ряду по 16 входов; опрос начинается с первого ряда и заканчивается **n**-м рядом,
4. 8 регистров ввода, начиная с регистра, заданного операндом **S**, подсоединены к **n** регистрам вывода, начиная с регистра, заданного операндом **D<sub>1</sub>**, образуя **n**-рядов выключателей. Состояния **n**-рядов выключателей считывается при матричном опросе и записываются в регистры, начиная с того, что задано операндом **D<sub>2</sub>**.
5. Для данной инструкции можно параллельно подсоединить до 8 рядов входных выключателей, чтобы получилось 64 входа (8×8=64).

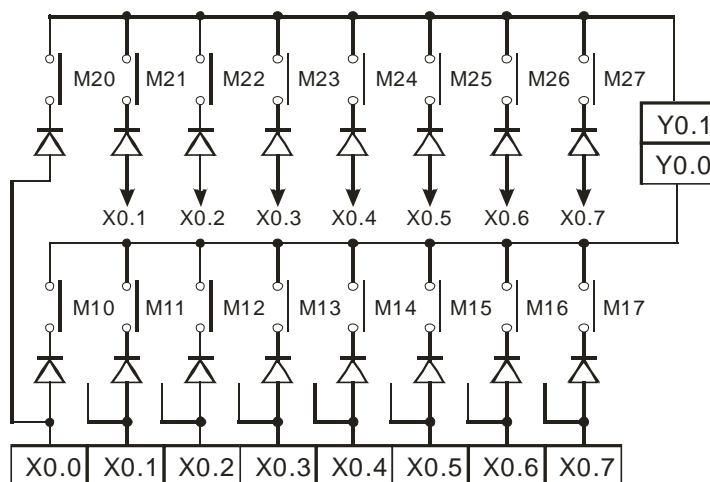
6. Интервал времени между поочередным исполнением инструкции должен быть больше времени, необходимого для обновления состояния входов и выходов модуля, иначе состояние входов правильно считать не получится.
7. Как правило, в качестве условного контакта для данной инструкции используется замыкающий контакт SM400.
8. Операнд **n** может иметь значение в диапазоне от 2 до 8.

**Пример 1:**

1. Когда M0 = ON, инструкция MTR начинает исполняться. Контроллер по очереди считывает состояние двух рядов выключателей и записывает результат во встроенные реле M10~M17 и M20~M27 соответственно.



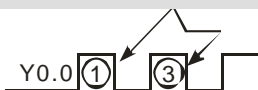
2. Ниже приведена схема внешних соединений матрицы размером 2x8, состоящей из входов X0.0~X0.7 и выходов Y0.0~Y0.7. Соответствующие внутренние реле 16 выключателей – это M10~M17 и M20~M27.



3. 8 регистров ввода, начиная с входа X0.0, подсоединены к двум регистрам вывода, начиная с выхода Y0.0, образуя два ряда выключателей. Состояния двух рядов выключателей считывается при матричном опросе и записываются в регистры, начиная с M10, заданного операндом D<sub>2</sub>. Состояния выключателей первого ряда записываются в M10~M17, а состояния выключателей второго ряда – в M20~M27.



Чтение входных сигналов второго ряда



Чтение входных сигналов первого ряда



#### Дополнительные замечания:

1. Если во время исполнения инструкции время цикла опроса будет слишком длинным или наоборот коротким, состояние выключателей считать правильно не удастся. Поэтому см. следующие рекомендации по решению данного вопроса.
  - Если цикл опроса слишком короткий, вход/выход может не успеть вовремя изменить состояние, поэтому результат считывания состояния входов может получиться недостоверным. Поэтому, чтобы избежать этого, можно ввести фиксированное время цикла опроса.
  - Если цикл опроса слишком длинный, выключатель может изменять состояние слишком долго. Поэтому можно записать данную инструкцию в прерывание по таймеру, чтобы определить фиксированное время исполнения данной инструкции.
2. Если  $S+7$ ,  $D_1+n-1$  или  $D_2+(n*8)-1$  превышает допустимый диапазон регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
3. Если значение  $n$  меньше 2 или значение  $n$  больше 8, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
4. Если операнд  $S$  объявлен в редакторе ISPSOft, типом данных будет МАССИВ [8] типа BOOL.

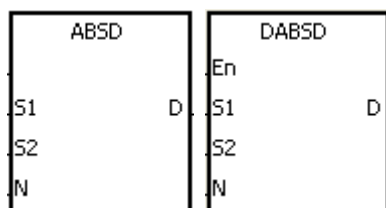
API	Код инструкции			Операнд								Функция					
0705	D	ABSD		$S_1, S_2, D, n$								Абсолютный многоустановочный счетчик					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$	●	●			●	●	●	●	●							
$S_2$	●	●			●	●	●	●	●							
D		●	●	●				●								
n	●	●						●	●			○	○	○	○	

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D	●												
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символьное обозначение:



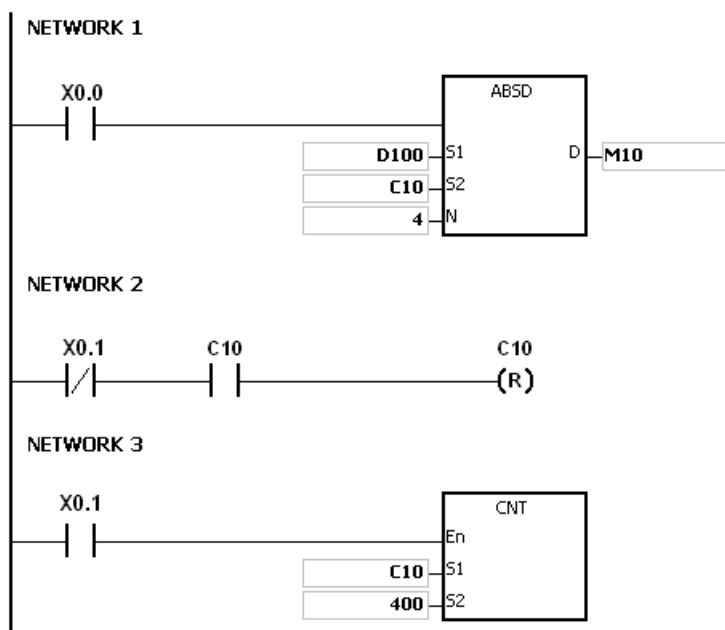
- $S_1$  : Начальный сравниваемый регистр
- $S_2$  : Сравнимое значение
- D : Результат сравнения
- n : Количество групп сравнения

Описание:

- Инструкция ABSD предназначена для формирования нескольких импульсов в зависимости от текущих значений счетчика.
- 32-битный счетчик может использоваться только в инструкции DABSD, а в регистре E – не может.
- Операнд n, используемый инструкции ABSD, может иметь значение в диапазоне от 1 до 256.

Пример 1:

- Перед исполнением инструкции ABSD заданные настройки записываются в регистры D100~D107 инструкцией MOV. В четных регистрах значения минимальные, а в нечетных – максимальные.
- Когда вход X0.0 = ON, текущее значение счетчика C10 сравнивается с максимальными и минимальными значениями в регистрах D100~D107, а результат сравнения записывается в реле M10~M13.
- Когда вход X0.0 = OFF, исходное состояние реле M10~M13 остается без изменений.

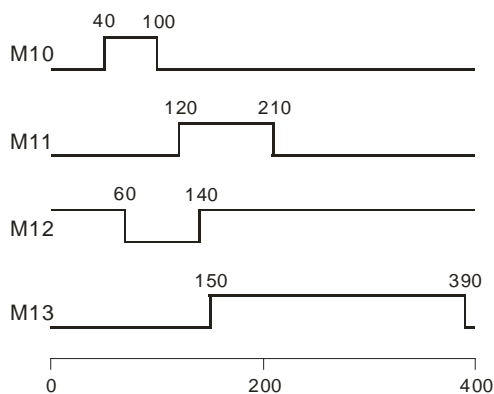


4. Пока текущее значение счетчика C10 остается в пределах диапазона, ограниченного минимальным и максимальным значениями, реле M10~M13 = ON. Если нет, реле M10~M13 становятся OFF.

Минимальное значение	Максимальное значение	Текущее значение счетчика C10	Результат
D100=40	D101=100	$40 \leq C10 \leq 100$	M10=ON
D102=120	D103=210	$120 \leq C10 \leq 210$	M11=ON
D104=140	D105=170	$140 \leq C10 \leq 170$	M12=ON
D106=150	D107=390	$150 \leq C10 \leq 390$	M13=ON

5. Предположим, что минимальное значение стало больше максимального. Когда текущее значение счетчика C10 меньше максимального значения ( $C10 < 60$ ) или когда текущее значение счетчика C10 больше минимального значения ( $C10 > 140$ ), реле M12 = ON. В остальных случаях реле M12 = OFF.

Минимальное значение	Максимальное значение	Текущее значение счетчика C10	Результат
D100=40	D101=100	$40 \leq C10 \leq 100$	M10=ON
D102=120	D103=210	$120 \leq C10 \leq 210$	M11=ON
D104=140	D105=60	$60 \leq C10 \leq 140$	M12=OFF
D106=150	D107=390	$150 \leq C10 \leq 390$	M13=ON



**Дополнительные замечания:**

1. Если  $S+2*n-1$  в инструкции ABSD больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если  $S+4*n-1$  в инструкции DABSD больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если  $D+n-1$  в инструкции ABSD больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
4. Если  $D+2*n-1$  в инструкции DABSD больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
5. Если значение операнда  $n$  в инструкции ABSD меньше 1 или больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

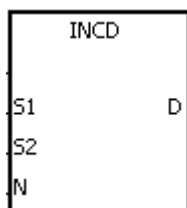
API	Код инструкции			Операнд								Функция				
0706		INCD		$S_1, S_2, n, D$								Инкрементный многоустановочный счетчик				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$	●	●			●	●		●	●							
$S_2$	●	●			●	●		●	●							
D		●	●	●				●								
n	●	●						●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
D	●												
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:



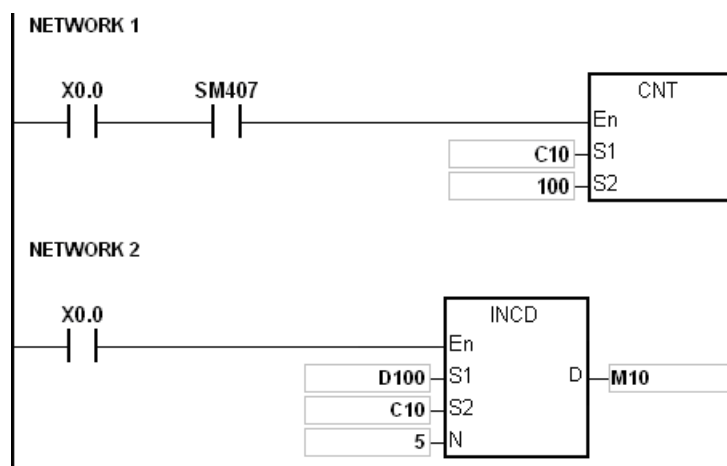
- $S_1$  : Начальный сравниваемый регистр
- $S_2$  : Номер счетчика
- D : Результат сравнения
- n : Количество групп сравнения

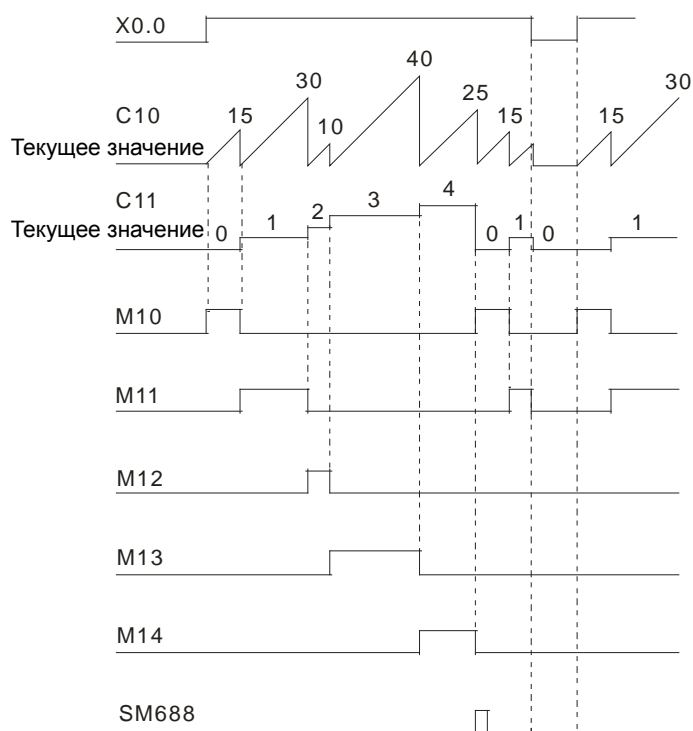
Описание:

1. Инструкция INCD предназначена для формирования несколько импульсов для пары счетчиков.
2. Текущее значение операнда  $S_2$  сравнивается с заданным значением, прописанным в операнде  $S_1$ . Если текущее значение совпадает с заданным, текущее значение операнда  $S_2$  сбрасывается в ноль, а текущий номер группы сравнения записывается в  $S_2+1$ .
3. После сравнения текущих значений  $S_2$  и n-групп значений флаг SM688 = ON и запускается цикл опроса.
4. Если условный контакт не активирован, значение операнда  $S_2 = 0$ , значение  $S_2+1 = 0$ ,  $D \sim D+n-1 = OFF$  и флаг SM688 = OFF.
5. При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.
6. Операнд n может иметь значение в диапазоне от 1 до 256.

**Пример:**

1. Перед исполнением инструкции INCD заданные настройки записываются в регистры D100~D104 инструкцией MOV. Значения регистров D100~D104: 15, 30, 10, 40, и 25 соответственно.
2. Текущее значение счетчика C10 сравнивается с заданными значениями, прописанными в регистрах D100~D104. Если текущее значение совпадает с заданным, счетчик C10 сбрасывается в ноль и отсчет начинается снова.
3. Текущий номер группы сравнения записывается в C11.
4. Если значение в C11 изменяется на единицу, реле M10~M14 реагируют соответственно. Подробнее см. временную схему ниже.
5. После сравнения текущих значений C10 и значений регистров D100~D104 флаг SM688 = ON и запускается цикл опроса.
6. Когда состояние входа X0.0 меняется с ON на OFF, C10 и C11 сбрасываются в ноль, а реле M10~M14 переходят в состояние OFF. Когда вход X0.0 снова становится ON, исполнение инструкции начинается с самого начала.





**Дополнительные замечания:**

1. Если значение  $S_2+1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $S_1+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
3. Если значение  $D+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
4. Если значение  $n$  меньше 1 или значение  $n$  больше 256, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
5. Если операнд  $S_2$  объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.

API	Код инструкции			Операнд										Функция			
0708	D	PIDE		См. список ниже										Алгоритм ПИД-регулирования			
Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F	
PID_RUN	●	●	●	●				●									
SV								●								○	
PV								●								○	
PID_MODE								●					○	○			
PID_MAN	●	●	●	●				●									
MOUT_AUTO	●	●	●	●				●									
CYCLE								●									
Kc_Kp								●									
Ti_Ki								●									
Td_Kd								●									
Tf								●									
PID_EQ	●	●	●	●				●									
PID_DE	●	●	●	●				●									
PID_DIR	●	●	●	●				●									
ERR_DBW								●	●							○	
MV_MAX								●	●							○	
MV_MIN								●	●							○	
MOUT								●									
BIAS								●	●							○	
I_MV								●									
MV								●									

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS



**Символьное обозначение:**

DPIDE	
En	
PID_RUN	MV
SV	
PV	
PID_MODE	
PID_MAN	
MOUT_AUTO	
CYCLE	
Kc_Kp	
Ti_Ki	
Td_Kd	
Tf	
PID_EQ	
PID_DE	
PID_DIR	
ERR_DBW	
MV_MAX	
MV_MIN	
MOUT	
BIAS	
I_MV	

<b>EN</b>	: Включение/выключение инструкции
<b>PID_RUN</b>	: Запуск алгоритма ПИД-регулирования
<b>SV</b>	: Заданное значение (SV)
<b>PV</b>	: Текущее значение (PV)
<b>PID_MODE</b>	: Режим ПИД-регулирования
<b>PID_MAN</b>	: Авт./ручной режим ПИД-регулирования
<b>MOUT_AUTO</b>	: Авт./ручной режим выходного значения
<b>CYCLE</b>	: Время опроса (ЦИКЛ)
<b>Kc_Kp</b>	: Пропорциональная составляющая
<b>Ti_Ki</b>	: Интегральная составляющая (с или 1/c)
<b>Td_Kd</b>	: Дифференциальная составляющая (с)
<b>Tf</b>	: Временная константа дифференциальной составляющей (с)
<b>PID_EQ</b>	: Тип формулы ПИД-регулирования
<b>PID_DE</b>	: Вычисление погрешности дифференциальной составляющей ПИД-регулирования
<b>PID_DIR</b>	: Направление ПИД-регулирования: прямое/обратное (PID_DIR)
<b>ERR_DBW</b>	: Диапазон, в пределах которого величина погрешности берется за ноль
<b>MV_MAX</b>	: Максимальное выходное значение (MV_MAX)
<b>MV_MIN</b>	: Минимальное выходное значение (MV_MIN)
<b>MOUT</b>	: Выходное значение в ручном режиме (MOUT)
<b>BIAS</b>	: Выходное значение при прямом направлении
<b>I_MV</b>	: Накопленное интегральное значение
<b>MV</b>	: Выходное значение (MV)

**Описание:**

1. Инструкция предназначена для организации ПИД-регулирования. По завершении времени опроса запускается алгоритм ПИД-регулирования. Под аббревиатурой ПИД-регулирования подразумевается пропорционально-интегрально-дифференциальное регулирование. ПИД-регулирование широко применяется для управления механическим, пневматическим и электронным оборудованием.
2. Порядок настройки параметров приведен ниже.

Операнд	Тип данных	Функция	Диапазон значений	Описание
<b>PID_RUN</b>	BOOL	Запуск алгоритма ПИД-регулирования		True: алгоритм ПИД-регулирования включен. False: выходное значение (MV) сбрасывается в ноль и алгоритм ПИД-регулирования выключен.
<b>SV</b>	REAL	SV	Диапазон чисел одинарной точности с плавающей запятой	Заданное значение
<b>PV</b>	REAL	PV	Диапазон чисел одинарной точности с плавающей запятой	Текущее значение
<b>PID_MODE</b>	DWORD/DINT	Режим ПИД-регулирования		0: автоматический Когда операнд PID_MAN меняет значение с True на False, запускается алгоритм автоматической обработки выходного значения (MV). 1: Параметры регулирования температуры настраиваются автоматически. По завершении настройки параметров регистр автоматически устанавливается в 0, и подставляются соответствующие значения параметров K <sub>c</sub> -K <sub>p</sub> , T <sub>i</sub> -K <sub>i</sub> , T <sub>d</sub> -K <sub>d</sub> и T <sub>f</sub> . Примечание: в режиме автоматического регулирования самостоятельно менять параметры настройки нельзя.
<b>PID_MAN</b>	BOOL	Авт./ручной режим ПИД-регулирования		True: ручной Выходное значение выдается в ручном режиме (MOUT), но при этом все равно остается в пределах от MV_MIN до MV_MAX. Если в операнде PID_MODE выставлена 1, данный параметр становится недействительным. False: авто

Операнд	Тип данных	Функция	Диапазон значений	Описание
				Выходное значение (MV) обрабатывается алгоритмом ПИД-регулирования и значение на выходе будет в пределах диапазона от MV_MIN до MV_MAX.
<b>MOUT_AUTO</b>	BOOL	Автоматическая смена режима MOUT	True: авто MOUT изменяется в зависимости от значения MV. False: стандарт MOUT не изменяется в зависимости от значения MV.	
<b>CYCLE</b>	DWORD/DINT	Время опроса (Ts)	1~40 000 (ед. изм.: мс)	При выполнении инструкции запускается алгоритм ПИД-регулирования, время опроса соответствует заданному, а значение MV обновляется (ПЛК должен выполнить инструкцию; время опроса не запустится автоматически). Если значение операнда Ts меньше 1, это все равно будет считаться как 1. Если значение операнда Ts больше 40 000, оно соответственно будет считаться как 40 000. При добавлении инструкции алгоритма ПИД-регулирования в интервальное прерывание время опроса будет равно интервалу между прерываниями по таймеру. Настройка времени опроса в данном случае будет недействительной.

Операнд	Тип данных	Функция	Диапазон значений	Описание
<b>Kc_Kp</b>	REAL	Вычисленная пропорциональная составляющая (Kc или Kp в зависимости от настройки PID_EQ)	Диапазон положительных чисел одинарной точности с плавающей запятой	Вычисленная пропорциональная составляющая регулирования (Kc или Kp) Если составляющая P меньше 0, тогда Kc_Kp будет равен 0. И при этом, если Kc_Kp равен 0, он не будет контролироваться P.
<b>Ti_Ki</b>	REAL	Интегральная составляющая (Ti или Ki в зависимости от настройки PID_EQ)	Диапазон положительных чисел одинарной точности с плавающей запятой (ед. изм.: Ti = c; Ki = 1/c)	Если вычисленная составляющая I меньше 0, Ti_Ki будет равен 0. Если Ti_Ki равен 0, он не будет контролироваться I.
<b>Td_Kd</b>	REAL	Дифференциальная составляющая (Td или Kd в зависимости от настройки PID_EQ)	Диапазон положительных чисел одинарной точности с плавающей запятой (ед. изм.: c)	Если вычисленная составляющая D меньше 0, Td_Kd будет равен 0. Если Ti_Ki равен 0, он не будет контролироваться D.
<b>Tf</b>	REAL	Временная константа дифференциальной составляющей	Диапазон положительных чисел одинарной точности с плавающей запятой (ед. изм.: c)	Если временная константа дифференциальной составляющей меньше 0, тогда Tf будет равен 0 и не будет контролироваться временной константой дифференциальной составляющей (сглаживание дифф. составляющей).

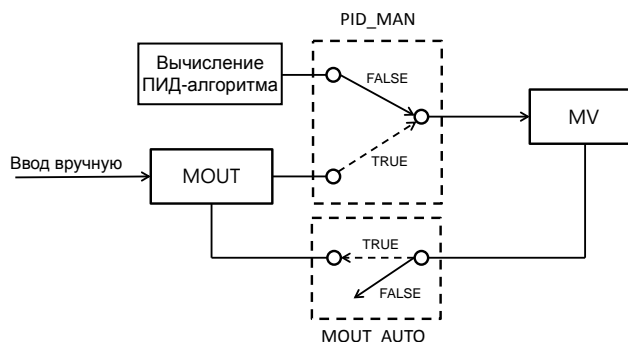
Операнд	Тип данных	Функция	Диапазон значений	Описание
PID_EQ	BOOL	Тип формулы ПИД-регулирования	True: зависимая формула False: независимая формула	
PID_DE	BOOL	Вычисление погрешности дифференциальной составляющей ПИД-регулирования	True: контрольное значение дифференциальной составляющей вычисляется по изменениям текущего значения PV (производная значения PV). False: контрольное значение дифференциальной составляющей вычисляется по изменению погрешности (E) (производная погрешности).	
PID_DIR	BOOL	Направление ПИД-регулирования: прямое/обратное	True: обратное ( $E=SV-PV$ ) False: прямое ( $E=PV-SV$ )	
ERR_DBW	REAL	Диапазон, в пределах которого величина погрешности берется за ноль	Диапазон чисел одинарной точности с плавающей запятой	Величина погрешности (E) представляет собой разность заданного (SV) и текущего (PV) значений. Когда операнд равен 0, значит данная функция выключена; во всех остальных случаях процессорный модуль контроллера проверяет, что погрешность меньше абсолютного значения ERR_DBW и что погрешность соответствует перекрестному условию состояния. Если погрешность меньше абсолютного значения ERR_DBW и при этом удовлетворяет перекрестному условию состояния, она считается равной 0 и алгоритм ПИД-регулирования

Операнд	Тип данных	Функция	Диапазон значений	Описание
				запускается. Если же нет, погрешность будет приведена в алгоритм ПИД-регулирования по правилам нормальной обработки.
<b>MV_MAX</b>	REAL	Максимальное выходное значение	Диапазон чисел одинарной точности с плавающей запятой	Предположим, что операнд <b>MV_MAX</b> выставлен равным 1 000. Когда выходное значение MV больше 1 000, на выходе будет 1 000. Значение у операнда <b>MV_MAX</b> должно быть больше значения операнда <b>MV_MIN</b> . Иначе максимум (MV) и минимум (MV) поменяются местами.
<b>MV_MIN</b>	REAL	Минимальное выходное значение	Диапазон чисел одинарной точности с плавающей запятой	Предположим, что операнд <b>MV_MIN</b> выставлен равным -1 000. Когда выходное значение MV меньше -1 000, на выходе будет -1 000.
<b>MOUT</b>	REAL	MV	Диапазон чисел одинарной точности с плавающей запятой	Если выбран ручной режим ПИД-регулирования (PID Manual), в качестве выходного значения (MV) будет выдаваться значение, заданное операндом MOUNT, и в диапазоне от MV_MAX до MV_MIN.
<b>BIAS</b>	REAL	Выходное значение при прямом направлении	Диапазон чисел одинарной точности с	Выходное значение, используемое для прямого ПИД-регулирования.

Операнд	Тип данных	Функция		Диапазон значений	Описание	
				плавающей запятой		
<b>I_MV</b> (занимает 15 DWordDevice □□□□□□)	REAL	I_MV	Накопленное интегральное значение	Диапазон чисел одинарной точности с плавающей запятой	Накопленное интегральное значение, которое обычно хранится в качестве справочного. При необходимости можно стереть его или изменить по собственному усмотрению. Если выходное значение MV больше значения MV_MAX или меньше значения MV_MIN, накопленное интегральное значение в операнде I_MV остается без изменений.	
		I_MV+1	В нем временно сохраняется предыдущее значение погрешности.			
		I_MV+2 ~ I_MV+5	Только для контроллера.			
		I_MV+6	В нем временно сохраняется предыдущее текущее значение (PV).			
		I_MV+7 ~ I_MV+14	Только для контроллера.			
<b>MV</b>	REAL	MV	Выходное значение MV в пределах от MV_MIN до MV_MAX.			

6

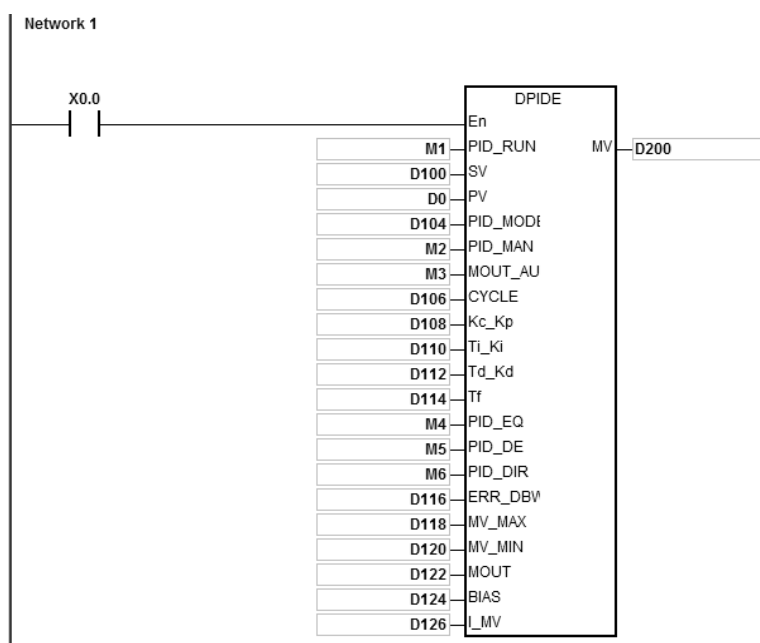
**Принцип переключения PID\_MAN / MOUT\_AUTO:**



1. При смене режима регулирования (**PID\_MAN=0**) с автоматического на ручной можно выставить MOUT\_AUTO = 1, и тогда выходное значение MOUT будет выдаваться вместе с выходным значением MV. При смене режима на ручной (**PID\_MAN=1**) в операнде MOUT\_AUTO можно выставить 0.
2. Когда значение операнда **PID\_RUN** меняется с TRUE на FALSE, контроллер сбрасывает выходное значение MV в ноль. Если нужно сохранить выходное значение MV, можно выставить в операнде EN значение FALSE, чтобы выключить инструкцию и оставить значение MV.

**Пример:**

1. Перед исполнением инструкции DPID необходимо настроить все параметры.
2. Когда вход X0.0 = ON, инструкция начинает исполняться. Когда M1 = ON, запускается алгоритм ПИД-регулирования. Когда M1 = OFF, выходное значение (MV) становится равным 0, а значение MV записывается в регистре D200. Когда вход X0.0 = OFF, инструкция не выполняется, и предыдущие данные остаются без изменений.

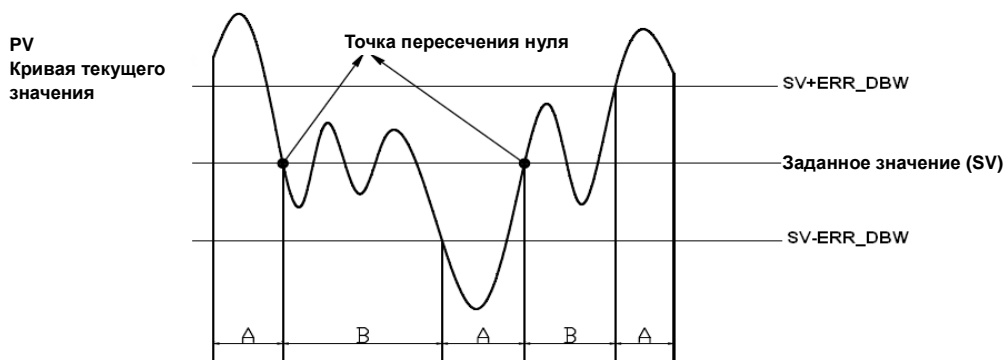


**Дополнительные замечания:**

1. Инструкция может использоваться многократно, но регистры, определенные **I\_MV~I\_MV+14**, должны быть разными.
2. Операнд **I\_MV** занимает 30 регистров. Операнд **I\_MV**, используемый в инструкции ПИД-регулирования на примере выше, занимает регистры D126~D155.
3. Инструкция DPID может применяться только в циклических операциях и интервальных прерываниях. При добавлении инструкции DPID в интервальное прерывание время опроса (ЦИКЛ) будет равно интервалу между прерываниями по таймеру.



4. При выполнении инструкции запускается алгоритм DPID, время опроса (ЦИКЛ) соответствует заданному, а значение MV обновляется. При добавлении инструкции в интервальное прерывание время опроса (ЦИКЛ) будет равно интервалу между прерываниями по таймеру. Алгоритм ПИД-регулирования запускается с интервалом, равным интервалу между прерываниями по таймеру.
5. Перед запуском алгоритма ПИД-регулирования текущее значение в инструкции PID должно стабилизироваться. Если алгоритм DPID должен запускаться по определенному значению на входе блока, необходимо отметить, сколько времени уходит на преобразование входного аналогового сигнала в цифровой входной сигнал.
6. Когда текущее значение (PV) находится в пределах диапазона значений операнда **ERR\_DBW**, величина погрешности сначала приводится в алгоритм ПИД-регулирования по правилам нормальной обработки, а затем процессорный модуль контроллера проверяет, что погрешность соответствует перекрестному условию состояния: текущее значение (PV) становится больше заданного значения (SV). Если условие удовлетворяется, текущая погрешность будет считаться равной нулю, и алгоритм ПИД-регулирования будет исполняться. Как только текущее значение (PV) покинет пределы диапазона значений операнда **ERR\_DBW**, текущая погрешность будет снова приведена в алгоритм ПИД-регулирования. Если операнд **PID\_DE = true**, это значит, что по изменению текущего значения (PV) будет рассчитываться контрольное значение производной, и если перекрестное условие состояния будет удовлетворено, контроллер будет расценивать  $\Delta PV$  равной нулю при запуске алгоритма ПИД-регулирования. ( $\Delta PV =$  текущее значение **PV** – предыдущее значение **PV**). На показанном ниже примере можно видеть, что на отрезке А текущая погрешность будет приведена в алгоритм ПИД-регулирования по правилам нормальной обработки, а на отрезке В текущая погрешность или  $\Delta PV$  будет считаться равной нулю при запуске алгоритма ПИД-регулирования.



**Алгоритм ПИД-регулирования:**

1. Когда операнд **PID\_MODE = 0**, алгоритм ПИД-регулирования работает в автоматическом режиме.

- **Независимая формула и производная E (PID\_EQ=False и PID\_DE=False)**

$$MV = K_p E + Ki \int_0^t E dt + K_d * \frac{dE}{dt} + BIAS \quad E = SV - PV \quad \text{или} \quad E = PV - SV$$

- **Независимая формула и производная PV (PID\_EQ=False и PID\_DE=True)**

$$MV = K_p E + K_i \int_0^t E dt - K_d * \frac{dPV}{dt} + BIAS \quad E = SV - PV$$

Или

$$MV = K_p E + K_i \int_0^t E dt + K_d * \frac{dPV}{dt} + BIAS \quad E = PV - SV$$

- **Зависимая формула и производная E (PID\_EQ=True и PID\_DE=False)**

$$MV = K_c \left[ E + \frac{1}{T_i} \int_0^t E dt + T_d * \frac{dE}{dt} \right] + BIAS \quad E = SV - PV \quad \text{или} \quad E = PV - SV$$

- **Зависимая формула и производная PV (PID\_EQ=True и PID\_DE=True)**

$$MV = K_c \left[ E + \frac{1}{T_i} \int_0^t E dt - T_d * \frac{dE}{dt} \right] + BIAS \quad E = SV - PV$$

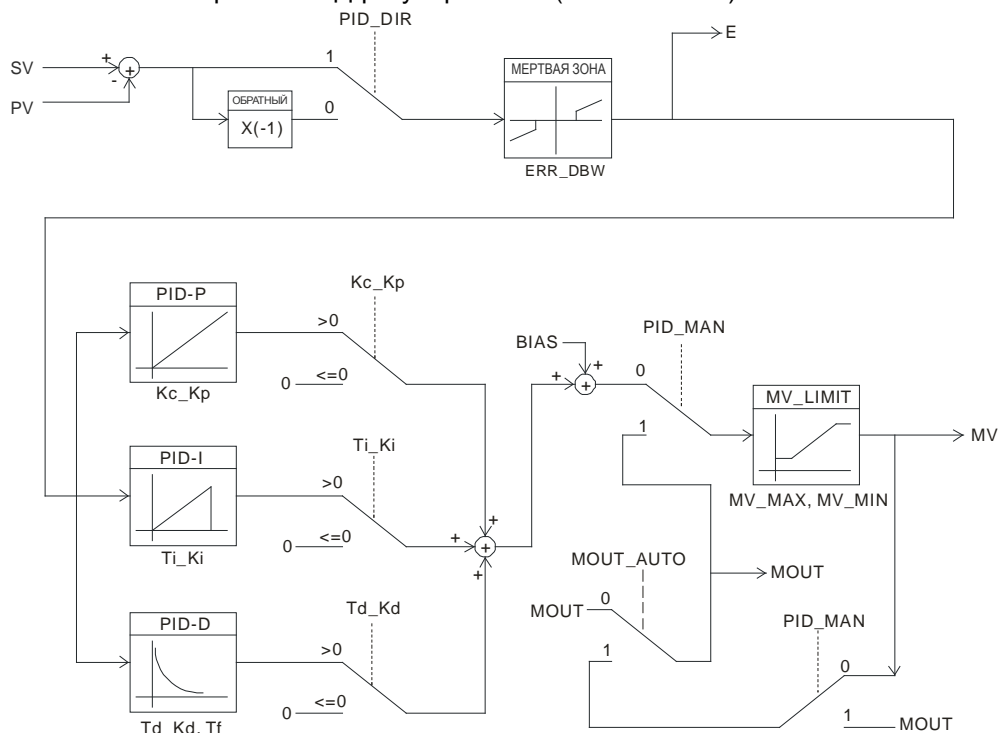
Или

$$MV = K_c \left[ E + \frac{1}{T_i} \int_0^t E dt + T_d * \frac{dE}{dt} \right] + BIAS \quad E = PV - SV$$

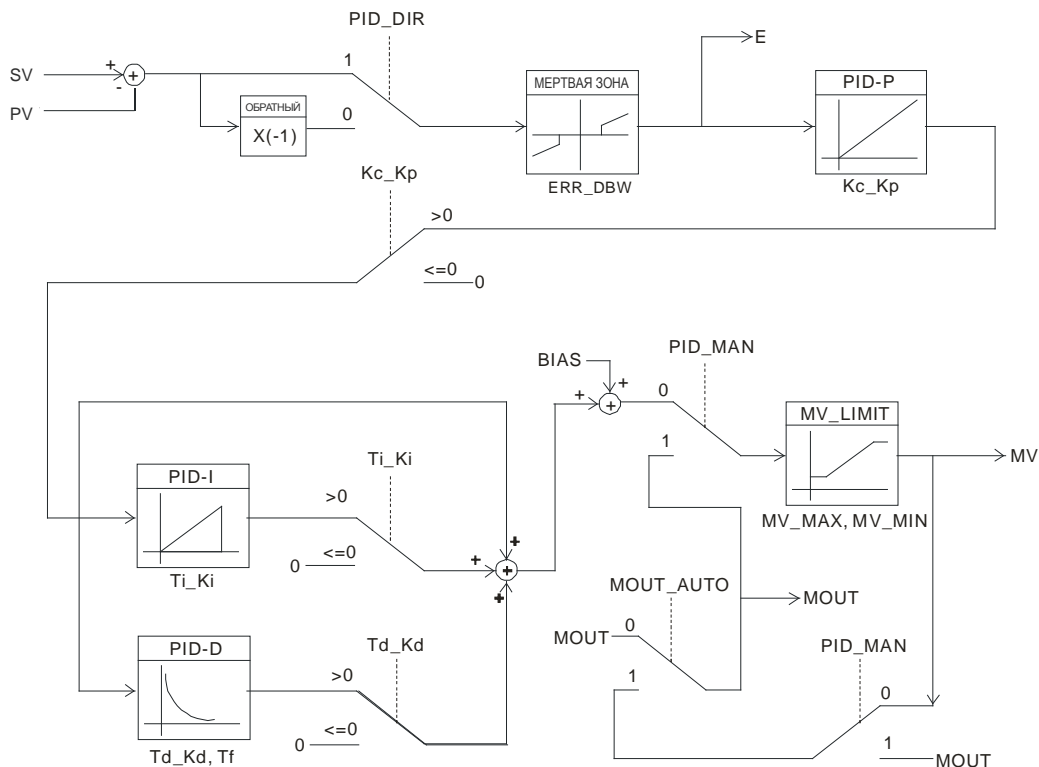
2. Когда операнд **PID\_MODE = 1**, запускается автоматическая настройка параметров ПИД-регулирования. После настройки параметров операнд **PID\_MODE** становится равным 0 и ПИД-регулирование начинает работать в автоматическом режиме.

Блок-схема алгоритма ПИД-регулирования:

Блок схема алгоритма ПИД-регулирования (независимый)



Блок схема алгоритма ПИД-регулирования (зависимый)



**Предложения:**

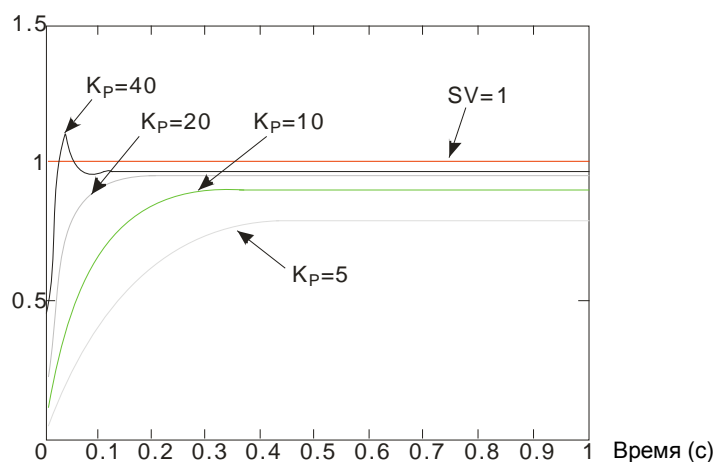
1. Учитывая, что инструкция DPID может применяться для решения широкого круга задач регулирования, можно подобрать соответствующие параметры. Например, во избежание ошибок регулирования работающего от двигателей оборудования не рекомендуется использовать данную инструкцию с операндом **PID\_MODE=1**.
2. Если настраиваются параметры **Kc\_Kp**, **Ti\_Ki** и **Td\_Kd** (**PID\_MODE = 0**), первым настраивается параметр **KP** (по опыту), а затем **Ti\_Ki** и **Td\_Kd** выставляются равными 0. После стабилизации регулирования можно увеличить **Ti\_Ki** и **Td\_Kd**. Если **Kc\_Kp = 1**, это значит, что пропорциональная составляющая 100 %. В этом случае величина погрешности увеличивается на единицу. Если пропорциональная составляющая меньше 100 %, величина погрешности уменьшается. Если пропорциональная составляющая больше 100 %, величина погрешности возрастает.
3. Чтобы значения автоматически настроенных параметров не скидывались после отключения электропитания контроллера, нужно записать их в регистры фиксированных данных, выставив **PID\_MODE = 1**. Автоматически настроенные параметры необязательно в точности подойдут для поставленной задачи регулирования. Поэтому автоматически настроенные параметры можно самостоятельно подстроить. Но при этом рекомендуется подстраивать только параметры **Ti\_Ki** и **Td\_Kd**.
4. Для правильной работы инструкции нужно настроить много параметров. Во избежание ошибок регулирования настоятельно не рекомендуется настраивать параметры произвольно.

**Пример 1:** Порядок настройки параметров инструкции ПИД-регулирования

Предположим, что функция переноса в оборудовании первого порядка, текущее значение  $G(s) = \frac{b}{s+a}$  (SV)

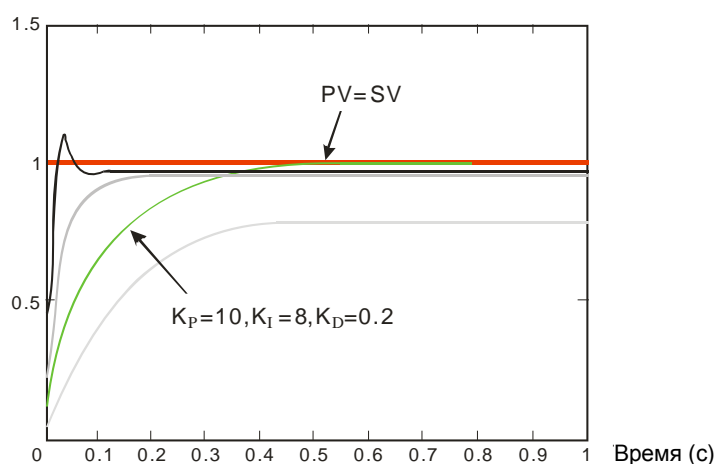
равно 1, а время опроса ( $T_s$ ) 10 мс. Рекомендуемый порядок настройки параметров в данном случае следующий.

**Шаг 1:** Сначала параметры  $K_I$  и  $K_D$  выставляются равными 0. Далее параметр  $K_P$  по очереди выставляется равным 5, 10, 20 и 40, и при каждом варианте записываются заданное и текущее значения. Результаты приведены на графике ниже.



**Шаг 2:** Когда  $K_P = 40$ , происходит слишком сильный скачок значения. Поэтому такой  $K_P$  не подходит. Когда  $K_P = 20$ , кривая изменения текущего значения (PV) близка к заданному значению (SV) и резкого скачка нет. Но из-за быстрого запуска выходное значение (MV) в момент перехода слишком большое. Поэтому такой  $K_P$  не подходит. Когда  $K_P = 10$ , кривая текущего значения (PV) плавно приближается к заданному значению (SV). Поэтому такой  $K_P$  подойдет. Когда  $K_P = 5$ , изменение значения получается недостаточным. Поэтому такой  $K_P$  не подходит.

**Шаг 3:** Выбрав оптимальный вариант  $K_P$ , равный 10, увеличьте значение параметра  $K_I$ . Например, по очереди выставьте  $K_I = 1, 2, 4$  и  $8$ . Значение параметра  $K_I$  не должно быть больше значения параметра  $K_P$ . Далее увеличьте значение параметра  $K_D$ . Например, по очереди выставьте  $K_D = 0.01, 0.05, 0.1$  и  $0.2$ . Значение параметра  $K_D$  не должно быть больше 10 % значения параметра  $K_P$ . Полученные в результате кривые текущего (PV) и заданного (SV) значений показаны ниже.



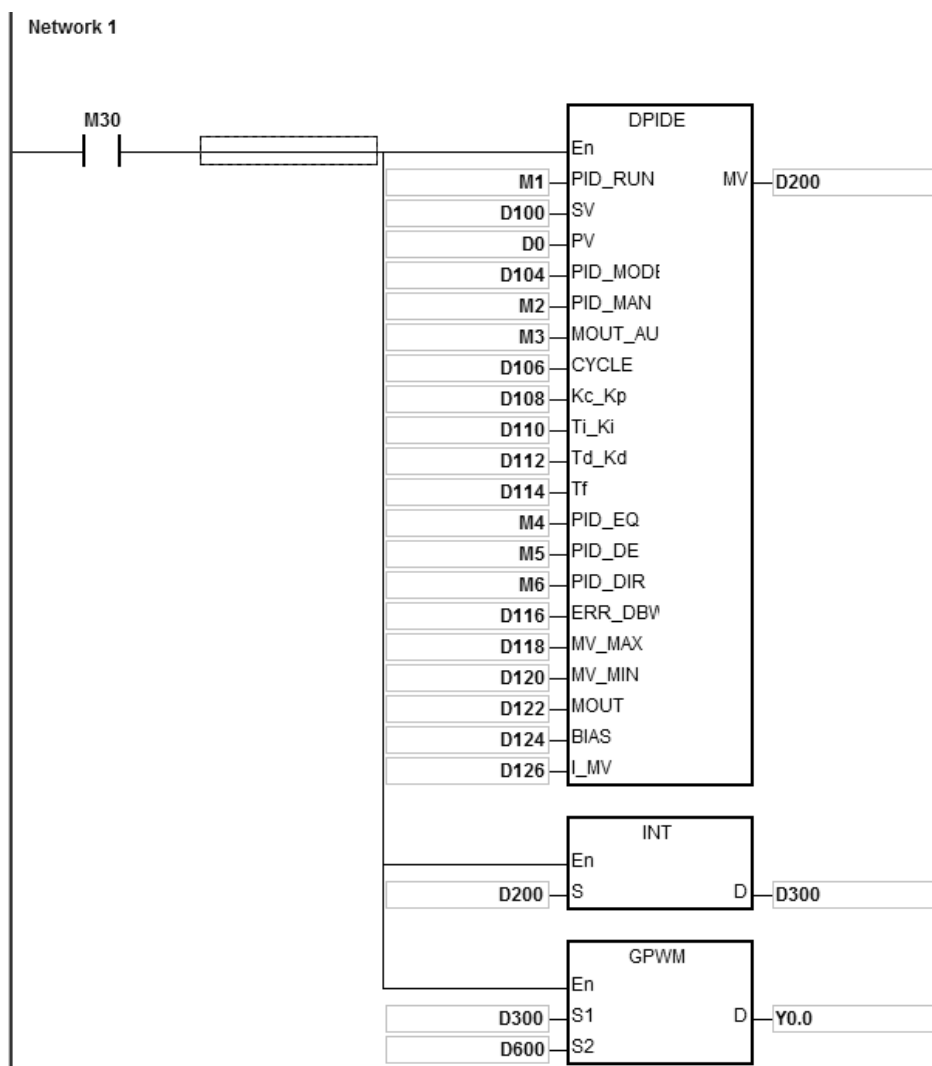
Примечание: Данный пример приведен просто в качестве справки. В каждом отдельном случае параметры нужно настраивать в зависимости от конкретной системы управления.

**Пример 2:** Применение функции автоматической настройки для регулирования температуры

Цель: При помощи функции автоматической настройки вычислить наиболее подходящие настройки алгоритма ПИД-регулирования температуры

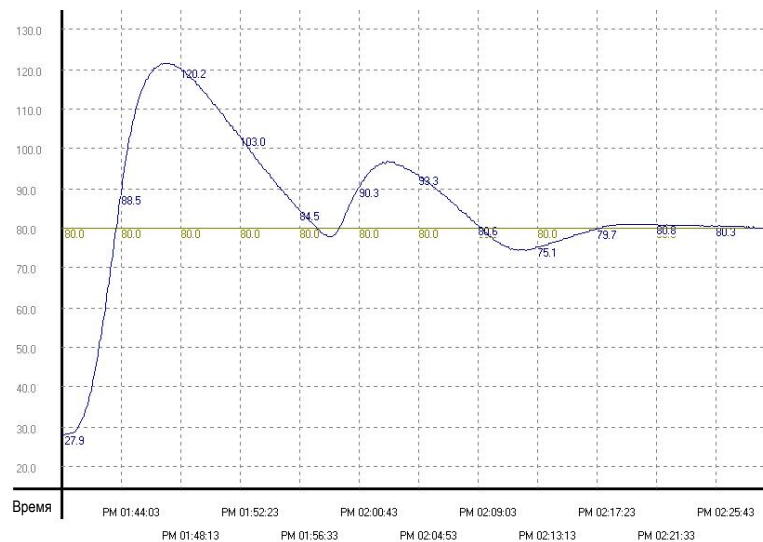
**Описание:**

Учитывая тот факт, что при первой настройке ПИД-регулирования температуры характеристики оборудования могут быть неизвестны, оптимально будет запустить автоматическую настройку параметров (**PID\_MODE = 1**). После настройки параметров операнд **PID\_MODE** становится равным 0. В данном случае в качестве оборудования берется печь. Пример настройки параметров приведен ниже.



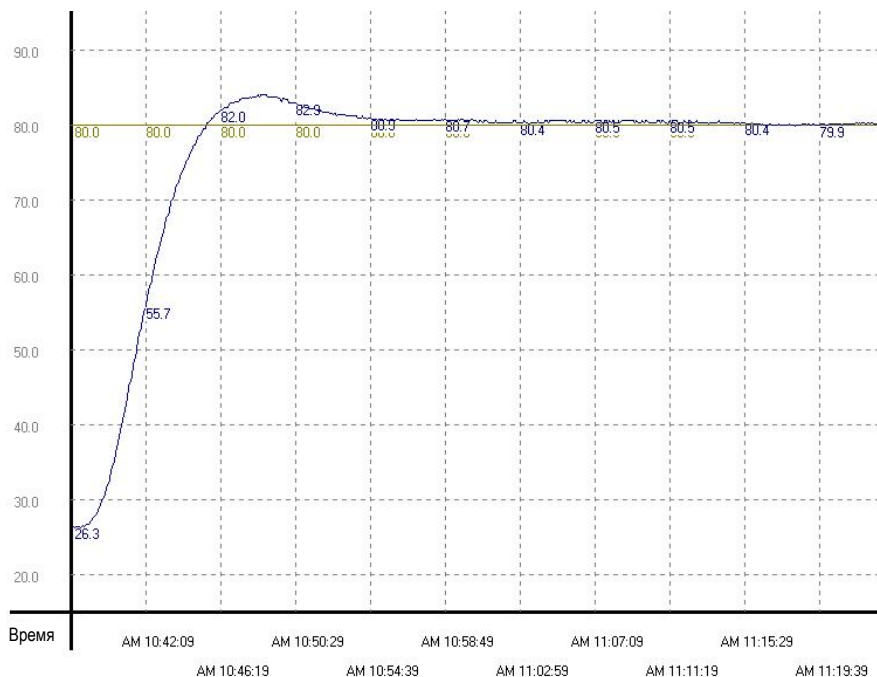
6

Экспериментальный результат автоматической настройки параметров регулирования приведен ниже.

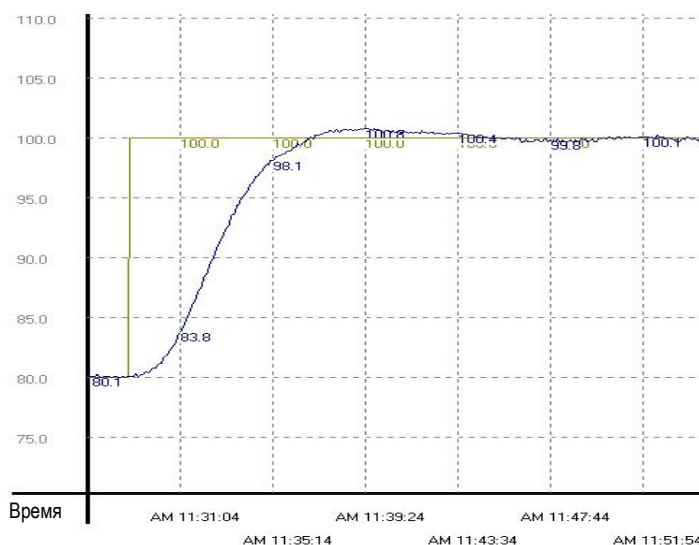




Экспериментальный результат работы ПИД-регулирования температуры с автоматически настроенными параметрами приведен ниже.



По графику выше видно, что после автоматической настройки параметров результат регулирования температуры положительный. Для выхода и поддержания заданной температуры достаточно примерно 20 мин. Когда заданная температура меняется с 80 °C на 100 °C, результат см. ниже.



По графику выше видно, что когда заданная температура меняется с 80 °C на 100 °C, ранее автоматически настроенные параметры все еще годятся для ее регулирования. При этом не требуется много времени для выхода и поддержания данной температуры.

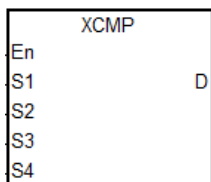
API	Код инструкции			Операнд								Функция					
0709		XCMP		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, D</b>								Сравнение входных сигналов от нескольких рабочих станций					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>	○															
S <sub>2</sub>							○									
S <sub>3</sub>								○								
S <sub>4</sub>								○								
D								○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>	●												
S <sub>2</sub>												●	
S <sub>3</sub>		●			●	●							
S <sub>4</sub>			●				●						
D			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- S<sub>1</sub>** : Тип триггерного входа
- S<sub>2</sub>** : Количество высокоскоростных счетчиков
- S<sub>3</sub>** : Количество рабочих станций и объектов
- S<sub>4</sub>** : Контрольное значение для сравнения и погрешности наблюдения
- D** : Начальный соответствующий регистр для результата сравнения в области стека

**Описание:**

1. Данная инструкция подходит только для ПЛК серии AS с микропрограммным обеспечением версии 1.04 и выше.
2. В операнде **S<sub>1</sub>** задается тип триггерного входа; для входов X0.0~X0.15 это высокочастотные входы, а для всех остальных – общего типа. При выполнении инструкции включается внешнее прерывание для входов (X0.0~X0.15). Поэтому не рекомендуется использовать входы с прерываниями, иначе при выполнении инструкции прерывания будут отключены и возобновятся только по завершении ее исполнения. Если вход общего типа, прерывания будут зависеть от времени опроса, хотя они подходят для задач, где входные сигналы не такие стабильные.



3. Операнд  $S_2$  работает с 32-битными счетчиками (HC0–HC255). Если входы типа высокочастотный триггерный вход, рекомендуется использовать аппаратный высокоскоростной счетчик и запускать его инструкцией DCNT. Если нужен высокоскоростной выход, можно инструкцией DMOV копировать текущее состояние выхода, например, копировать ось SR460 в HC0 (DMOV SR460 HC0).
4. Операнд  $S_3$  занимает три 16-битных регистра подряд.  $S_{3+0}$  – это количество рабочих станций, а  $S_{3+1}$  – это максимальное количество объектов.  $S_{3+2}$  – это результат фильтруемого объекта. У операндов  $n$  и  $m$  диапазон значений от 1 до 32. Когда значение вне данного диапазона, оно берется как максимальное (32) или минимальное (1). У  $S_{3+2}$  (количество фильтров) диапазон значений от 0 до 32767. Если значение меньше 0, оно считается равным 0, а если значение равно 0, значит функция фильтра выключена. Рекомендуется объявлять массив из 3 слов или 3 переменных типа word подряд.
5. Рекомендуется выставлять максимальное значение для  $S_{3+1}$  ( $m$ ). Если  $m < n$ , нужно проверить объекты и убедиться, что их достаточно на производственной линии.
6. Операнд  $S_4$  занимает подряд  $3 \times n$  32-битных регистра ( $6 \times n$  16-битных регистров). Если занимаемое пространство больше диапазона значений регистра  $D$ , инструкция не выполняется. Операнд  $n$  – это количество рабочих станций, заданное в операнде  $S_3$ . Функции каждого регистра и соответствующее значение операнда  $S_4$  перечислены ниже. Рекомендуется объявлять массив из  $3n$  двойных слов или 3 переменных типа double word подряд.

Функция	Рабочая станция 1	Рабочая станция 2	...	Рабочая станция $n$
Контрольное значение для сравнения (32 бита)	$S_{4+0}$	$S_{4+2}$	...	$S_{4+(n-1) \times 2}$
Погрешность наблюдения на входе (32 бита)	$S_{4+2 \times n}$	$S_{4+2 \times n + 2}$	...	$S_{4+(2 \times n - 1) \times 2}$
Погрешность наблюдения на выходе (32 бита)	$S_{4+4 \times n}$	$S_{4+4 \times n + 2}$	...	$S_{4+(4 \times n - 1) \times 2}$

Если контрольное значение для любой станции задано равным 0, эта станция перестает работать. Таким способом можно управлять рабочей станцией.

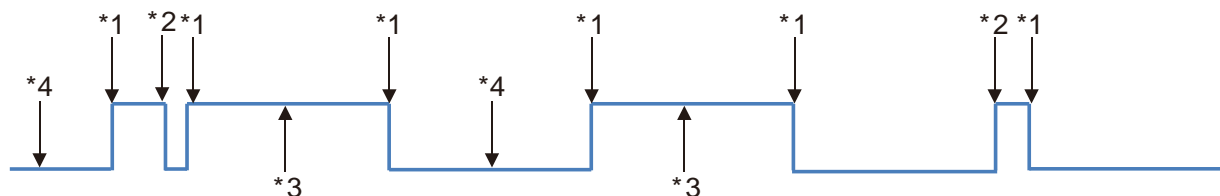
7. Операнд  $D$  – это начальный соответствующий регистр для результатов сравнения в области стека. Операнд  $D$  занимает подряд  $2 \times n$  16-битных регистров и  $2 \times m \times n$  32-битных регистров (или  $4 \times m \times n$  16-битных регистров). Если занимаемое пространство больше диапазона значений регистра  $D$ , инструкция не выполняется. Функции каждого регистра и соответствующее значение операнда  $D$  перечислены ниже.

Функция	Рабочая станция 1	Рабочая станция 2	...	Рабочая станция $n$
Значение головного индекса (16 бит)	$D+0$	$D+1$	...	$D+(n-1)$
Значение хвостового индекса (16 бит)	$D+n$	$D+(n+1)$	...	$D+(2 \times n - 1)$
Результат сравнения счетчика 1 объектов на входе (32 бита)	$D+2 \times n$	$D+2 \times n + 2$	...	$D+2 \times n + 2(n-1)$
Результат сравнения счетчика 1 объектов на выходе (32 бита)	$D+4 \times n$	$D+4 \times n + 2$	...	$D+4 \times n + 2(n-1)$

Функция	Рабочая станция 1	Рабочая станция 2	...	Рабочая станция n
:	:	:	:	:
Результат сравнения счетчика m объектов на входе (32 бита)	$D+4mxn-2$ $xn$		...	$D+4mxn-2$
Результат сравнения счетчика m объектов на выходе (32 бита)	$D+4mxn$		...	$D+4mxn+2$ (n-1)

Получается, что занимает больше места в области стека. Если занимаемое место больше диапазона значений регистра D, контроллер исполняет только ту часть, что является действительной в области хранения, и предупреждения при этом не выдается. Рекомендуется объявлять массив из  $2xn+4mxn$  слов.

8. Ограничений по количеству исполнений инструкции не существует, но в отдельно взятый момент времени может исполняться только одна инструкция.
9. Предполагается, что данная инструкция будет использоваться вместе с инструкцией API0710 YOUT и у этих инструкций будет одинаковое Начальный соответствующий регистр для результата сравнения в области стека (операнд D).
10. Ниже показан принцип работы высокоскоростного счетчика и фильтра по времени (чтение идет справа налево).



- \*1. ПЛК считывает текущее значение счетчика.
- \*2. Сброс значения счетчика: количество считанных фильтров меньше количества заданных фильтров.
- \*3. Запись значения счетчика: уровень сигнала высокий (есть импульс), и значение счетчика записывается в область стека сравнения на входе.
- \*4. Запись значения счетчика: уровень сигнала низкий (нет импульса), и значение счетчика записывается в область стека сравнения на выходе.

11. Когда регистрируется передний/задний фронт импульса и фильтр завершает обработку, контроллер считывает значение высокоскоростного счетчика и добавляет единицу к значению головного индекса. Контроллер записывает результаты счетчиков на входе и выходе каждой рабочей станции. Результат сравнения счетчика = текущее значение счетчика + контрольное значение + погрешность наблюдения. Единица добавляется в головной индекс независимо от зарегистрированного фронта импульса – переднего или заднего. Максимальное значение головного индекса –  $mx2$  (максимальное количество объектов).

12. Значение головного индекса циклически увеличивается, когда регистрируется передний/задний фронт импульса и завершается обработка фильтрами (по умолчанию триггерный вход OFF). Максимальное значение головного индекса –  $mx2$  (максимальное количество объектов). Например, количество объектов задано равным 10, значение головного индекса (по умолчанию: 0) будет увеличиваться до 1, 2, 3 и так до 20, а затем снова до 1, 2, 3 и до 20. Если значение головного индекса равно 0, это значит, что после исполнения инструкции ни один объект не вошел на линию. После добавления единицы к значению головного индекса контроллер проверяет значение хвостового индекса. Если значение (после добавления единицы) головного индекса равняется значению хвостового индекса, добавление единицы отменяется, и результат счетчика записывается.
13. Если во время исполнения инструкции состояние начального = **OFF**, **триггер переднего фронта импульса** будет соответствовать **нечетным значениям** головного индекса, а **триггер заднего фронта импульса** будет соответствовать **четным значениям** головного индекса.
14. Если во время исполнения инструкции состояние начального = **ON**, **триггер заднего фронта импульса** будет соответствовать **нечетным значениям** головного индекса, а **триггер переднего фронта импульса** будет соответствовать **четным значениям** головного индекса.
15. Во время исполнения инструкции значения в области накопления и области индекса не стираются. Если данные в фиксированной области и необходимо разрешить снова, нужно при помощи инструкции ZRST стереть значения головного и хвостового индексов.

**6**

**Пример:**

Подробнее см. пример в инструкции API0710 YOUT.

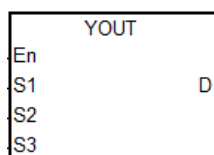
API	Код инструкции			Операнд								Функция					
0710		YOUT		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D</b>								Сравнение выходных сигналов нескольких рабочих станций					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>							○									
S <sub>2</sub>								○								
S <sub>3</sub>								○								
D		○	○													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>												●	
S <sub>2</sub>		●			●	●							
S <sub>3</sub>			●				●						
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- S<sub>1</sub>** : Количество высокоскоростных счетчиков
- S<sub>2</sub>** : Количество рабочих станций и объектов
- S<sub>3</sub>** : Начальный соответствующий регистр для результата сравнения в области стека
- D** : Начальный соответствующий регистр для выхода рабочей станции

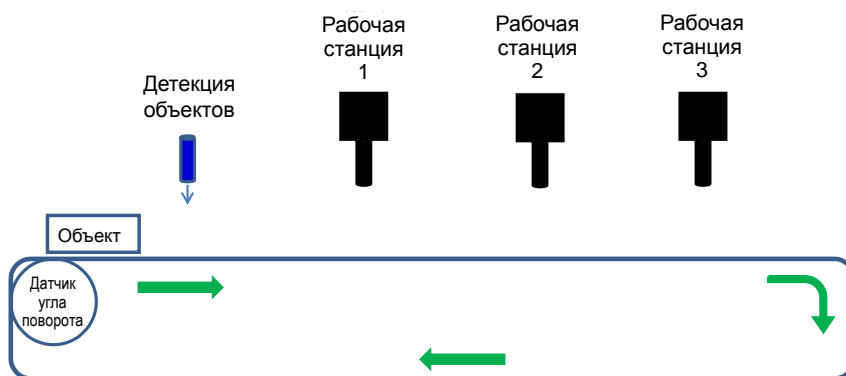
**Описание:**

1. Данная инструкция подходит только для ПЛК серии AS с микропрограммным обеспечением версии 1.04 и выше.
2. В операнде **S<sub>1</sub>** задается высокоскоростной счетчик. Высокоскоростной счетчик должен быть таким же, что и высокоскоростной счетчик, заданный в инструкции API0709 XCMP.
3. Операнд **S<sub>2</sub>** занимает два 16-битных регистра подряд. **S<sub>2</sub>+0** – это количество рабочих станций, а **S<sub>2</sub>+1** – это максимальное количество объектов. У операндов n и m диапазон значений от 1 до 32. Когда значение вне данного диапазона, оно берется как максимальное (32) или минимальное (1). Эти настройки должны быть одинаковыми с настройками инструкции API0709 XCMP.
4. Операнд **S<sub>3</sub>** – это начальный соответствующий регистр для результатов сравнения в области стека. Операнд **S<sub>3</sub>** занимает подряд 2хn 16-битных регистров и 2хmхn 32-битных регистров (или 4хmхn 16-битных регистров). Функции каждого регистра и соответствующее значение операнда **D** см. в инструкции API0709 XCMP. Предлагается использовать одинаковую переменную с инструкцией API0709 XCMP.
5. Ограничений по количеству исполнений инструкции не существует, но в отдельно взятый момент времени может исполняться только одна инструкция.

6. Предполагается, что данная инструкция будет использоваться вместе с инструкцией API0709 XCMP и у этих инструкций будет одинаковое Начальный соответствующий регистр для результата сравнения в области стека (операнд S<sub>3</sub>).
7. Операнд D предназначен только для выходов Y, регистров M и данных типа BOOL. Он занимает xp рабочих станций подряд. Если он используется как выход или регистр M, инструкция запускается для обновления состояния выходов.
8. Соответствующие нечетные значения головного индекса (например, 1, 3, 5,...) называются результатом сравнения счетчика объектов на входе. Соответствующие четные значения головного индекса (например, 2, 4, 6,...) называются результатом сравнения счетчика объектов на выходе.
9. Когда результат сравнения счетчиков на входе и выходе в области стека равен 0, действия в этой области выполняться не будут, и состояние соответствующего выхода рабочей станции будет OFF. К значению хвостового индекса добавляется 2, но это значение (после добавления 2) не будет превышать значение головного индекса.
10. Когда исполняется инструкция YOUT, каждая рабочая станция проверяет результат сравнения на входе и выходе в хвостовом индексе. Если значение счетчика больше или равно значению сравнения на входе, соответствующий выход переходит в состояние ON и к значению хвостового индекса добавляется единица. Если значение счетчика больше или равно результату сравнения на выходе, соответствующий выход переходит в состояние OFF и единица добавляется к значению хвостового индекса, но это значение (после добавления единицы) не будет превышать значения головного индекса.

6

**Пример: Пример трех рабочих станций и количество объектов до 4 шт.**



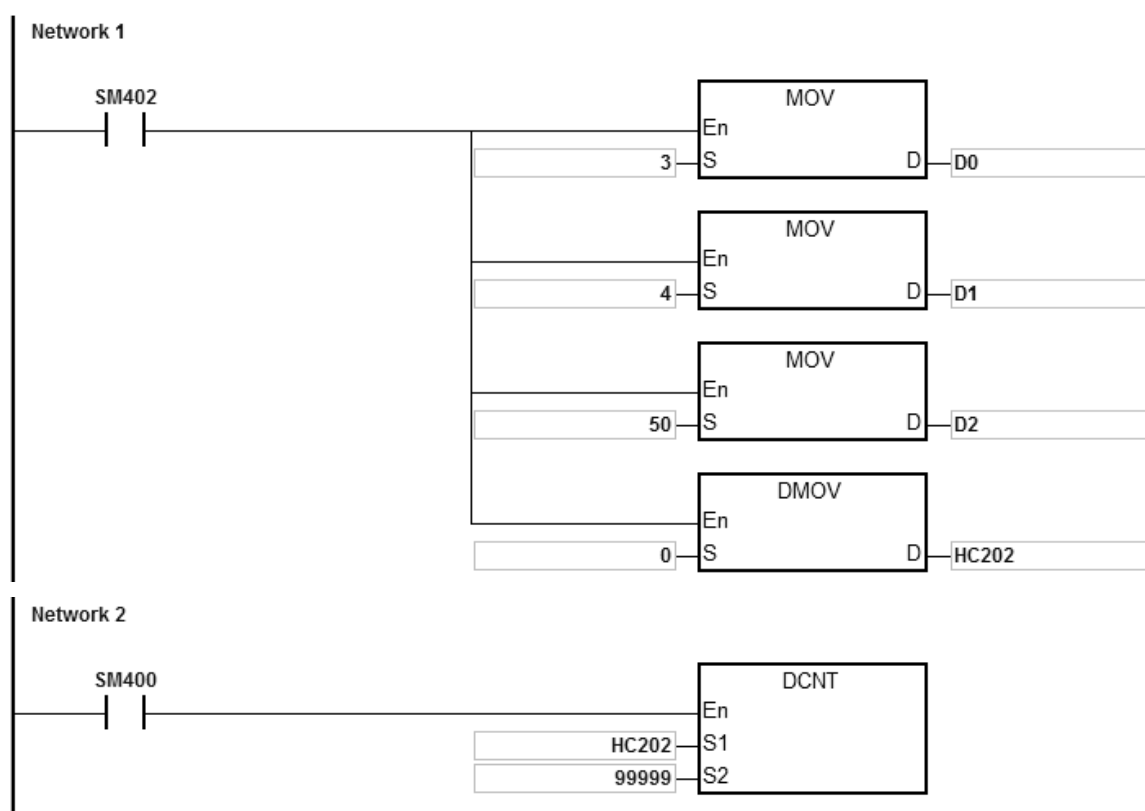
**Шаг 1:** Вход X0.4 будет использоваться для прерываний детекции объектов, HC202 – это высокоскоростной счетчик ДУП, а выход Y0.0 – это начальный выход рабочей станции.

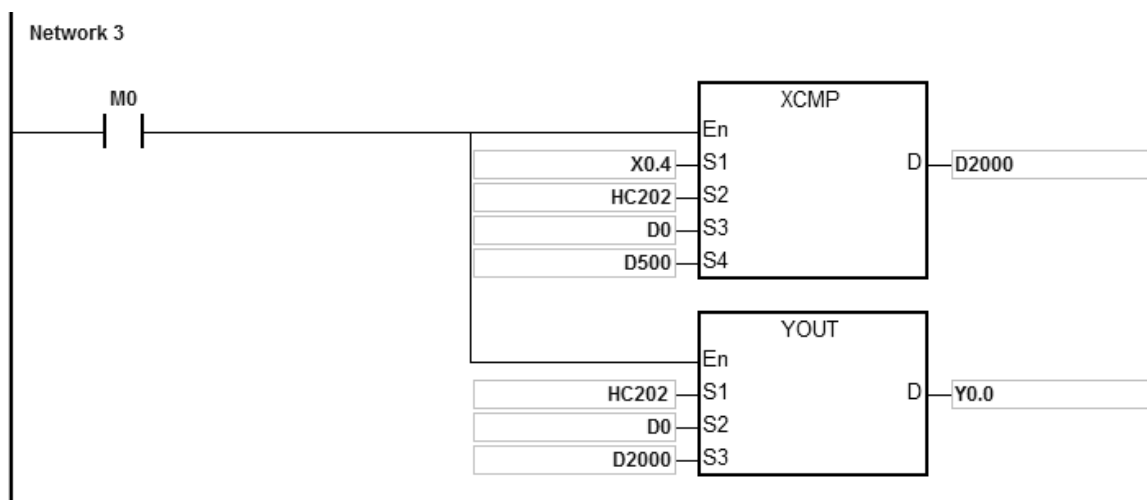
**Шаг 2:** При помощи регистра для редактирования изменим контрольное значение, погрешность наблюдения на входе и выходе.

Регистр D	D500	D502	D504
Контрольное значение для сравнения (32 бита)	K2000	K3000	K4000

Регистр D	D506	D508	D510
Погрешность наблюдения на входе (32 бита)	K100	K120	K130
Регистр D	D512	D514	D516
Погрешность наблюдения на выходе (32 бита)	K50	K-20	K20
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K0	K0	K0
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K0	K0	K0

**Шаг 3: Ввод начальных значений и запись программ**





Запишем 3 рабочих станции в D0, 4 объекта в D1 и 50 фильтров в D2. После срабатывания контакта M0 можно настроить детекцию объектов, значение для сравнения, результат сравнения счетчиков на входе и выходе, а также выходы каждой рабочей станции. Например, система обнаружила 2 объекта на входе производственной линии и есть 4 триггера для записи результатов сравнения счетчиков – 3000, 3500, 4500 и 5000 – в HC202 (HC202=K5060). Значение сравнения и головной/хвостовой индексы в области стека приведены ниже.

Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K1	K1	K1
Регистр D	D2006	D2008	D2010
Результат сравнения счетчиков 1 объектов на входе (32 бита)	K5100	K6120	K7130
Регистр D	D2012	D2014	D2016
Результат сравнения счетчиков 1 объектов на выходе (32 бита)	K5550	K6480	K7520
Регистр D	D2018	D2020	D2022
Результат сравнения счетчиков 2 объектов на входе (32 бита)	K6600	K7620	K8630
Регистр D	D2024	D2026	D2028
Результат сравнения счетчиков 2 объектов на выходе (32 бита)	K7050	K7980	K9020

Регистр D	D2030	D2032	D2034
Результат сравнения счетчиков 3 объектов на входе (32 бита)	K0	K0	K0
Регистр D	D2036	D2038	D2040
Результат сравнения счетчиков 3 объектов на выходе (32 бита)	K0	K0	K0

Когда высокоскоростной счетчик HC202 доходит до 5200, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
16-битное значение	ON	OFF	OFF
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K2	K1	K1

Когда высокоскоростной счетчик HC202 доходит до 6200, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
16-битное значение	OFF	ON	OFF
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K3	K2	K1

Когда высокоскоростной счетчик HC202 доходит до 6800, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
---------	------	------	------



16-битное значение	ON	OFF	OFF
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K4	K3	K1

Когда высокоскоростной счетчик HC202 доходит до 7300, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
16-битное значение	OFF	OFF	ON
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K4	K3	K2

Когда высокоскоростной счетчик HC202 доходит до 7700, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
16-битное значение	OFF	ON	OFF
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K4	K4	K3

Когда высокоскоростной счетчик HC202 доходит до 8000, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
Состояние выхода	OFF	OFF	OFF
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K4	K4	K3

Когда высокоскоростной счетчик HC202 доходит до 8700, состояние выхода Y становится следующим:

Выход Y	Y0.0	Y0.1	Y0.2
Состояние выхода	OFF	OFF	ON
Регистр D	D2000	D2001	D2002
Значение головного индекса (16 бит)	K4	K4	K4
Регистр D	D2003	D2004	D2005
Значение хвостового индекса (16 бит)	K4	K4	K4

API	Код инструкции			Операнд								Функция				
0711		YOUT		S <sub>1</sub> , S <sub>2</sub> , S <sub>3</sub> , D								Задание времени восхода и захода солнца				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
Longi								●								○
Lati								●								○
TimeZ								●					○			
DST	●	●	●	●												
Year								●					○			
Month								●					○			
Date								●					○			
RHour								●								
RMin								●								
RSec								●								
SHour								●								
SMin								●								
SSEC								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Longi									●				
Lati									●				
TimeZ		●				●							
DST	●												
Year		●			●	●							
Month		●			●	●							
Date		●			●	●							
RHour		●			●	●							
RMin		●			●	●							
RSec		●			●	●							
SHour		●			●	●							
SMin		●			●	●							
SSEC		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:

SUNRS	
EN	
Longi	RHour
Lati	RMin
TimeZ	RSec
DST	SHour
Year	SMin
Month	SSEC
Date	

**Longi** : Долгота (тип REAL)

**Lati** : Широта (тип REAL)

**TimeZ** : Часовой пояс (целое число) (-12 ~ +14)

**DST** : Летнее время

**Year** : Год

**Month** : Месяц

<b>Date</b>	: Дата
<b>RHour</b>	: Час восхода солнца в установленную дату (24-часовой формат)
<b>RMin</b>	: Минута восхода солнца в установленную дату
<b>RSec</b>	: Секунда восхода солнца в установленную дату
<b>SHour</b>	: Час захода солнца в установленную дату (24-часовой формат)
<b>SMin</b>	: Минута захода солнца в установленную дату
<b>SSec</b>	: Секунда захода солнца в установленную дату

**Описание:**

1. Инструкция работает с прошивкой версии V1.04.50 или выше. Время восхода и захода солнца может быть не таким точным, как определяется местными органами прогодных прогнозов, поскольку введенные вами значения могут быть неправильными или высота места установки устройства может снижать точность. Когда результат не удовлетворяет точности, вы можете отрегулировать значения вручную. После самодиагностики диапазон ошибок этой инструкции составляет менее 5 минут.
2. Введите значения для локальной долготы и широты в числах. Например, долгота и широта Таюань, Тайвань - 121.30098 и 24.99363. Широта севернее экватора обозначаются положительным знаком. Широта к югу от экватора обозначается отрицательными значениями.
3. Введите значения для местного часового пояса, диапазон от -12 до +14. Часовой пояс не может быть рассчитан через заданную долготу и широту; если настройка находится вне допустимого диапазона или значение неверно, будет отображаться сообщение об ошибке.
4. Когда в регионе действует летнее время, инструкция проверяет, включено ли летнее время на ПЛК. Когда на ПЛК включено летнее время, в часы восхода и захода солнца будет добавлен один час.
5. Введите значения для локальной даты, месяца и года в десятичном формате. Убедитесь, что вы ввели правильные значения. Инструкция не проверяет правильность ввода значений.
6. После вычисления команда выдает часы, минуты и секунды для восхода и захода солнца в целых числах, в десятичном формате и 24-часовом формате времени.

## 6.9 Логические инструкции

### 6.9.1 Описание логических инструкций

API	Код инструкции			Операнды								Функция				
0800	D	WAND	P	$S_1 \cdot S_2 \cdot D$								Логическое И				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

#### Символ:

WAND	WANDP
En	En
S1 D	S1 D
S2	S2

DANDP	DAND
En	En
S1 D	S1 D
S2	S2

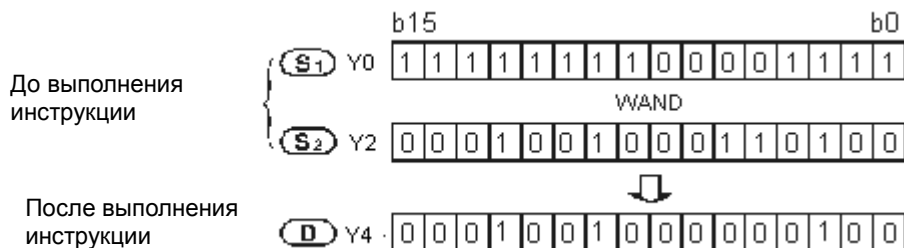
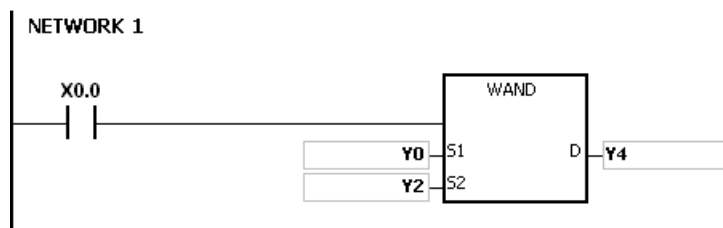
- $S_1$  : Источник данных 1
- $S_2$  : Источник данных 2
- D : Результат операции

#### Описание:

- Для значений в двоичном формате в  $S_1$  и  $S_2$  данная инструкция выполняет логическую операцию И на каждой паре соответствующих битов. Результат операции сохраняется в D.
- Только 32-битная инструкция DAND может использовать 32-битный счетчик.
- Результат по каждой позиции равен 1, если первый и второй биты равны 1. В других случаях результат равен 0.

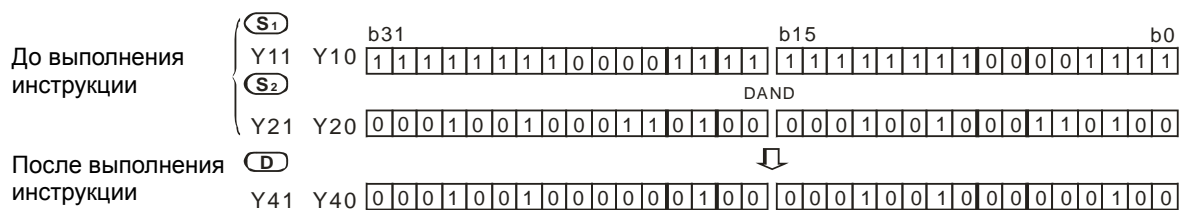
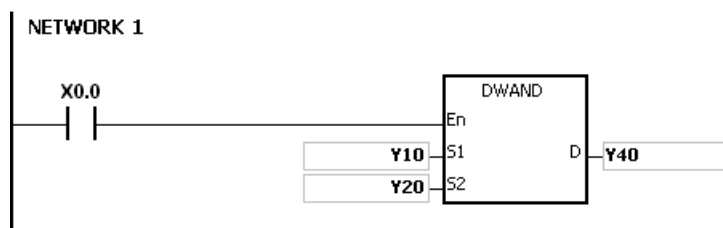
#### Пример 1:

Когда X0.0 включен, инструкция выполняет логическую операцию И на каждой паре соответствующих битов для данных в 16-битном объекте Y0 и 16-битном объекте Y2. Результат операции сохраняется в Y4.



**Пример 2:**

Когда X0.0 включен, инструкция выполняет логическую операцию И на каждой паре соответствующих битов для данных в 32-битном объекте (Y11, Y10) и 32-битном объекте (Y21, Y20). Результат операции сохраняется в (Y41, Y40).



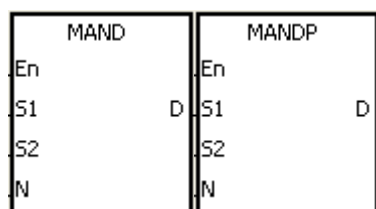
API	Код инструкции			Операнды							Функция					
0801		MAND	P	$S_1 \cdot S_2 \cdot D \cdot n$							Матричное И					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●							
$S_2$	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



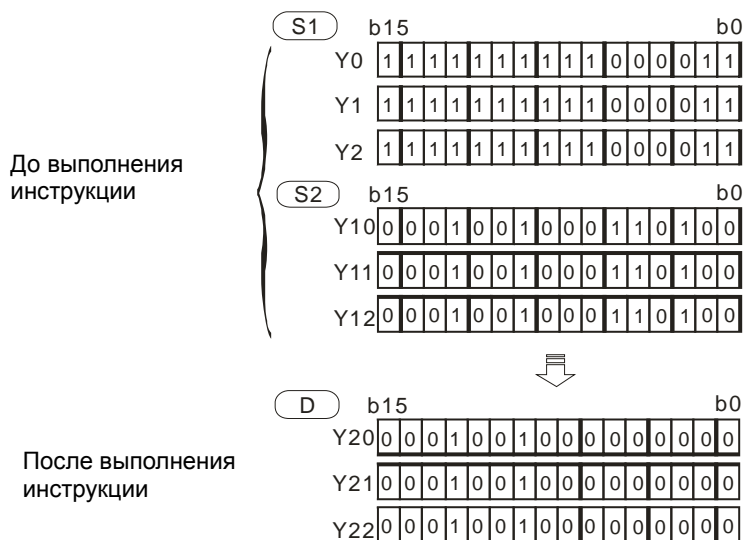
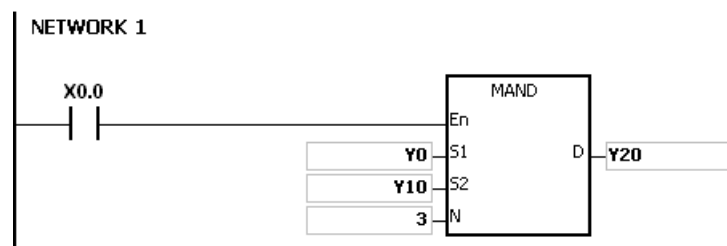
- $S_1$  : Источник матричных данных 1
- $S_2$  : Источник матричных данных 2
- D : Результат операции
- n : Длина массива

**Описание:**

- Для n строк значений в двоичном формате в  $S_1$  и n строк значений в двоичном формате в  $S_2$  данная инструкция выполняет логическую операцию И на каждой паре соответствующих битов. Результат операции сохраняется в D.
- Результат равен по каждой позиции 1, если первый и второй биты равны 1. В других случаях результат равен 0.
- Значение операнда n находится в диапазоне от 1 до 256.

**Пример:**

Когда X0.0 включен, инструкция выполняет операцию матричное И на каждой паре соответствующих битов в 16-битных объектах Y0 ~ Y2 и в 16-разрядных объектах Y10 ~ Y12. Результат операции сохраняется в 16-битных объектах Y20 ~ Y22.

**Примечание:**

- Если  $S_1+n-1$ ,  $S_2+n-1$ , or  $D+n-1$  превышают допустимый предел, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
- Если значение  $n$  меньше 1 или больше 256, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.
- Описание матричных инструкций:
  - Матрица состоит из нескольких 16-битных регистров. Число регистров в матрице - это длина массива  $n$ . В матрице имеется  $16 \times n$  бит, а матричные операции выполняются побитово.
  - Матричная инструкция определяет  $16 \times n$  бит в матрице как строку бит, а не как их значения. Матричная операция матрицы выполняется на одном указанном бите.
  - В матричной инструкции в основном обрабатывается схема «один ко многим» или статус «многие ко многим», такие как перемещение, копирование, сравнение и поиск.
  - При выполнении матричной инструкции необходим 16-битный регистр для указания определенного бита среди  $16n$  бит в матрице. Такой 16-битный регистр называется указателем и задается пользователем. Значение в регистре находится в диапазоне от 0 до  $16n-1$  и соответствует битам в диапазоне от  $b0$  до  $b16n-1$ .



- В матричной операции может быть реализован сдвиг указанных данных или вращение указанных данных. Кроме того, число бит уменьшается слева направо, как показано ниже.



- Ширина матрицы (C): 16 бит.
- Pr представляет собой указатель. Когда значение Pr равно 15, указывается бит b15.

Пример: Матрица ниже состоит из трех 16-битных объектов Y0, Y1 и Y2. Данные в Y0 равны 16 # AAAA, данные в Y1 равны 16 # 5555, а данные в Y2 равны 16 # AAFF.

C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	Y0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	Y1
1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	Y2

Пример: Матрица ниже состоит из трех 16-битных объектов X0, X1 и X2. Данные в X0 равны 16#37, данные в X1 равны 16#68 и данные в X2 равны 16#45.

C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0	
0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X0
0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X2

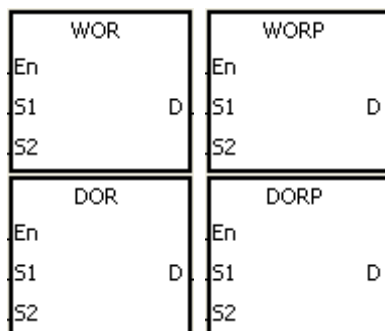
API	Код инструкции			Операнды							Функция						
0802	D	WOR	P	$S_1 \cdot S_2 \cdot D$							Логическое ИЛИ						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



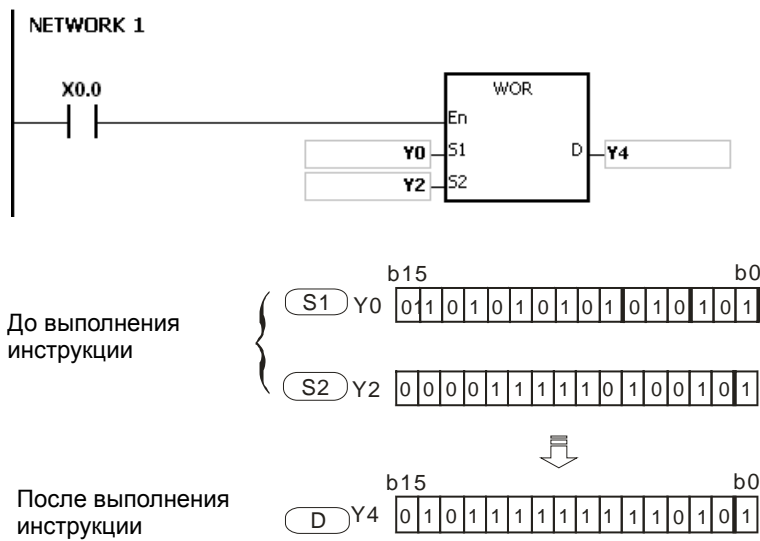
- $S_1$  : Источник данных 1
- $S_2$  : Источник данных 2
- D : Результат операции

**Описание:**

- Для значений в двоичном формате в  $S_1$  и  $S_2$  данная инструкция выполняет логическую операцию ИЛИ на каждой паре соответствующих битов. Результат операции сохраняется в D.
- Только 32-битная инструкция DWOR может использовать 32-битный счетчик.
- Результат по каждой позиции равен 1, если первый и второй биты равны 1. В других случаях результат равен 0.

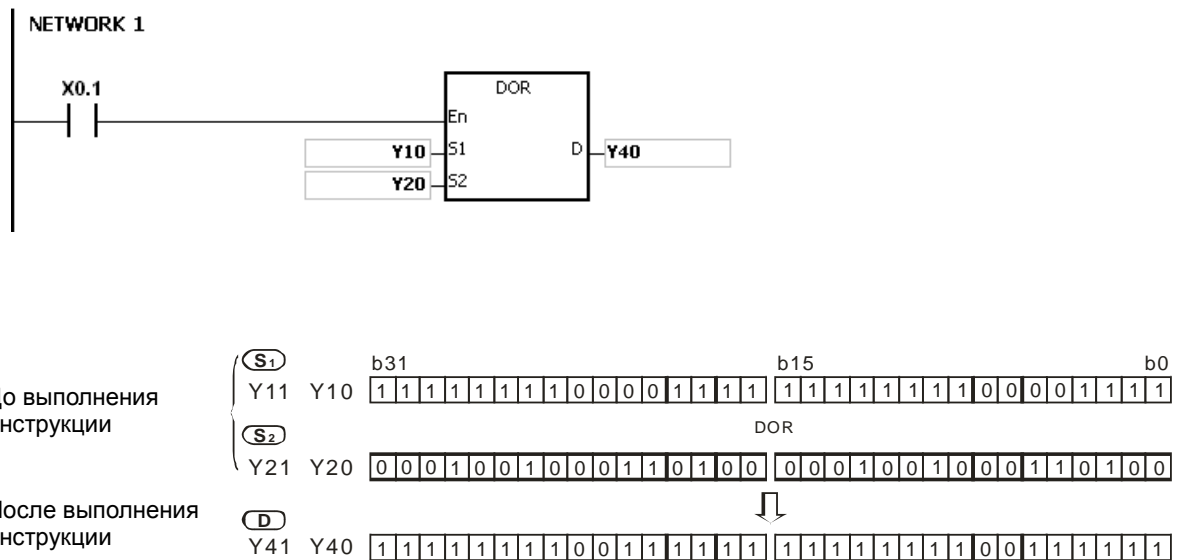
**Пример 1:**

Когда X0.0 включен, инструкция выполняет логическую операцию ИЛИ на каждой паре соответствующих битов для данных в 16-битном объекте Y0 и 16-битном объекте Y2. Результат операции сохраняется в Y4.



**Пример 2:**

Когда X0.0 включен, инструкция выполняет логическую операцию ИЛИ на каждой паре соответствующих битов для данных в 32-битном объекте (Y11, Y10) и 32-битном объекте (Y21, Y20). Результат операции сохраняется в (Y41, Y40).



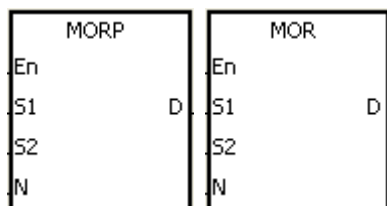
API	Код инструкции			Операнды						Функция					
0803		MOR	P	$S_1 \cdot S_2 \cdot D \cdot n$						Матричное ИЛИ					

Device	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●							
$S_2$	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●				○	○		

Data type	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



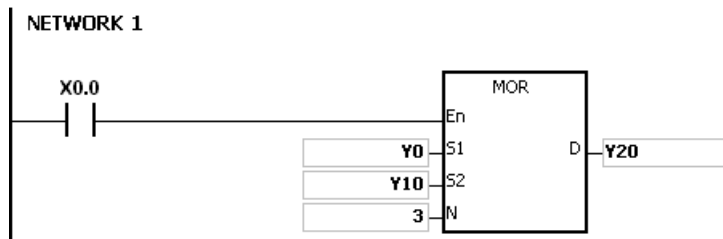
- $S_1$  : Источник матричных данных 1
- $S_2$  : Источник матричных данных 2
- D : Результат операции
- n : Длина массива

**Описание:**

- Для n строк значений в двоичном формате в  $S_1$  и n строк значений в двоичном формате в  $S_2$  данная инструкция выполняет логическую операцию матричное ИЛИ на каждой паре соответствующих битов. Результат операции сохраняется в D.
- Результат равен по каждой позиции 1, если первый и второй биты равны 1. В других случаях результат равен 0.
- Значение операнда n находится в диапазоне от 1 до 256.

**Пример:**

Когда X0.0 включен, инструкция выполняет операцию матричное ИЛИ на каждой паре соответствующих битов в 16-битных объектах Y0 ~ Y2 и в 16-разрядных объектах Y10 ~ Y12. Результат операции сохраняется в 16-битных объектах Y20 ~ Y22.



**Примечание:**

1. Если  $S_1+n-1$ ,  $S_2+n-1$ , or  $D+n-1$  превышают допустимый предел, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или больше 256, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

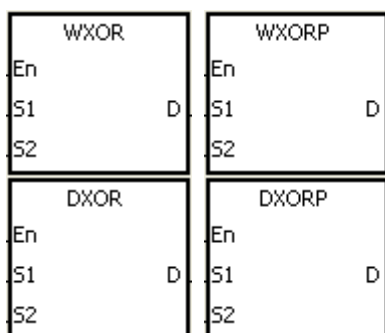
API	Код инструкции			Операнды							Функция						
0804	D	WXOR	P	$S_1 \cdot S_2 \cdot D$							Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



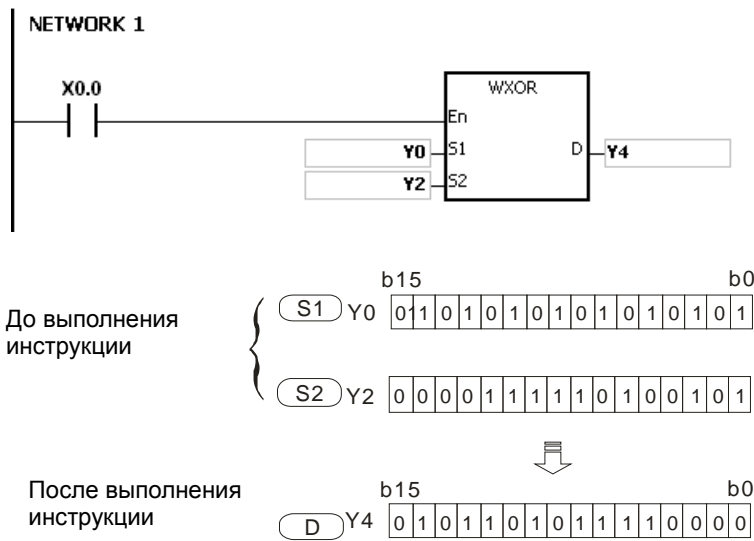
- $S_1$  : Источник данных 1
- $S_2$  : Источник данных 2
- D : Результат операции

**Описание:**

1. Для значений в двоичном формате в  $S_1$  и  $S_2$  данная инструкция выполняет логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ на каждой паре соответствующих битов. Результат операции сохраняется в D.
2. Только 32-битная инструкция DXOR может использовать 32-битный счетчик.
3. Результат по каждой позиции равен 1, если первый и второй биты равны 1. В других случаях результат равен 0.

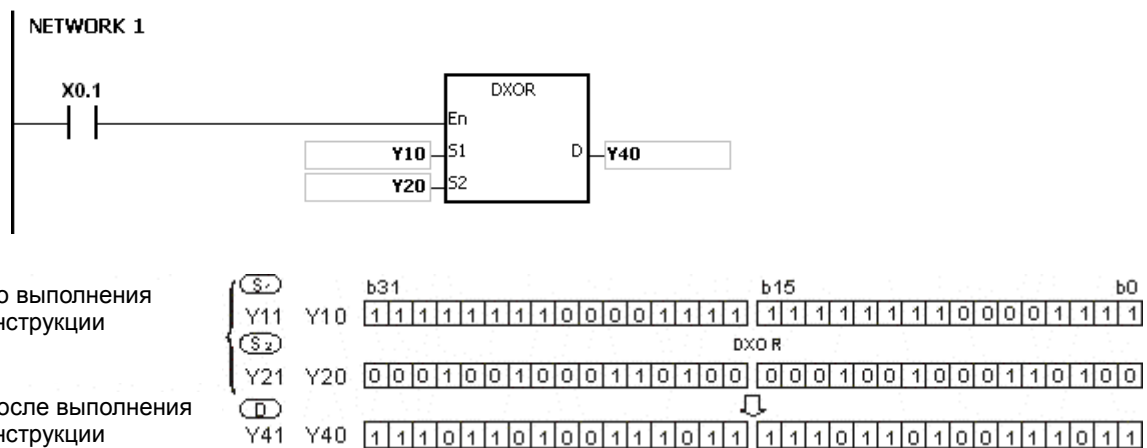
**Пример 1:**

Когда X0.0 включен, инструкция выполняет логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ на каждой паре соответствующих битов для данных в 16-битном объекте Y0 и 16-битном объекте Y2. Результат операции сохраняется в Y4.



**Пример 2:**

Когда X0.0 включен, инструкция выполняет логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ на каждой паре соответствующих битов для данных в 32-битном объекте (Y11, Y10) и 32-битном объекте (Y21, Y20). Результат операции сохраняется в (Y41, Y40).

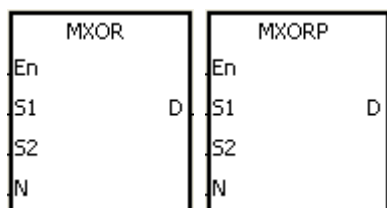


API	Код инструкции			Операнды							Функция						
0805		MXOR	P	$S_1 \cdot S_2 \cdot D \cdot n$							Матричное ИСКЛЮЧАЮЩЕЕ ИЛИ						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●							
$S_2$	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:** $S_1$  : Источник матричных данных 1 $S_2$  : Источник матричных данных 2

D : Результат операции

n : Длина массива

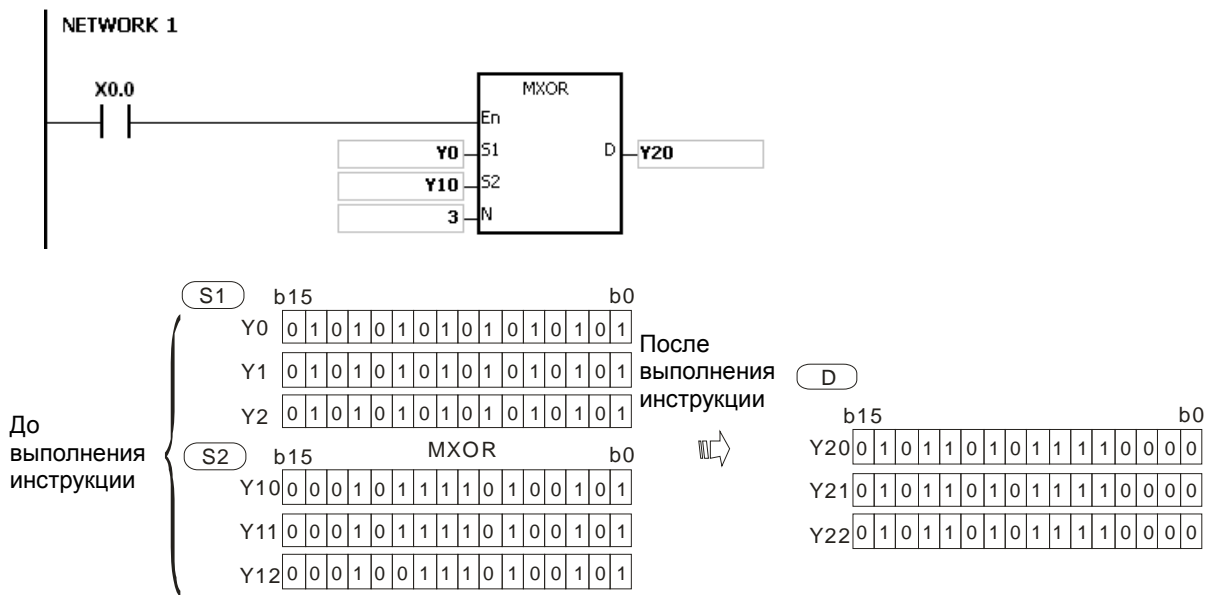
**Описание:**

- Для n строк значений в двоичном формате в  $S_1$  и n строк значений в двоичном формате в  $S_2$  данная инструкция выполняет логическую операцию матричное ИСКЛЮЧАЮЩЕЕ ИЛИ на каждой паре соответствующих битов. Результат операции сохраняется в D.
- Результат равен по каждой позиции 1, если первый и второй биты равны 1. В других случаях результат равен 0.
- Значение операнда n находится в диапазоне от 1 до 256.

**Пример:**

Когда X0.0 включен, инструкция выполняет операцию матричное ИСКЛЮЧАЮЩЕЕ ИЛИ на каждой паре соответствующих битов в 16-битных объектах Y0 ~ Y2 и в 16-разрядных объектах Y10 ~ Y12. Результат операции сохраняется в 16-битных объектах Y20 ~ Y22.





**Примечание:**

1. Если  $S_1+n-1$ ,  $S_2+n-1$ , or  $D+n-1$  превышают допустимый предел, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение n меньше 1 или больше 256, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.

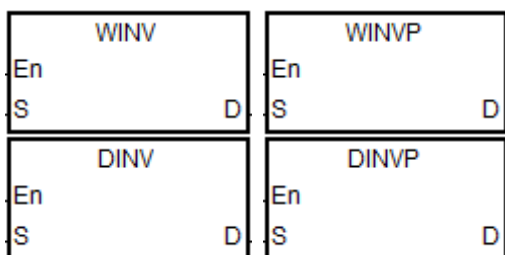
API	Код инструкции			Операнды								Функция					
0808	D	WINV	P	S · D								Логическое инвертирование INV					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○	○	○	○		
D	●	●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символ:**



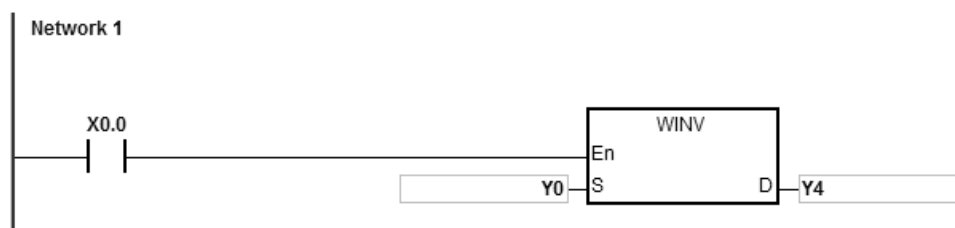
**S** : Источник данных  
**D** : Результат операции

**Описание:**

1. Инструкция инвертирует данные из **S**, результат сохраняется в **D**.
2. Только 32-битная инструкция DINV может использовать 32-битный счетчик, объект E использовать его не может.
3. При выполнении инструкции INV выполняется процесс реверсирования состояния. Если состояние равно 0 до выполнения инструкции INV, оно изменится на 1 в результате выполнения инструкции INV.

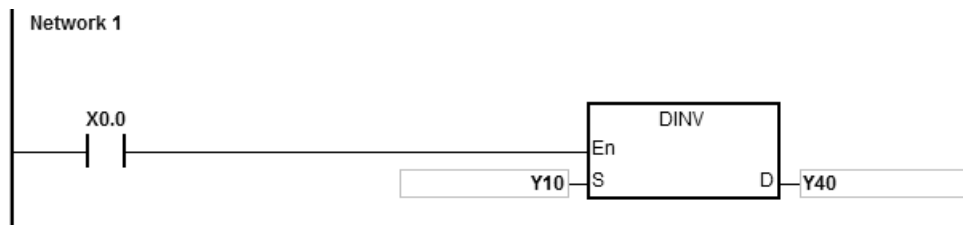
**Пример 1:**

Когда X0.0 включен, инструкция WINV инвертирует (операция INV) данные в 16-битном объекте Y0 по соответствующим битам. Результат операции сохраняется в 16-битном объекте Y4.



**Пример 2:**

Когда X0.0 включен, инструкция DINV инвертирует (операция INV) данные в 32-битном объекте (Y11, Y10) по соответствующим битам. Результат операции сохраняется в 32-битном объекте (Y41, Y40).



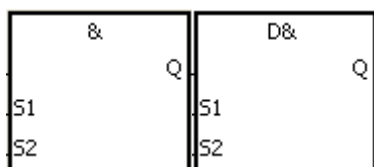
API	Код инструкции			Операнды							Функция						
0809~ 0811	D	LD #		$S_1 \cdot S_2$							Операция логического сравнения LD #						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●				●	●	
$S_2$		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ:



$S_1$  : Источник данных 1

$S_2$  : Источник данных 2

Инструкции для примера: LD& и DLD&

Описание:

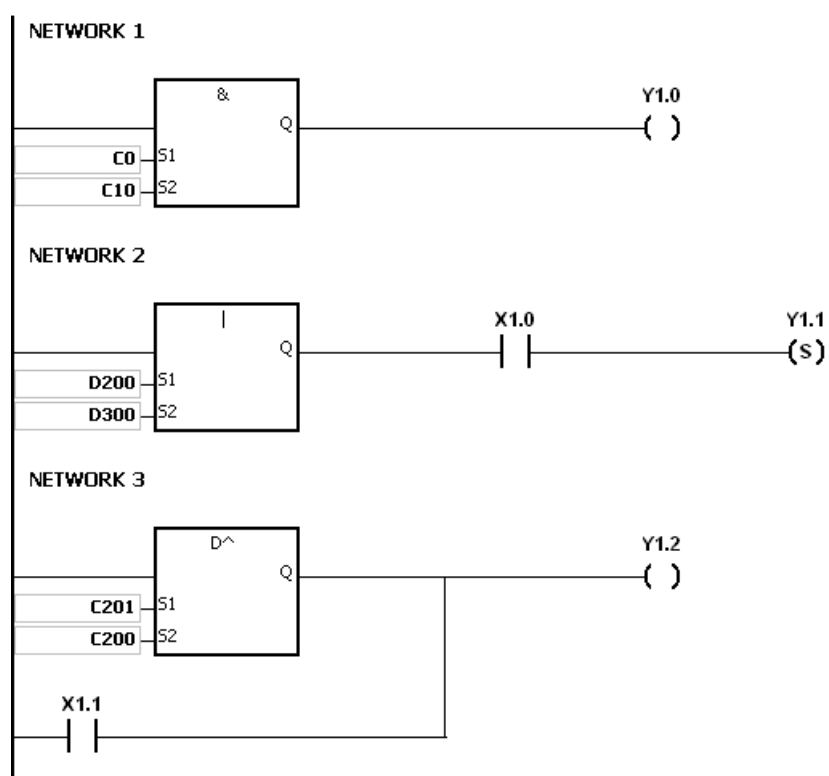
1. Инструкция сравнивает данные в  $S_1$  с данными в  $S_2$ . Если результат сравнения равен 0, это является условием непрерывности цепи. Если результатом сравнения является неравенство 0, это является условием прерывания цепи.
2. 32-битный счетчик используется только 32-битной инструкцией DLD #, объект E такие счетчики не использует.
3. Инструкция LD # может быть напрямую связана с базовой шиной.

API	16-битная инструкция	32-битная инструкция	Результат сравнения	
			Непрерывность	Прерывание
0809	LD&	DLD&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0810	LD	DLD	$S_1   S_2 \neq 0$	$S_1   S_2 = 0$
0811	LD^	DLD^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

4. &: Логическое И
5. |: Логическое ИЛИ
6. ^: Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ

**Пример:**

1. Инструкция LD& берет данные в C0 и C1 и выполняет логическую операцию И на каждой паре соответствующих битов. Когда результат операции не равен 0, Y1.0 включен.
2. Инструкция LD| берет данные в D200 и D300 и выполняет логическую операцию ИЛИ по каждой паре соответствующих битов. Когда результат операции не равен 0 и включен X1.0, Y1.1 включен.
3. Инструкция LD^ берет данные в C201 и C200 и выполняет логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ по каждой паре соответствующих битов. Когда результат операции не равен 0 или когда X1.1 включен, Y1.2 включен.



**Примечание:**

Если значения в S<sub>1</sub> или S<sub>2</sub> некорректны, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#2003.

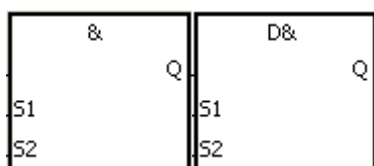
API	Код инструкции			Операнды							Функция						
0812~ 0814	D	AND #		$S_1 \cdot S_2$							Операция логического сравнения AND #						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●				●	●	
S <sub>2</sub>		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

**Символ:**



S<sub>1</sub> : Источник данных 1

S<sub>2</sub> : Источник данных 2

Инструкции для примера: AND& и DAND&

**Описание:**

1. Инструкция сравнивает данные в S<sub>1</sub> с данными в S<sub>2</sub>. Если результат сравнения равен 0, это является условием непрерывности цепи. Если результатом сравнения является неравенство 0, это является условием прерывания цепи.
2. 32-битный счетчик используется только 32-битной инструкцией DAND #, объект E такие счетчики не использует.
3. Инструкция AND # и контакт подключаются последовательно.

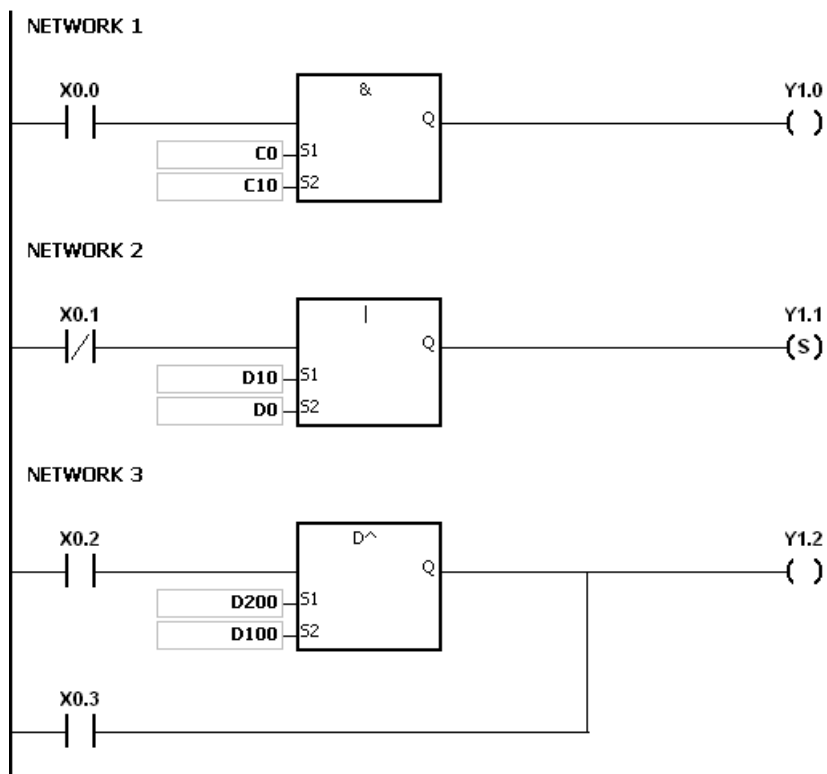
API	16-битная инструкция	32-битная инструкция	Результат сравнения	
			Непрерывность	Прерывание
0812	AND&	DAND&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0813	AND	DAND	$S_1   S \neq 0$	$S_1   S_2 = 0$
0814	AND^	DAND^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S = 0$

4. &: Логическое И
5. |: Логическое ИЛИ

6.  $\wedge$ : Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ

**Пример:**

1. Когда X0.0 включен, инструкция AND& берет данные в C0 и C10 и выполняет логическую операцию И на каждой паре соответствующих битов. Когда результат операции не равен 0, Y1.0 включен.
2. Когда X0.1 выключен, инструкция AND| берет данные в D10 и D0 и выполняет логическую операцию ИЛИ по каждой паре соответствующих битов. Когда результат операции не равен 0, Y1.1 остается включенным.
3. Когда X0.2 включен, инструкция AND $\wedge$  берет данные в 32-битном регистре (D200, D201) и в 32-битном регистре (D100, D101), затем выполняет логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ по каждой паре соответствующих битов. Когда результат операции не равен 0 или когда X0.3 включен, Y1.2 включен.



**Примечание:**

Если значения в S<sub>1</sub> или S<sub>2</sub> некорректны, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#2003.

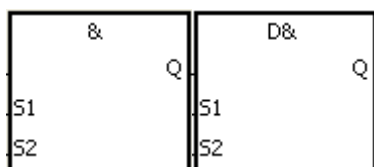
API	Код инструкции			Операнды							Функция						
0815~ 0817	D	OR #		$S_1 \cdot S_2$							Операция логического сравнения OR #						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●				●	●	
S <sub>2</sub>		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ:



S<sub>1</sub> : Источник данных 1

S<sub>2</sub> : Источник данных 2

Инструкции для примера OR& и DOR&

Описание:

- Инструкция сравнивает данные в S<sub>1</sub> с данными в S<sub>2</sub>. Если результат сравнения равен 0, это является условием непрерывности цепи. Если результатом сравнения является неравенство 0, это является условием прерывания цепи.
- 32-битный счетчик используется только 32-битной инструкцией DOR #, объект E такие счетчики не использует.
- Инструкция OR # и контакт подключаются параллельно.

API	16-битная инструкция	32-битная инструкция	Результат сравнения	
			Непрерывность	Прерывание
0815	OR&	DOR&	$S_1 \& S_2 \neq 0$	$S_1 \& S_2 = 0$
0816	OR	DOR	$S_1   S_2 \neq 0$	$S_1   S_2 = 0$
0817	OR^	DOR^	$S_1 \wedge S_2 \neq 0$	$S_1 \wedge S_2 = 0$

4. &: Логическое И

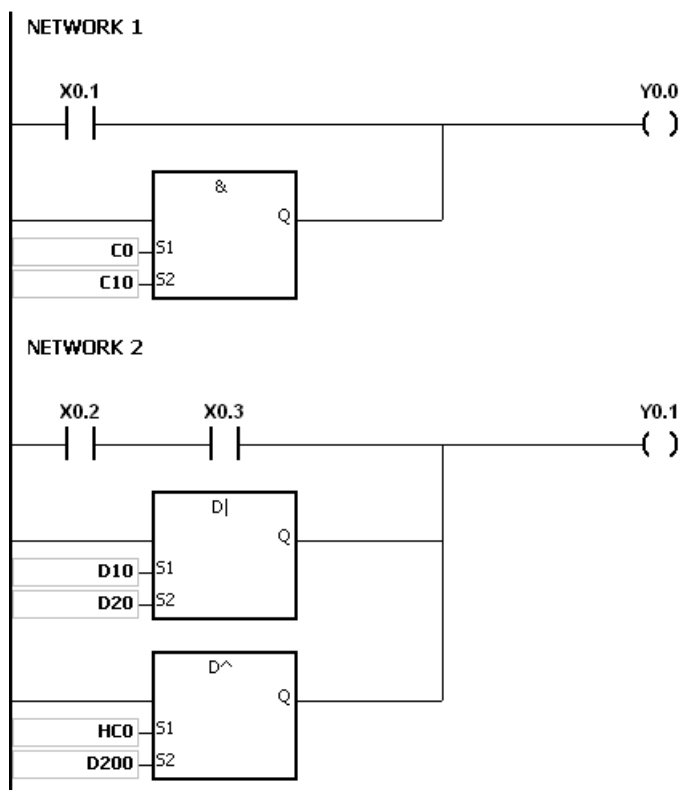
5. |: Логическое ИЛИ



6.  $\wedge$ : Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ

**Примечание:**

1. Когда X0.1 выключен, Y0.0 включен. Кроме того, когда инструкция OR& выполняет логическую операцию И на каждой паре соответствующих битов в C0 и C10 и результат операции не равен 0, Y0.0 включен.
2. Когда X0.2 и X0.3 включены, Y0.1 включен. Когда инструкция OR| выполняет операцию логического ИЛИ по каждой паре соответствующих битов в 32-битном регистре (D10, D11) и 32-битном регистре (D20, D21), а результат операции не равен 0, включается Y0.1. Кроме того, Когда инструкция OR $\wedge$  выполняет логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ по каждой паре соответствующих битов в 32-разрядном счетчике HC0 и 32-битном регистре (D200, D201) и результат операции не равен 0, также включается Y0.1.



**Примечание:**

Если значения в S<sub>1</sub> или S<sub>2</sub> некорректны, инструкция не выполняется, включается флаг SM0, в SR0 записывается код ошибки 16#2003.

## 6.10 Инструкции вращения

### 6.10.1 Описание инструкций вращения

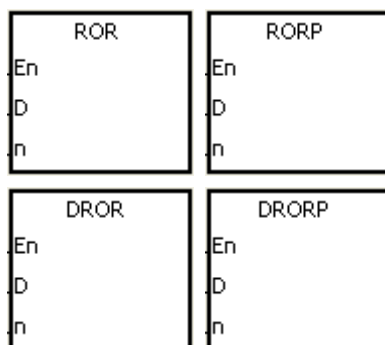
API	Код инструкции			Операнд								Функция				
0900	D	ROR	P	D, n								Вращение вправо				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
D		●			●	●	●	●			○	○				
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символьное обозначение:



D : Вращаемый регистр

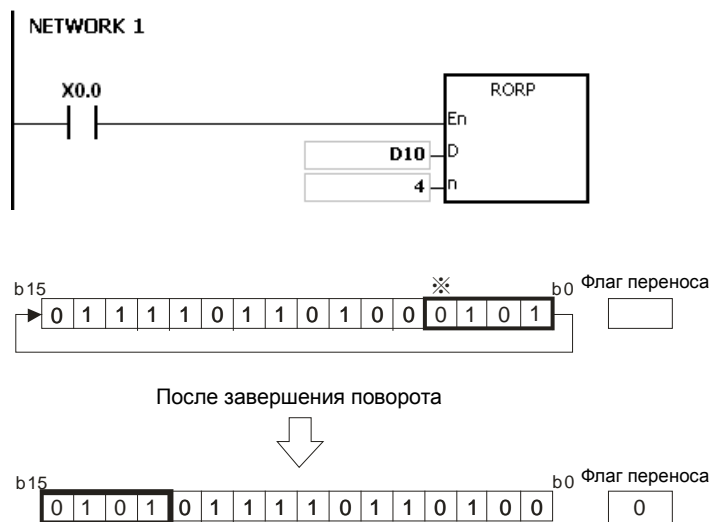
n : Количество битов в группе

Описание:

1. Значения битов регистра, заданного параметром **D**, делятся по группам (**n**-битов в группе), и эти группы вращаются вправо.
2. 32-битный счетчик может использоваться только в инструкции DROR, а в регистре E – не может.
3. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 16. Операнд **n**, используемый в 32-битной инструкции, может иметь значение в диапазоне от 1 до 32. Если операнд **n** меньше 0, инструкция не выполняется; если значение операнда **n** вне пределов указанного диапазона, инструкция выполняется, а значением операнда считается максимальное значение диапазона (32).
4. Как правило, используются импульсные инструкции RORP и DRORP.

**Пример:**

Если состояние входа X0.0 меняется с OFF на ON, значения битов регистра D10 делятся по группам (по четыре бита в группе) и эти группы поворачиваются вправо (значение бита, помеченного знаком ※, передается флагу переноса SM602).



**Дополнительные замечания:**

1. Если значение регистра вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

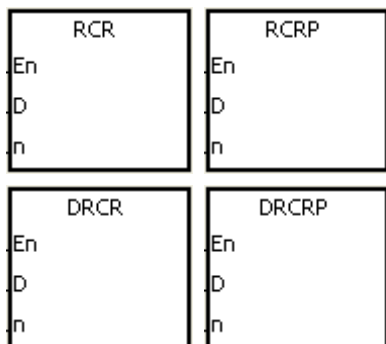
API	Код инструкции			Операнд							Функция						
0901	D	R	C	P	D, n							Вращение вправо с флагом переноса					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
D		●			●	●	●	●			○	○				
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



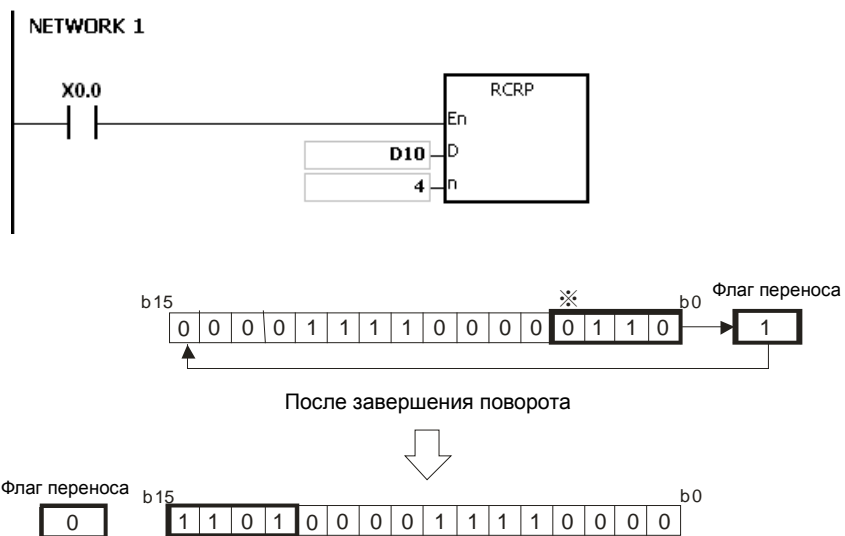
- D** : Вращаемый регистр
- n** : Количество битов в группе

**Описание:**

1. Значения битов регистра, заданного параметром **D**, делятся по группам (**n**-битов в группе), и эти группы вращаются вправо с флагом переноса SM602.
2. 32-битный счетчик может использоваться только в инструкции DR CR, а в регистре E – не может.
3. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 16. Операнд **n**, используемый в 32-битной инструкции, может иметь значение в диапазоне от 1 до 32. Если операнд **n** меньше 0, инструкция не выполняется; если значение операнда **n** вне пределов указанного диапазона, инструкция выполняется, а значением операнда считается максимальное значение диапазона (32).
4. Как правило, используются импульсные инструкции RCRP и DR CRP.

**Пример:**

Если состояние входа X0.0 меняется с OFF на ON, значения битов регистра D10 делятся по группам (по четыре бита в группе) и эти группы поворачиваются вправо с флагом переноса SM602 (значение бита, помеченного знаком ※, передается флагу переноса SM602).



**Дополнительные замечания:**

1. Если значение регистра вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

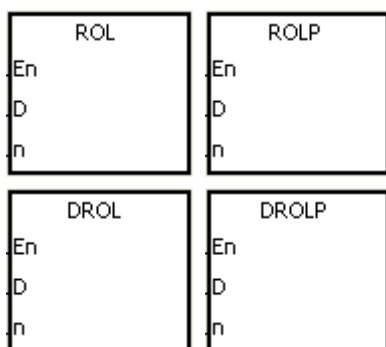
API	Код инструкции			Операнд							Функция					
0902	D	ROL	P	D, n							Вращение влево					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
D		●			●	●	●	●			○	○				
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



**D** : Вращаемый регистр

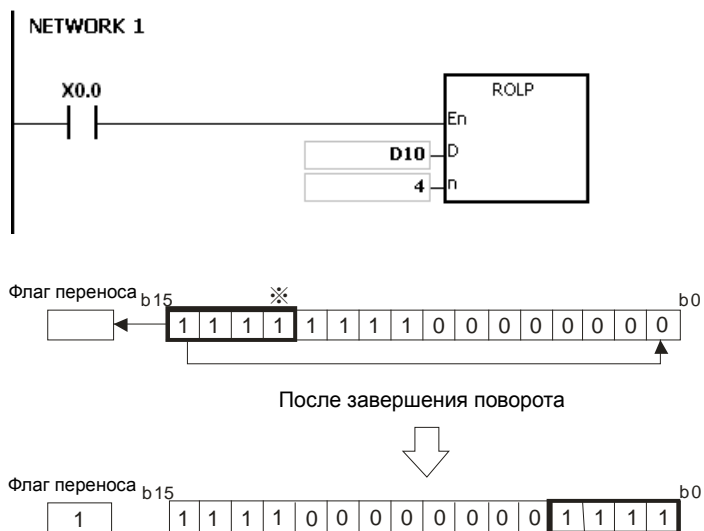
**n** : Количество битов в группе

**Описание:**

1. Значения битов регистра, заданного параметром **D**, делятся по группам (**n**-битов в группе) и эти группы вращаются влево.
2. 32-битный счетчик может использоваться только в инструкции DROL, а в регистре E – не может.
3. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 16. Операнд **n**, используемый в 32-битной инструкции, может иметь значение в диапазоне от 1 до 32. Если операнд **n** меньше 0, инструкция не выполняется; если значение операнда **n** вне пределов указанного диапазона, инструкция выполняется, а значением операнда считается максимальное значение диапазона (32).
4. Как правило, используются импульсные инструкции ROLP и DROLP.

**Пример:**

Если состояние входа X0.0 меняется с OFF на ON, значения битов регистра D10 делятся по группам (по четыре бита в группе) и эти группы поворачиваются влево (значение бита, помеченного знаком ※, передается флагу переноса SM602).



**Дополнительные замечания:**

1. Если значение регистра вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

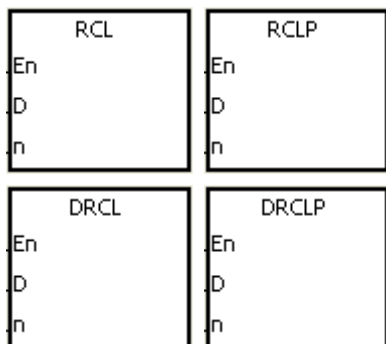
API	Код инструкции			Операнд							Функция					
0903	D	RCL	P	D, n							Вращение влево с флагом переноса					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
D		●			●	●	●	●			○	○				
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



- D** : Вращаемый регистр
- n** : Количество битов в группе

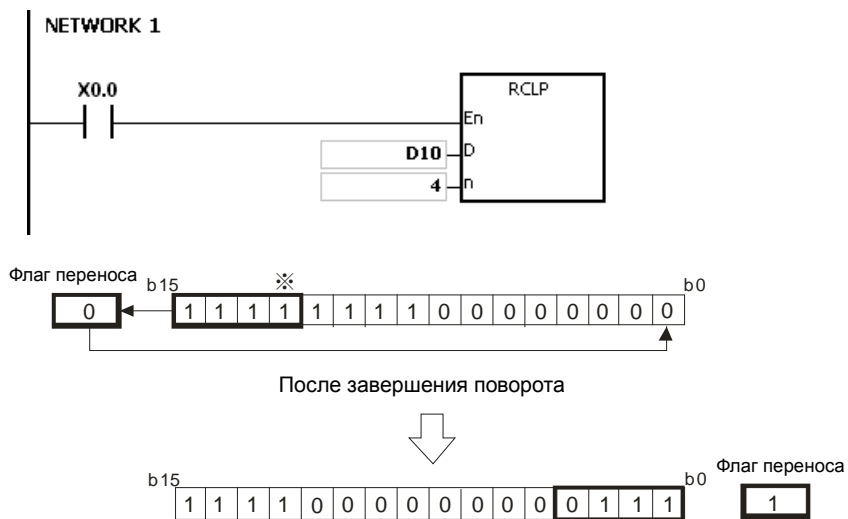
**Описание:**

1. Значения битов регистра, заданного параметром **D**, делятся по группам (**n**-битов в группе), и эти группы вращаются влево с флагом переноса SM602.
2. 32-битный счетчик может использоваться только в инструкции DRCL, а в регистре E – не может.
3. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 16. Операнд **n**, используемый в 32-битной инструкции, может иметь значение в диапазоне от 1 до 32. Если операнд **n** меньше 0, инструкция не выполняется; если значение операнда **n** вне пределов указанного диапазона, инструкция выполняется, а значением операнда считается максимальное значение диапазона (32).
4. Как правило, используются импульсные инструкции RCLP и DRCLP.

**Пример:**

Если состояние входа X0.0 меняется с OFF на ON, значения битов регистра D10 делятся по группам (по четыре бита в группе) и эти группы поворачиваются влево с флагом переноса SM602 (значение бита, помеченного знаком ※, передается флагу переноса SM602).





**Дополнительные замечания:**

1. Если значение регистра вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

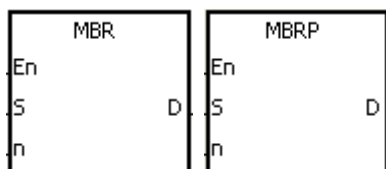
API	Код инструкции			Операнд							Функция					
0904		MBR	P	S, D, n							Вращение битов матрицы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



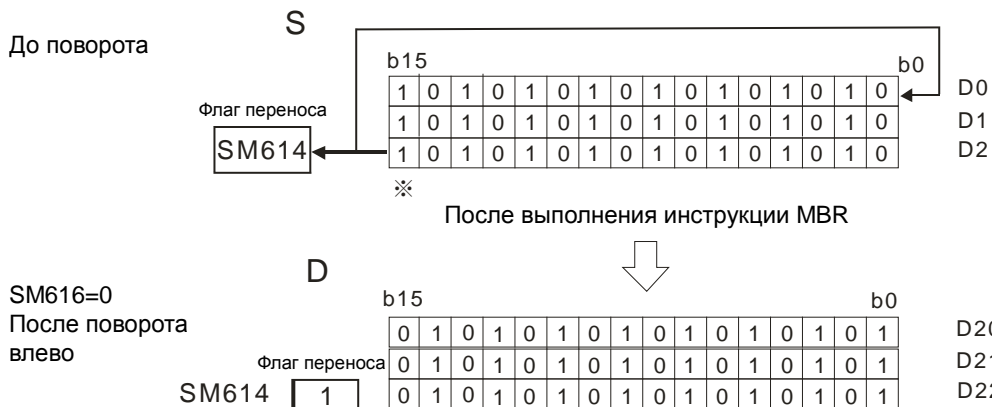
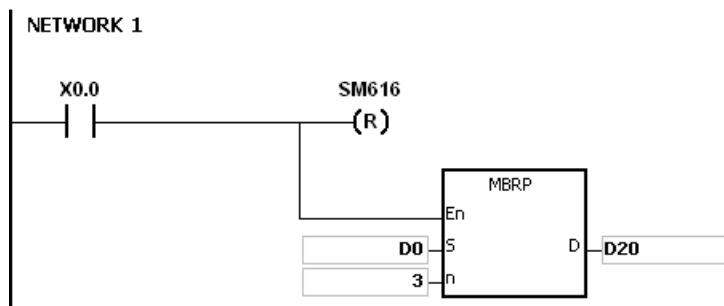
- S** : Источник матрицы
- D** : Результат операции
- n** : Длина массива

**Описание:**

1. Значения битов n-рядов источника матрицы **S** вращаются вправо или влево. Если SM616 = OFF, значения битов вращаются влево. Если SM616 = ON, значения битов вращаются вправо. В освободившееся после поворота место записывается значение бита, который был повернут последним, а результат операции записывается в параметр **D**. Значение бита, который был повернут последним, записывается не только на освободившееся место, но и передается флагу переноса SM614.
2. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 16. Операнд **n**, используемый в 32-битной инструкции, может иметь значение в диапазоне от 1 до 32. Если операнд **n** меньше 0, инструкция не выполняется; если значение операнда **n** вне пределов указанного диапазона, инструкция выполняется, а значением операнда считается максимальное значение диапазона (32).
3. Как правило, используется импульсная инструкция MBRP.

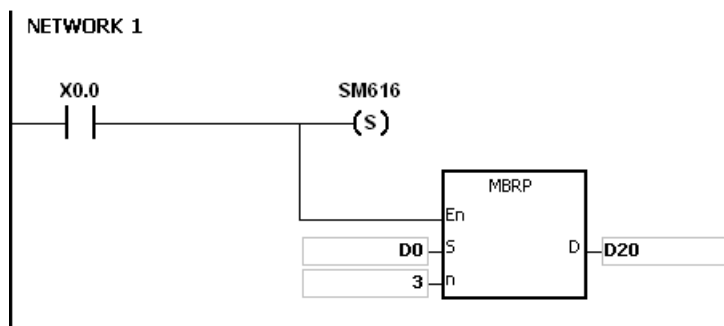
**Пример 1:**

Вход X0.0 = ON и SM616 = OFF. Значения битов 16-битных регистров D0–D2 поворачиваются влево, а результат операции записывается в 16-битные регистры D20–D22. Значение бита, помеченного знаком ✖, передается флагу переноса SM614.



**Пример 2:**

Вход X0.0 = ON и SM616 = ON. Значения битов 16-битных регистров D0~D2 поворачиваются вправо, а результат операции записывается в 16-битные регистры D20~D22. Значение бита, помеченного знаком ※, передается флагу переноса SM614.





## 6.11 Инструкции таймеров и счетчиков

### 6.11.1 Описание инструкций таймеров и счетчиков

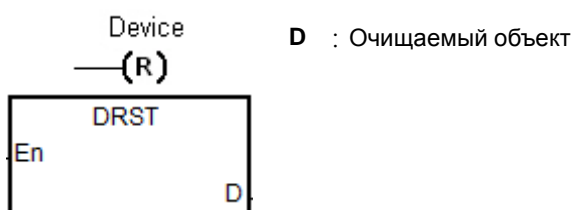
API	Код инструкции			Операнды									Функция			
1000	D	RST		D									Сброс контакта на ВЫКЛ или очистка значения в регистре			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D		○	○	○	○	○	○	○		○	○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●	●	●			●	●		●		●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символ:



Описание:

- Инструкция DRST сбрасывает значение только 32-битных счетчиков HC и двух последовательных 16-битных регистров D. Для других объектов используйте инструкцию RST.
- Действия объектов при выполнении программы RST.

Объект	Состояние
Bit	Выключение катушки или контакта
T · C	Сброс на 0 текущего значения таймера или счетчика. Выключение катушки или контакта
Word	Сброс на 0 16-битного значения
DWord · HC · Real	Сброс на 0 32-битных значений, включая числа с плавающей запятой

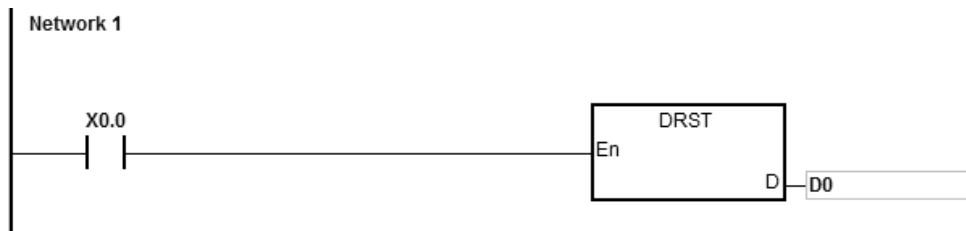
- Если инструкция RST не выполняется, состояние определенных для инструкции объектов не изменяется.
- Инструкция поддерживает прямые выходы.

**Пример:**

Когда X0.0 включен, Y0.5 выключается.



Когда X0.0 включен, 32-битное значение в D1 или D0 сбрасывается на 0.



API	Код инструкции			Операнды								Функция				
1001		TMR		<b>S<sub>1</sub> · S<sub>2</sub></b>								16-битный таймер (шаг: 100 мс)				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>					○											
S <sub>2</sub>								○				○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>											●		
S <sub>2</sub>		●				●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**



- S<sub>1</sub> : Номер таймера
- S<sub>2</sub> : Заданное значение таймера

**Описание:**

См. описание инструкции API1002 TMRH.

API	Код инструкции				Операнды								Функция				
1002		TMRH			$S_1 \cdot S_2$								16-битный таймер (шаг: 1 мс)				
Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F	
$S_1$					○												
$S_2$								○				○	○	○			

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$											●		
$S_2$		●				●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:

<b>TMRH</b>
$E_n$
$S_1$
$S_2$

$S_1$  : Номер таймера

$S_2$  : Заданное значение таймера

Описание:

1. Таймер T, работающий с инструкцией TMR имеет шаг 100 мс, а таймер T, работающий с инструкцией TMRH имеет шаг 1 мс.
2. Диапазон задания значений таймера для инструкций TMR и TMRH: 0~32767.
3. Если один и тот же таймер T используется повторно в программе, в том числе используется в разных инструкциях TMR и TMRH, значение его настройки – это значение, заданное первому таймеру по порядку в программе.
4. Значение таймера T автоматически сбрасывается на 0 при отключении назначенного таймеру контакта.
5. Добавление символа S перед T для таймера, работающего с инструкцией TMR, превращает его в аккумулятивный таймер. При отключении назначенного контакта значение аккумулятивного таймера не сбрасывается, а при повторном включении назначенного контакта подсчет начинается с сохраненного значения. Аккумулятивные таймеры ST для сброса значения применяют инструкцию RST.
6. Если один и тот же таймер используется в программе повторно T, он отключается при отключении любого из назначенных контактов.
7. Если один и тот же таймер T используется в программе повторно как ST, таймер T выключается, когда выключается один из назначенных контактов.



8. Когда выполняется инструкция TMR, при включении назначенной катушки начнется отсчет. Когда значение таймера достигнет заданного, включается назначенный контакт.
9. Таймеры T0~T411 определяются как общие таймеры. T412~T511 по умолчанию являются таймерами подпрограмм. Если диапазоны двух типов таймеров необходимо изменить, используйте программное обеспечение аппаратной конфигурации HWCONFIG.
10. Общие таймеры сравнивают значения времени при сканировании во время выполнения инструкции TMR. Они применяются при каждом сканировании инструкции TMR.

Для таймеров подпрограмм, система подсчитывает время и сравнивает значения после выполнения инструкции END. Эти таймеры применяются не при каждом сканировании инструкции TMR, но когда необходим длительный срок подсчета.

**Пример 1:**

Когда X0.0 включен, заданное значение 50 вводится в таймер T0. Когда значение таймера T0 достигает 50, контакт таймера T0 включается.



6

**Пример 2:**

Когда X0.0 включен, заданное значение 50 вводится в таймер T0. Когда значение таймера T0 достигает 25 и включается X0.0, T0 подсчитывает от 25 до 50, контакт таймера T0 включается.



**Пример 3:**

Когда X0.0 включен, заданное значение 1000 вводится в таймер T5. Когда значение таймера T5 достигает 500 и включается X0.0, T5 подсчитывает от 500 до 1000, контакт таймера T5 включается.

**Пример 4:**

Когда X0.0 включен, заданное значение 1000 вводится в таймер ST5. Когда значение таймера ST5 достигает 500 и включается X0.0, ST5 подсчитывает от 500 до 1000, контакт таймера ST5 включается.

**Примечание:**

Когда операнд S1 задается через ПО ISPSOft, для общего таймера T должен быть выбран тип данных TIMER. Если используется аккумулятивный таймер ST, необходимо задать тип ST.

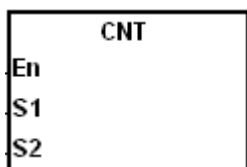
API	Код инструкции			Операнды								Функция				
1003		CNT		S <sub>1</sub> · S <sub>2</sub>								16-битный счетчик				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>						○										
S <sub>2</sub>								○				○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>												●	
S <sub>2</sub>		●				●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**



- S<sub>1</sub> : Номер счетчика
- S<sub>2</sub> : Заданное значение счетчика

**Описание:**

1. Когда инструкция CNT активируется, включается назначенная счетчику катушка и значение счетчика увеличивается на 1. Когда значение счетчика достигает заданного, включается назначенный счетчику контакт.
2. Когда значение счетчика достигает заданного, состояние контакта и значение счетчика остаются неизменными при вводе любого счетного импульса. Для сброса счетчика используйте инструкцию RST (после чего возможно проведение следующего подсчета).

**Пример:**

Когда SM408 включается первый раз, заданное значение 10 вводится в счетчик C0 и начинается отсчет. После включения SM408 десять раз, значение в счетчике C0 достигает 10 и включается контакт, заданный для счетчика C0. После после включения контакта C0 включен, SM408 продолжает включаться, но значение счетчика C0 увеличиваться не будет.



**Примечание:** Если операнд S1 задается в ПО ISPSOft, для него выбирается тип данных COUNTER.

API	Код инструкции			Операнды								Функция				
1004		DCNT		$S_1 \cdot S_2$								32-битный счетчик				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$							○									
$S_2$								○					○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$													
$S_2$			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символ:

DCNT
En
S1
S2

 $S_1$  : Номер счетчика $S_2$  : Заданное значение счетчика

Описание:

- DCNT - это инструкция для включения 32-разрядного счетчика в диапазоне между HC0 и HC255.
- Если операнд S1 задается в ПО ISPSOft, тип данных CNT не может быть выбран, необходимо задать номер объекта HC.
- Для счетчиков прямого/обратного счета HC0 ~ HC63, когда заданный контакт для инструкции DCNT включается, счетчик будет осуществлять прямой подсчет, увеличивая значения на 1 при выключенных SM621 ~ SM684 или обратный подсчет, уменьшая значения на 1 при включенных SM621 ~ SM684.
- Для счетчиков прямого подсчета HC64~HC199, подсчет с увеличением значения на 1 при включении заданного инструкции DCNT контакта.
- При прекращении выполнения инструкции DCNT счетчик прекращает подсчет. Значение счетчика останется неизменным, и сбросить его можно с помощью инструкции RST.
- Более подробную информацию о высокоскоростных счетчиках HC200~HC255 см. ниже.

**Пример 1:**

NETWORK1→

При запуске ПЛК, значение счетчика HC0 очищается, и, когда SM621 выключен, начинается подсчет. В данном примере SM408 включается первый раз. Таким образом, заданное значение 10 будет загружено в счетчик HC0, и счетчик начнет подсчет.

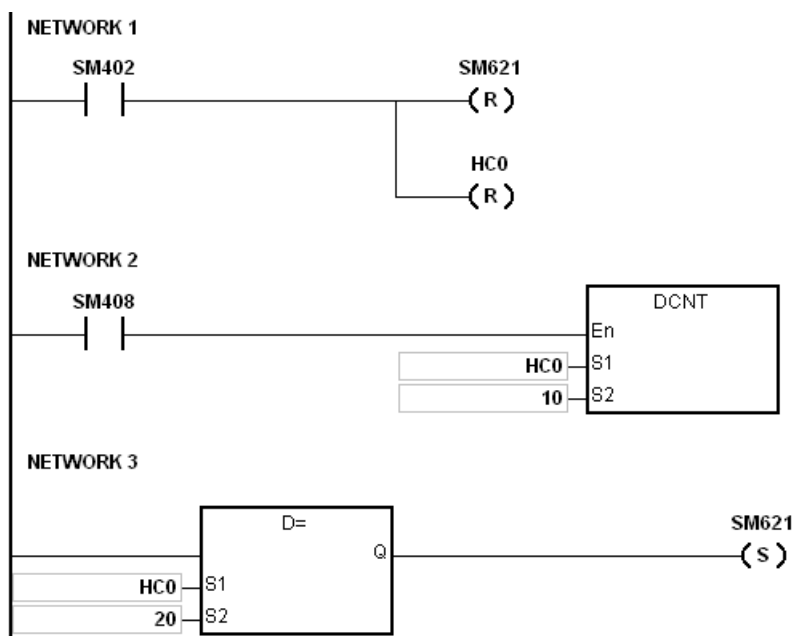
NETWORK2→

После включения SM408 десять раз, значение в счетчике HC0 достигает заданного (10) и включается заданный для HC0 контакт. После включения HC0, значение в счетчике будет увеличиваться, поскольку SM408 продолжает включаться, хотя значение в HC0 достигло заданного.

NETWORK3→

Когда HC0 продолжает подсчет и значение достигает 20, при включенном SM621, счетчик переключится на обратный счет. После того, как SM408 включится 10 раз, а значение в HC0 уменьшится с 10 до 9, заданный HC0 контакт будет отключен.

После отключения контакта значение в HC0 будет продолжать уменьшаться, поскольку SM408 продолжает включаться.



**Примечание:**

По режиму настройки SM621–SM684, см. описание 32-битных счетчиков HC в Главе 2.

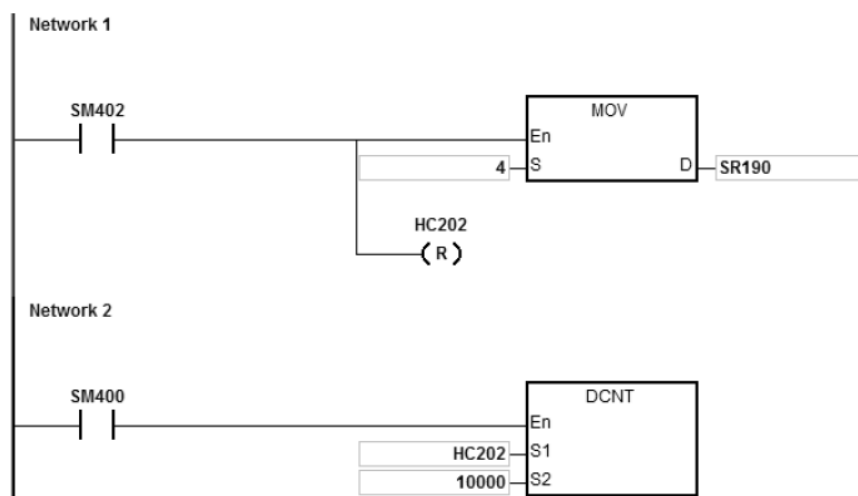
**Пример 2:**

NETWORK1→

Когда ПЛК запускается, установите значение счетчика HC202 на четыре временных частоты (настройка режима должна быть установлена перед выполнением команды DCNT). И тогда значение счетчика HC202 очищается.

NETWORK2→

После того, как значение счетчика HC202 достигло заданного значения 1000, контакт HC202 включается.



**Описание высокоскоростных счетчиков:**

Высокоскоростные счетчики в ПЛК AS можно разделить на аппаратные счетчики (максимум до 200 кГц и, для дифференцированных входов, до 4 МГц) и программные счетчики (максимум до 10 кГц).

**Аппаратные счетчики:**

Вход № HC	X0.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HC200	P <sup>#1</sup>												R <sup>#1</sup>			
HC201	P	D <sup>#1</sup>											R			
HC202	A <sup>#1</sup>	B <sup>#1</sup>											R			
HC203	-- <sup>#2</sup>	--											--			
HC204			P											R		
HC205			P	D										R		
HC206			A	B										R		
HC207			--	--										--		

HC208					P											R	
HC209					P	D										R	
HC210					A	B										R	
HC211					--	--										--	
HC212							P										R
HC213							P	D									R
HC214							A	B									R
HC215							--	--									--
HC216									P								
HC217									P	D							
HC218									A	B							
HC219									--	--							
HC220											P						
HC221											P	D					
HC222											A	B					
HC223											--	--					

6

Примечание 1: P: однофазный импульсный вход, D: вход сигнала направления, A и B: двухфазные входы, R: вход сигнала сброса.

Примечание 2: Знак '--' означает, что режим подсчета зарезервирован и не может использоваться сейчас. Пустое поле показывает отсутствие функций.

Примечание 3: См. описание SM/SR для выбора прямого/обратного подсчета и кратность умножения частоты импульсов высокоскоростного счетчика.

Примечание 4: Функция R (вход сигнала сброса) по умолчанию отключена. См. описание SM/SR для работы с данной функцией R.

Возьмем для примера счетчик HC200. SM291 включается и запускает функцию R, а затем сигнал по переднему фронту на X0.12 запускает сброс значения счетчика HC200.

## Программные счетчики:

Вход № HC	X0.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
HC232													P			
HC233													P	D		
HC234													A	B		
HC235													UP <sup>#5</sup>	DN <sup>#5</sup>		
HC236															P	
HC237															P	D
HC238															A	B
HC239															UP	DN
HC240		P														
HC241	UP	DN														
HC242				P												
HC243			UP	DN												
HC244						P										
HC245					UP	DN										
HC246								P								
HC247							UP	DN								
HC248										P						
HC249									UP	DN						
HC250												P				
HC251											UP	DN				
HC252														P		
HC253																P

Примечание 5: UP: Однофазный вход прямого счета (аналогично CW), DN: Однофазный вход обратного счета (аналогично CCW)

Высокоскоростные счетчики в диапазоне HC200 ~ HC255, которые не указаны в таблице, являются зарезервированными объектами внутри ПЛК. Разработчику программы для использования они не рекомендуются.



**Высокоскоростные счетчики, объекты SM/SR и соответствующие функции.**

HC No.	Функции прямого/обратного счета			Запуск функции сброса значений	Реверсирование направления	Режим подсчета
	SM No.	Атрибут	Описание	SM No.	SM No.	SR No.
HC200	SM300	R/W	Отображение / задание	SM291	SM281 (применим для HC201)	SR190
HC201	SM301	R	Отображение			
HC202	SM302	R	Отображение			
HC204	SM304	R/W	Отображение / задание	SM292	SM282 (применим для HC205)	SR191
HC205	SM305	R	Отображение			
HC206	SM306	R	Отображение			
HC208	SM308	R/W	Отображение / задание	SM293	SM283 (применим для HC209)	SR192
HC209	SM309	R	Отображение			
HC210	SM310	R	Отображение			
HC212	SM312	R/W	Отображение / задание	SM294	SM284 (применим для HC213)	SR193
HC213	SM313	R	Отображение			
HC214	SM314	R	Отображение			
HC216	SM316	R/W	Отображение / задание	--	SM285 (применим для HC217)	SR194
HC217	SM317	R	Отображение			
HC218	SM318	R	Отображение			
HC220	SM320	R/W	Отображение / задание	--	SM286 (применим для HC221)	SR195
HC221	SM321	R	Отображение			
HC222	SM322	R	Отображение			
HC232	SM332	R/W	Отображение / задание	--	SM287 (применим для HC233)	SR196
HC233	SM333	R	Отображение			
HC234	SM334	R	Отображение			
HC235	SM335	R	Отображение	--	SM288 (применим для HC237)	SR197
HC236	SM336	R/W	Отображение / задание			
HC237	SM337	R	Отображение			
HC238	SM338	R	Отображение	--	--	Поддержка только 1-кратной частоты и переключения по переднему фронту сигнала
HC239	SM339	R	Отображение			
HC240	SM340	R/W	Отображение / задание			
HC241	SM341	R	Отображение	--	--	Поддержка только 1-кратной частоты
HC242	SM342	R/W	Отображение / задание			
HC243	SM343	R	Отображение			
HC244	SM344	R/W	Отображение / задание	--	--	Поддержка только 1-кратной частоты
HC245	SM345	R	Отображение			
HC246	SM346	R/W	Отображение /			

HC No.	Функции прямого/обратного счета			Запуск функции сброса значений	Реверсирование направления	Режим подсчета
	SM No.	Атрибут	Описание	SM No.	SM No.	SR No.
			задание			и переключения по переднему фронту сигнала
HC247	SM347	R	Отображение	--	--	Поддержка только 1-кратной частоты и переключения по переднему фронту сигнала
HC248	SM348	R/W	Отображение / задание			
HC249	SM349	R	Отображение			
HC250	SM350	R/W	Отображение / задание			
HC251	SM351	R	Отображение			
HC252	SM352	R/W	Отображение / задание			
HC253	SM353	R	Отображение			

Примечание 1: Все флаги SM в таблице по умолчанию отключены.

Примечание 2: Когда SM в разделе «Функция прямого / обратного счета» выключен, это означает, что соответствующий счетчик осуществляет прямой подсчет или отображает прямой подсчет. Если SM включен, это означает, что соответствующий счетчик осуществляет обратный подсчет или отображает обратный подсчет.

Примечание 3: «R» имеет атрибут «Только чтение», а «W» означает «Задать»

Примечание 4: Специальные регистры SR в «Режиме подсчета» по умолчанию имеют входную 1-кратную частоту. Входное значение 2 указывает на 2-кратную частоту, а 4 - на 4-кратную. 4-кратная частота применяется только для счетчика 2-фазного входа A / B. Если значение в SR не равно 1, 2 и 4, ПЛК будет работать с 1-кратной частотой.

Примечание 5: Все однофазные счетчики производят подсчет с использованием 1-кратной частоты и режима подсчета по переднему фронту.

Примечание 6: Реверс направления применим для счетчиков типа «P» (импульсный вход) и «D» (направление). Когда SM включен, текущее направление подсчета (прямое / обратное) будет отменено. Например, когда вход предустановленного направления выключен, счетчик осуществляет прямой подсчет. Когда включается SM, счетчик переключается на обратный подсчет.

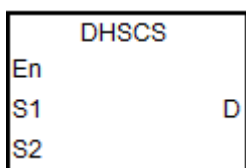
API	Код инструкции			Операнды							Функция						
1005	D	HSCS		$S_1 \cdot S_2 \cdot D$							Настройка высокоскоростного сравнения						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$							○									
$S_2$								●					○	○		
<b>D</b>		○	○	○				○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$												●	
$S_2$			●				●						
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

**Символ:**



$S_1$  : Номер счетчика

$S_2$  : Сравнительное значение

**D** : Результат сравнения

**Описание:**

1. Данная инструкция применяется с высокоскоростными счетчиками HC200 и выше по номеру. Если значение в высокоскоростном счетчике, заданном в  $S_1$  увеличивается или уменьшается на 1, инструкция DHSCS немедленно проводит сравнение. Если текущее значение в счетчике равно сравнительному значению в  $S_2$ , включается контакт, заданный с операндом **D**. Этот контакт, заданный с **D**, будет оставаться включенным, даже если результат сравнения изменится на неравенство.
2. Если для **D** заданы выходы Y0.0 ~ Y0.15, то при равенстве значений в  $S_2$  и текущего значения высокоскоростного счетчика, на выходные клеммы Y0.0 ~ Y0.15 будет выводиться сигнал. На другие объекты типа Y будут влиять результаты цикла сканирования. Но все эти объекты обновляются немедленно и их состояние не зависит от цикла сканирования.
3. Операнд **D** может также быть задан как прерыватель I в диапазоне: I2□□, □□=00~67.
4. Высокоскоростные счетчики можно разделить на счетчик программного обеспечения и счетчик аппаратных средств. Доступные высокоскоростные компараторы и номера регистров прерываний перечислены в следующей таблице.

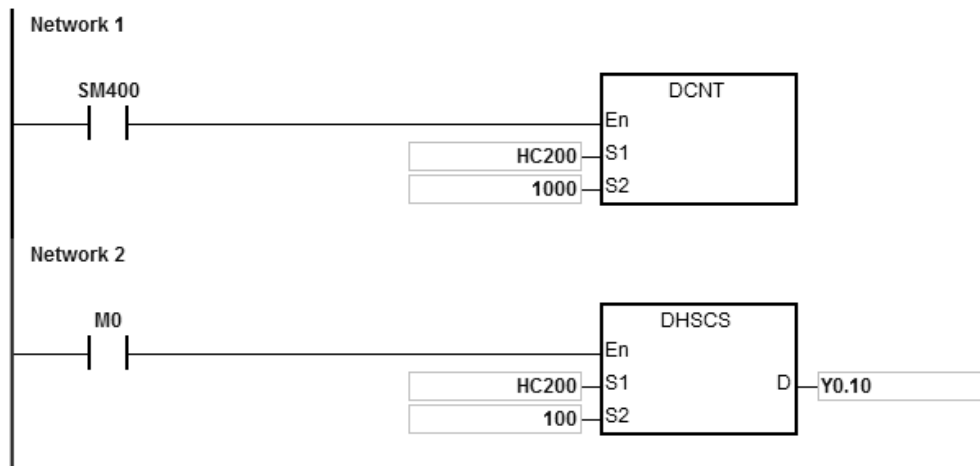
Тип	Диапазон счетчиков	Диапазон высокоскоростных компараторов	Диапазон высокоскоростных прерывателей
Аппаратный счетчик	HC200 ~ HC203	Компаратор: HCC00~HCC03	I200~I203
	HC204 ~ HC207	Компаратор: HCC04~HCC07	I210~I213
	HC208 ~ HC211	Компаратор: HCC08~HCC11	I220~I223
	HC212 ~ HC215	Компаратор: HCC12~HCC15	I230~I233
	HC216 ~ HC219	Компаратор: HCC16~HCC19	I240~I243
	HC220 ~ HC223	Компаратор: HCC20~HCC23	I250~I253
Программный счетчик	HC232 ~ HC253	-	I260~I267

5. Описание аппаратных компараторов для инструкций DHSCS, DHSCR и DHSZ:
  - Каждая группа аппаратных счетчиков объединяет 4 высокоскоростных компаратора. Одна инструкция DHSCS или DHSCR работает с 1 высокоскоростным компаратором. Инструкция DHSZ использует 2 высокоскоростных компаратора.
  - Во время редактирования программы каждая группа аппаратных счетчиков может использовать максимум 4 высокоскоростных компаратора для работы инструкций DHSCS, DHSCR или DHSZ. В противном случае отображается синтаксическая ошибка.
6. Описание программных компараторов для инструкций DHSCS и DHSCR:
  - Для сравнения функций Set или Reset применяется 8 программных компараторов. Каждая инструкция DHSCS или DHSCR использует один высокоскоростной компаратор.
  - Программные компараторы сравнивают прерывания, назначая соответствующий программный компаратор в соответствии с номерами прерываний. Отмечается, что один и тот же номер прерывания нельзя использовать повторно.
  - Для инструкций DHSCS и DHSCR число компараторов функций Set или Reset в программе не может превышать 8 единиц. В противном случае отображается синтаксическая ошибка.
7. Описание программных компараторов для инструкции DHSZ:
  - Существует 8 программных компараторов для зонного сравнения. В одной инструкции DHSZ используется один компаратор.
  - Инструкция DHSZ может использовать максимум 8 программных компараторов. При превышении этого количества отображается синтаксическая ошибка.

**Пример 1:**

Когда M0 включен, начинает выполняться инструкция DHSCS.

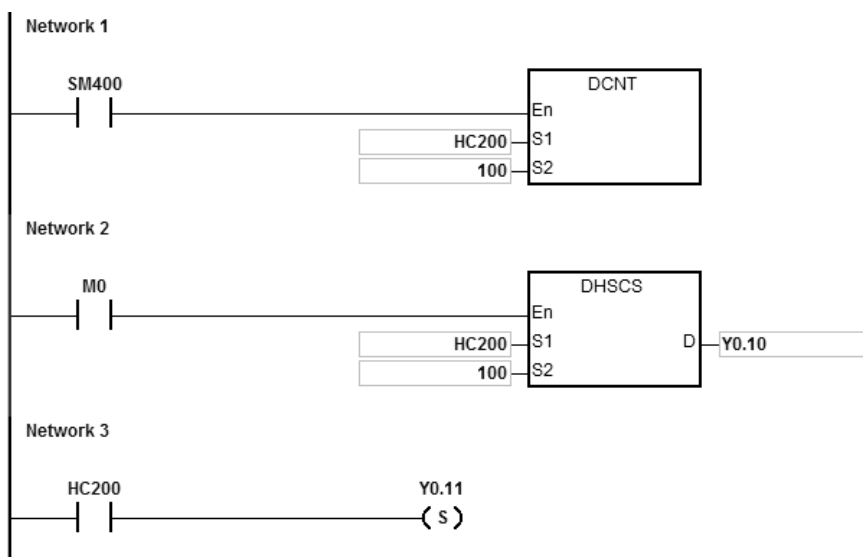
Когда текущее значение HC200 меняется от 99 до 100 или от 101 до 100, включается объект Y0.10, который будет выводить сигнал на выходную клемму Y0.10 в реальном времени, выход остается включенным.



**Пример 2:**

Выход Y для инструкции DHSCS отличается от общего выхода Y.

1. Когда M0 включен, начинает выполняться инструкция DHSCS. Когда текущее значение HC200 меняется от 99 до 100 или от 101 до 100, Y0.10 немедленно выведет свое состояние на выходную клемму, независимо от цикла сканирования программы.
2. Когда текущее значение HC200 меняется от 99 до 100, контакт HC200 включается немедленно. Но когда функция SET Y0.11 выполнена, объект Y0.11 будет зависеть от цикла сканирования программы и выведет свое состояние на выходную клемму только после выполнения инструкции END .

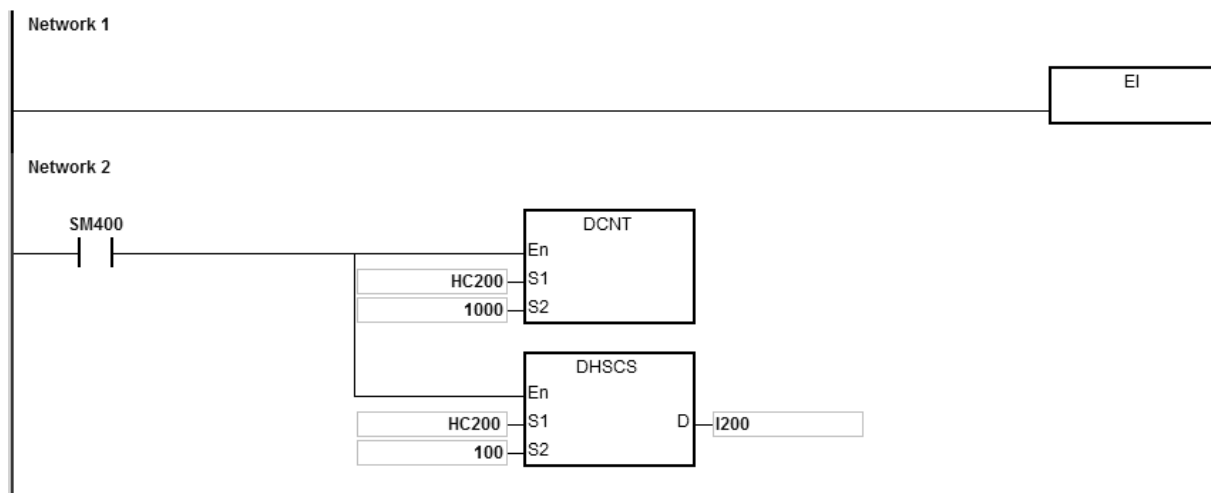


**Пример 3:**

Прерывание для аппаратного высокоскоростного сравнения:

Когда текущее значение HC200 меняется от 99 до 100 или от 101 до 100, программа переходит к указателю прерывания для выполнения подпрограммы прерывания, Y0.10 при этом включается.

Основная программа:



Программа прерывания для I200:



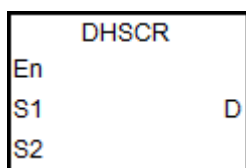
API	Код инструкции			Операнды								Функция					
1006	D	HSCR		$S_1 \cdot S_2 \cdot D$								Сброс высокоскоростного сравнения					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$							○									
$S_2$								●					○	○		
<b>D</b>		○	○	○			○	○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$												●	
$S_2$			●				●						
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

**Символ:**



$S_1$  : Номер счетчика

$S_2$  : Сравнительное значение

**D** : Результат сравнения

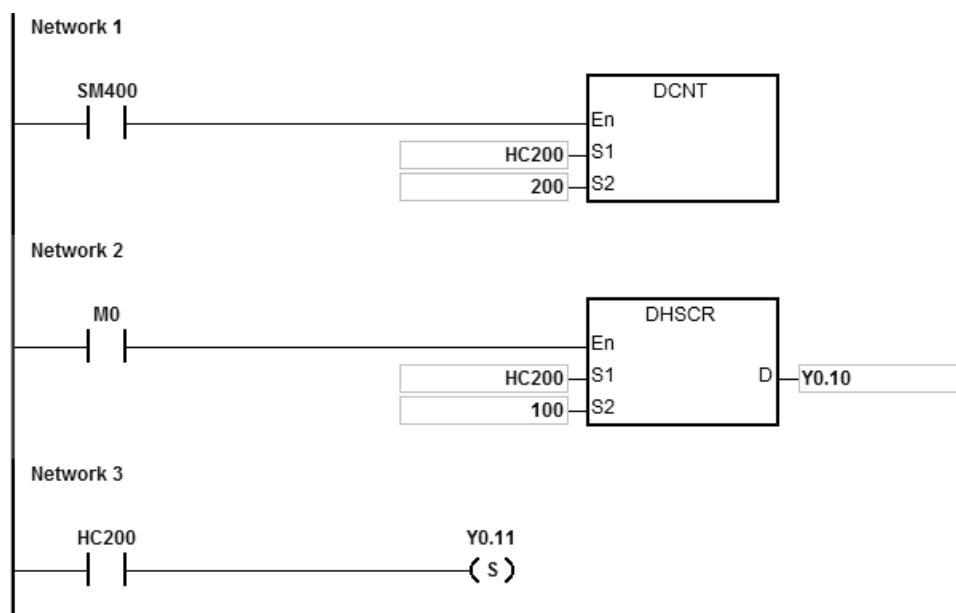
**Описание:**

1. Данная инструкция применяется с высокоскоростными счетчиками HC200 и выше по номеру. Если значение в высокоскоростном счетчике, заданном в  $S_1$  увеличивается или уменьшается на 1, инструкция HSCR немедленно проводит сравнение. Если текущее значение в счетчике равно сравнительному значению в  $S_2$ , контакт, заданный с операндом **D**, выключается. Этот контакт будет оставаться выключенным, даже если результат сравнения изменится на неравенство.
2. Если для **D** заданы выходы Y0.0 ~ Y0.15, то при равенстве значений в  $S_2$  и текущего значения высокоскоростного счетчика, на выходные клеммы Y0.0 ~ Y0.15 будет выводиться сигнал. На другие объекты типа Y будут влиять результаты цикла сканирования. Но все эти объекты обновляются немедленно и их состояние не зависит от цикла сканирования.
3. Операнд **D** также может указать, что значение счетчика HC было сброшено, ограничение операции существует только по условию совпадения номера счетчика с заданным в операнде  $S_1$ .
4. См. описание инструкции DHSCS для дополнительной информации.

**Пример 1:**

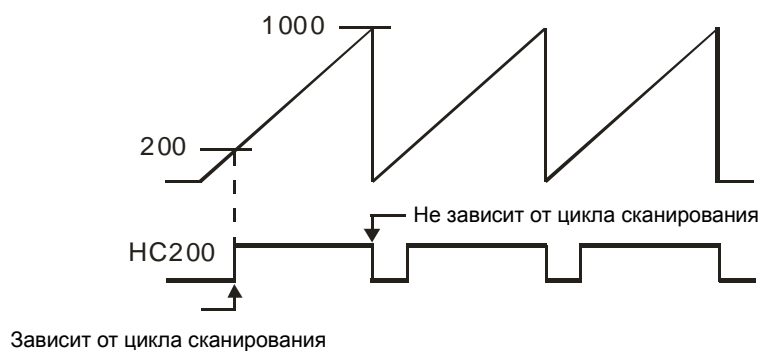
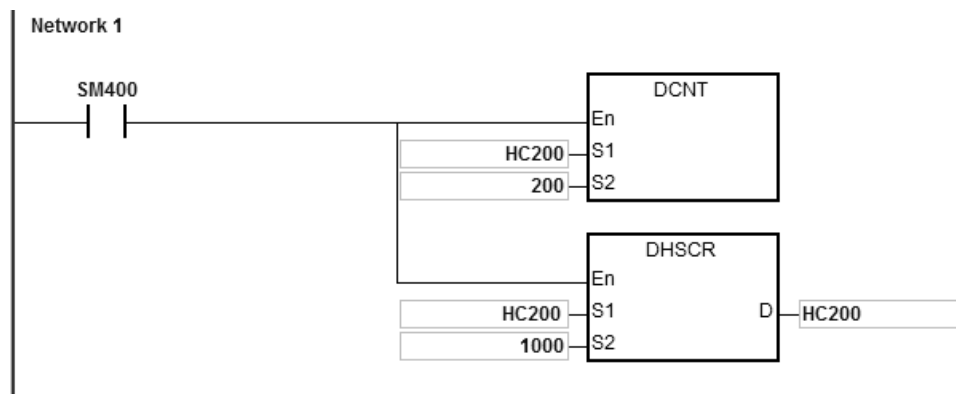
1. Когда текущее значение HC200 меняется от 99 до 100 или от 101 до 100, Y0.10 выключается.

2. Когда значение HC200 меняется от 199 до 200, включается заданный контакт для HC200, Y0.11 также включается. Выходной сигнал будет задержан до выполнения цикла сканирования программы.



**Пример 2:**

Для HC200, указанного в качестве аппаратного высокоскоростного счетчика с тем же номером, заданный контакт HC200 отключен при изменении текущего значения HC200 от 999 до 1000 или от 1001 до 1000.





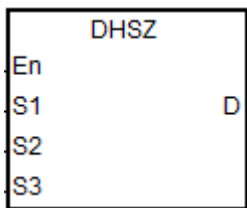
API	Код инструкции			Операнды								Функция					
1007	D	HSZ		$S_1 \cdot S_2 \cdot S_3 \cdot D$								Зонное сравнение высокоскоростных входов					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$							○									
$S_2$								●					○	○		
$S_3$								●					○	○		
D		○	○	○				○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$												●	
$S_2$			●				●						
$S_3$			●				●						
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символ:



- $S_1$  : Номер счетчика
- $S_2$  : Нижняя граница зоны сравнения
- $S_3$  : Верхняя граница зоны сравнения
- D : Результат сравнения (3 последовательных регистра)

Описание:

- Данная инструкция применяется с высокоскоростными счетчиками HC200 и выше по номеру. Нижняя граница в  $S_2$  должна быть меньше верхней границы в  $S_3$ . Если параметры зоны сравнения настроены ненадлежащим образом, ПЛК настроит их автоматически.
- Если  $S_1$  определен как программный счетчик и указанный счетчик изменяет значение увеличением или уменьшением на 1, инструкция DHSZ проводит сравнение немедленно. Результат сравнения и состояние контакта счетчика показаны ниже.

Условие сравнения	Состояние D+0	Состояние D+1	Состояние D+2
Значение в $S_1 <$ Нижняя граница в $S_2$	ВКЛ	ВЫКЛ	ВЫКЛ
Нижняя граница в $S_2 \leq$ Значение в $S_1 <$ Верхняя граница в $S_3$	ВЫКЛ	ВКЛ	ВЫКЛ
Значение в $S_1 \geq$ Верхняя граница в $S_3$	ВЫКЛ	ВЫКЛ	ВКЛ

Примечание: Нижняя граница в  $S_2$  должна быть меньше верхней границы в  $S_3$ . Если параметры зоны сравнения настроены ненадлежащим образом, ПЛК настроит их автоматически.

3. Если  $S_1$  определен как аппаратный счетчик и значение счетчика достигает нижней границы зоны сравнения, заданной в  $S_2$  или верхней границы, заданной в  $S_3$ , инструкция DHSZ немедленно начинает сравнение в соответствии с направлением счета (вверх / вниз). Результаты сравнения и состояние контакта счетчика показаны ниже.

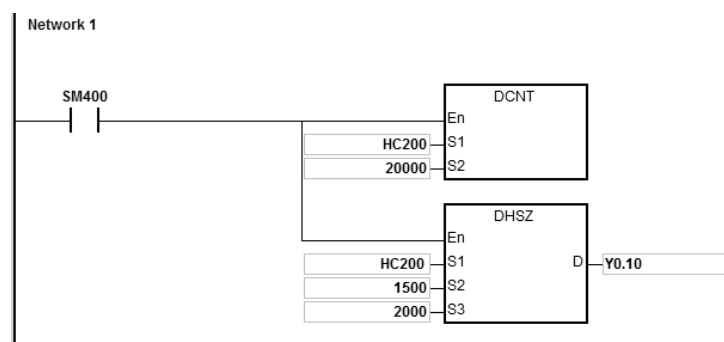
Направление счета	Условие сравнения	Состояние D+0	Состояние D+1	Состояние D+2
Счет вверх	Значение в $S_1 \geq$ Нижняя граница в $S_2$	ВЫКЛ	ВКЛ	ВЫКЛ
	Значение в $S_1 \leq$ Верхняя граница в $S_3$	ВЫКЛ	ВЫКЛ	ВКЛ
Счет вниз	Значение в $S_1 \geq$ Нижняя граница в $S_2$	ВКЛ	ВЫКЛ	ВЫКЛ
	Значение в $S_1 \leq$ Верхняя граница в $S_3$	ВЫКЛ	ВКЛ	ВЫКЛ

4. Если для  $D$  заданы выходы  $Y0.0 \sim Y0.15$ , то, в зависимости от результата сравнения, на выходные клеммы  $Y0.0 \sim Y0.15$  будет выводиться сигнал. На другие объекты типа  $Y$  будут влиять результаты цикла сканирования. Но все эти объекты обновляются немедленно и их состояние не зависит от цикла сканирования.
5. См. описание инструкции DHSCS для дополнительной информации по высокоскоростному зонному сравнению.

**Пример:**

1. Когда  $D$  задан как  $Y0.10$ ,  $Y0.11 \sim Y0.12$  занимают автоматически.
2. Инструкция DHSZ сравнивает текущее значение в HC200 с нижней/верхней границами зоны сравнения (1500/2000),  $Y0.10 \sim Y0.12$  включаются в зависимости от результатов сравнения.
3. Если текущее значение  $HC200 < 1500$ , включается  $Y0.10$ . When  $1500 \leq$  текущее значение в  $HC200 < 2000$ , включается  $Y0.11$ .

Если текущее значение в  $HC200 \geq 2000$ , включается  $Y0.12$ .



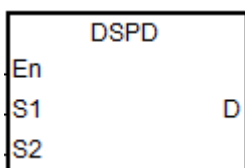
API	Код инструкции			Операнды								Функция					
1008	D	SPD		$S_1 \cdot S_2 \cdot D$								Измерение скорости					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$							○									
$S_2$								○	○				○	○		
<b>D</b>								○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$												●	
$S_2$			●				●						
<b>D</b>			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

**Символ:**



$S_1$  : Значение счетчика

$S_2$  : Заданное значение времени цикла

**D** : Измеренное значение скорости

**Описание:**

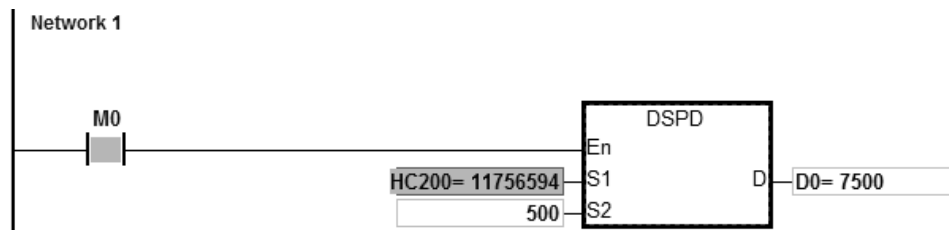
1. При выполнении этой инструкции измерения скорости,  $S_1$  следует использовать с инструкцией DCNT для включения высокоскоростного счетчика с номерами до HC200 (включая HC200).
2.  $S_2$  – это заданное значение времени цикла сканирования в миллисекундах (мс). Диапазон задания: 10~1000. Если значение выходит за пределы этого диапазона, ПЛК обозначит его как минимальное или максимальное значение, и сообщение об ошибке возникать не будет.
3. При достижении заданного в  $S_2$  значения, данная инструкция сохранить число импульсов в **D**. Вследствие этого, цикл сканирования ПЛК на данный процесс не влияет.
4. Эта инструкция позволяет одновременно выполнять 8 наборов инструкций по обнаружению скорости. 9-й или большие наборы инструкции определения скорости будут проигнорированы, сообщений об ошибке формироваться не будет. При выполнении этой инструкции будут записаны заданные параметры операнда. Таким образом, во время выполнения этой инструкции редактирование параметров не допускается.

**Пример:**

Если на X0.0 поступает импульсный входной сигнал, для определения скорости может использоваться инструкция DSPD. Когда M0 включен, DSPD будет фиксировать количество импульсов, подсчитанных

счетчиком HC200, в D0 каждые 500 мс.

В следующем примере значение в D0 равно 7500, а фактическая входная частота импульсов на вход X0.0 составляет 15 кГц (7500/500 мс).



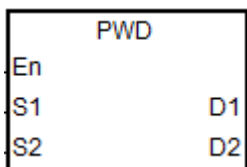
API	Код инструкции			Операнды								Функция					
1009		PWD		$S_1 \cdot S_2 \cdot D$								Измерение ширины импульса					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	○															
$S_2$								○	○				○	○		
$D_1$								○								
$D_2$		○	○	○												

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$	●												
$S_2$		●				●							
$D_1$			●				●						
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



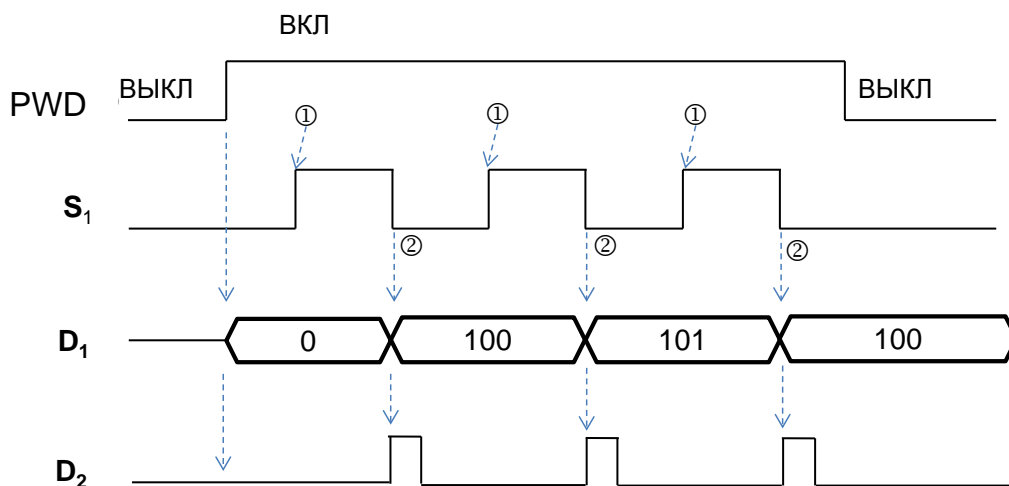
- $S_1$  : Номер входа
- $S_2$  : Единица измерения
- $D_1$  : Время измерения ширины импульса (32-битное значение)
- $D_2$  : Флаг обновления

Описание:

- $S_1$  поддерживает следующие 8 входов: X0.0/X0.1/X0.2/X0.3/X0.4/X0.6/X0.8/X0.10.  $S_1$  не поддерживает одни и те же входы с высокоскоростным счетчиком.
- $S_2$  это единица измерения. Инструкция не будет выполняться, если значение  $S_2$  отличается от представленных в таблице ниже.

Код $S_2$	Единица измерения	Диапазон измерения	Диапазон частоты	Примечание
0	1 мкс	Импульс 	1 ~ 10 кГц	
1	1 мс		0.02 ~ 100 Гц	
2	10 нс		10 Гц ~ 1 МГц	Не поддерживает X0.1 и X0.3.
4	1 мкс	Цикл 	1 Гц ~ 10 кГц	
5	1 мс		0.02 ~ 100 Гц	
Другие		Инструкция PWD не выполняется		

3.  $D_1$  используется для сохранения времени измерения ширины импульса (32-битное значение), диапазон измерения составляет 0 ~ 100 000 000. Если значение превышает максимальное значение, оно будет считаться максимальным значением. Если значение равно 0, это означает, что во время выполнения этой инструкции вход не включался.
4.  $D_2$  является флагом обновления. Когда измерение на входе  $S_1$  завершается и процесс сканирования заканчивается, флаг обновления включается в течение одного периода цикла сканирования. С помощью флага обновления, пользователь может проверить наличие измеренного значения. Когда инструкция запускается в первый раз, флаг обновления также будет отключен.
5. Когда значение в  $S_2$  равно 0, 1 и 2, см. временную диаграмму ниже для определения таких процедур, как сохранение измеренных значений и включение флагов. Время запуска таймера – это точка включения  $S_1$  как показано знаком ① на диаграмме. Время сохранения измеренного результата – это точка отключения  $S_1$  как показано знаком ②.

Рис. 1 Режим измерения при  $S_2$  равном 0, 1 или 2

6. Когда значение в  $S_2$  равно 4 и 5, см. временную диаграмму ниже для определения таких процедур, как сохранение измеренных значений и включение флагов. Время запуска таймера – это точка включения  $S_1$  как показано знаком ① на диаграмме. Время сохранения измеренного результата – это точка включения  $S_1$  как показано знаком ②.

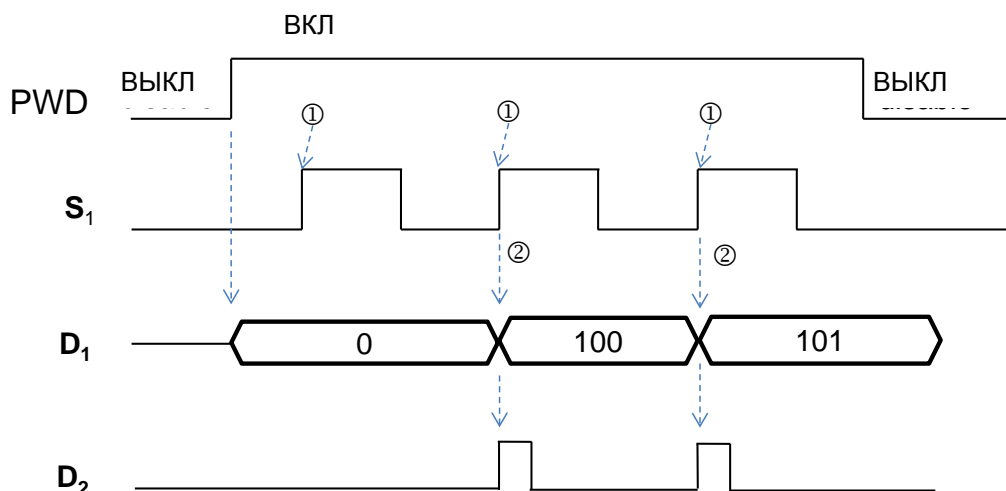


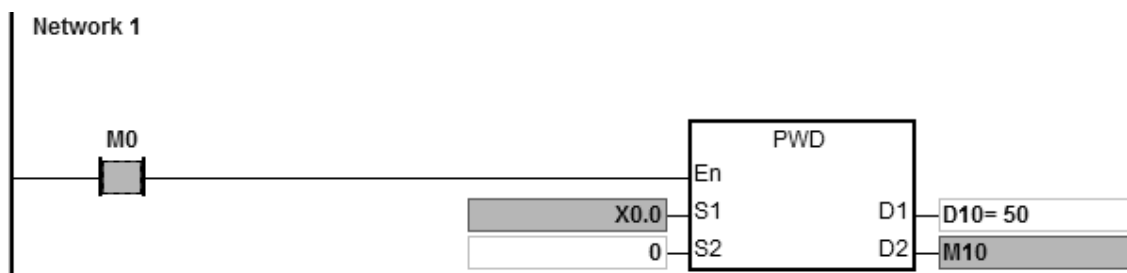
Рис. 2 Режим измерения при  $S_2$  равном 4 или 5

7. Эта инструкция позволяет одновременно выполнять 8 наборов инструкций по обнаружению скорости. 9-й или большие наборы инструкции определения скорости будут проигнорированы, сообщений об ошибке формироваться не будет. При выполнении этой инструкции будут записаны заданные параметры операнда. Таким образом, во время выполнения этой инструкции редактирование параметров не допускается.
8. Перед выполнением этой инструкции, пожалуйста, проверьте время отклика на оборудование и время импульса, установленное в HWCONFIG. Например, когда значение в  $S_2$  установлено как 0 или 2, это означает, что единица измерения – микросекунда (мкс). Значение  $S_1$  должно быть установлено как 0, чтобы отключить в HWCONFIG время фильтрации входов.

6

**Пример:**

На вход X0.0 подан импульсный сигнал частотой 10 кГц. Когда M0 включен, инструкция PWD обнаруживает входной сигнал на X0.0 и измеренная ширина импульса сохраняется в D10/D11 (32-битное значение), единица измерения задана как 0 и измеренная ширина импульса в D10 равна 50 мкс.



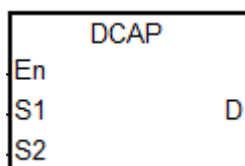
API	Код инструкции			Операнды									Функция				
1010	D	CAP		$S_1 \cdot S_2 \cdot D$									Захват высокоскоростного значения подсчета при внешнем входном прерывании				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	○															
$S_2$							○									
D								○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$	●												
$S_2$												●	
D			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символ:



- $S_1$  : Номер точки внешнего входного прерывания
- $S_2$  : Номер высокоскоростного счетчика
- D : Регистр сохранения захваченного значения

Описание:

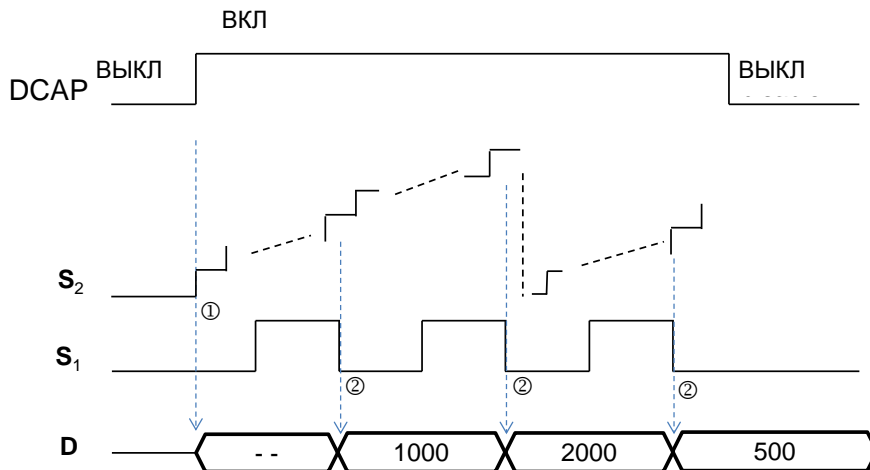
- Операнд  $S_1$  может использовать только 16 входов ПЛК: X0.0~X0.15. Они используют функцию внешнего прерывания. Операнд  $S_1$  не должен использовать один и тот же вход с высокоскоростным счетчиком.
- Высокоскоростной счетчик HC задается в  $S_2$ . Счетчик HC должен использоваться совместно с инструкцией DCNT для запуска функции подсчета.
- При осуществлении прерывания захваченное значение высокоскоростного счетчика сохраняется в 32-битном операнде D. Время сохранения данных – это момент прерывания, оно не зависит от цикла сканирования ПЛК.

Порядок работы с инструкцией показан ниже: (например, что входное прерывание запускается по заднему фронту).

① → Когда начинается выполнение инструкции, значение в D не изменится, и пользователь может ввести значение по умолчанию.

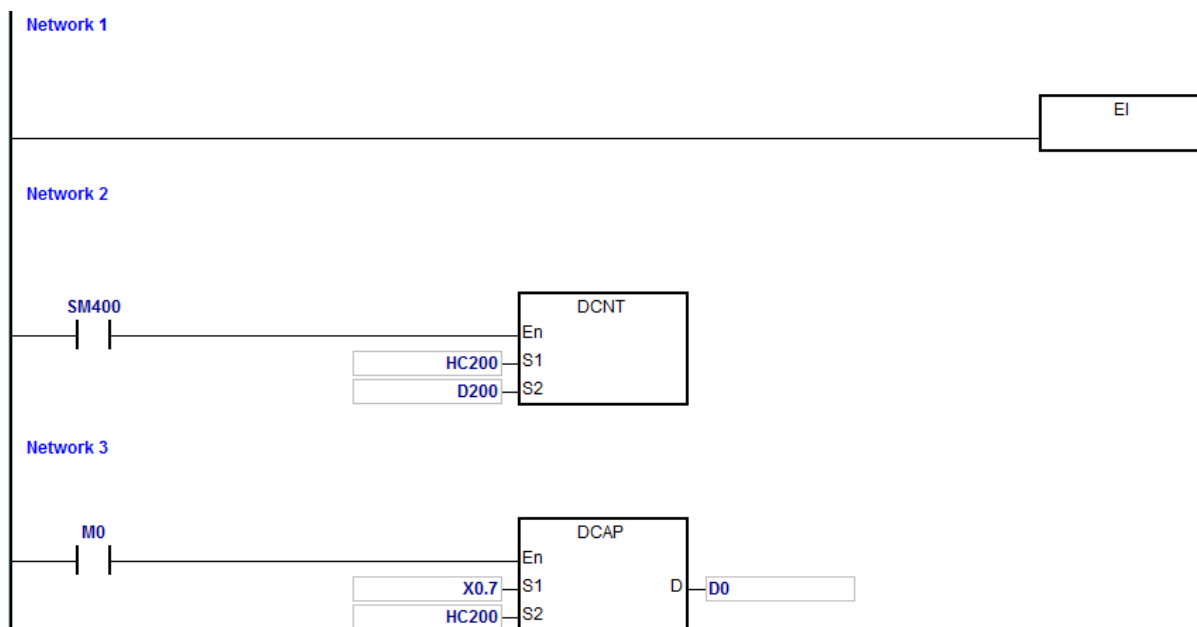
② → при осуществлении прерывания  $S_1$  значение в счетчике, заданном в  $S_2$ , захватывается немедленно и сохраняется в D.





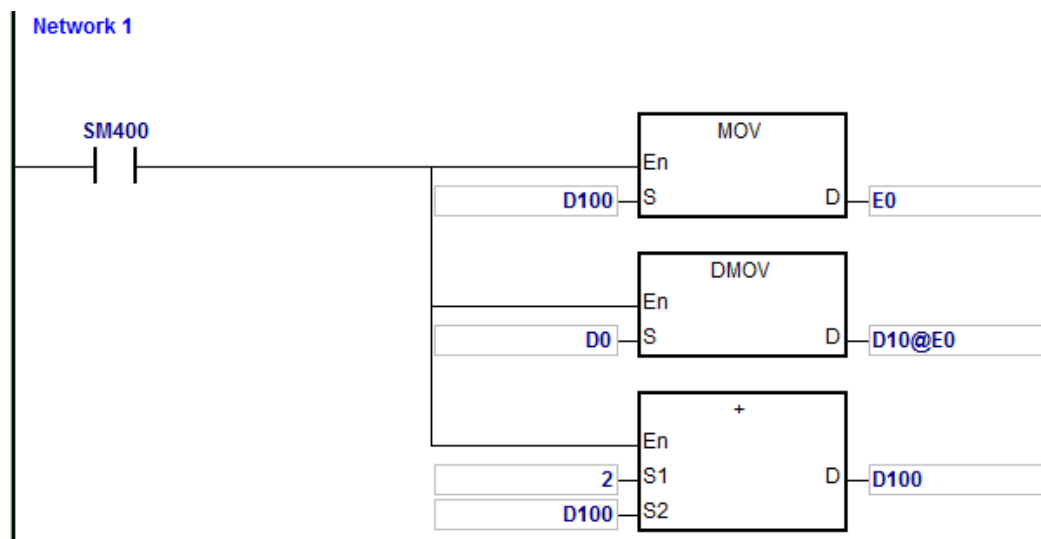
4. Инструкции DCAP могут работать с четырьмя разными входами. Если один вход задан для внешнего прерывания, инициируемого по переднему или заднему фронту, время захвата значения – это точка активации входа по переднему или заднему фронту соответственно, значение подсчета сохраняется в заданном операнде **D**. Когда две инструкции указывают один и тот же вход для прерывания, использовать вход для прерывания будет инструкция, которая запускается первой.
5. Номер счетчика HC задается в **S<sub>2</sub>**. Рекомендуются высокоскоростные счетчики в диапазоне HC200–HC255. Для более подробной информации см. описание инструкции API1004 DCNT.

**Пример:**

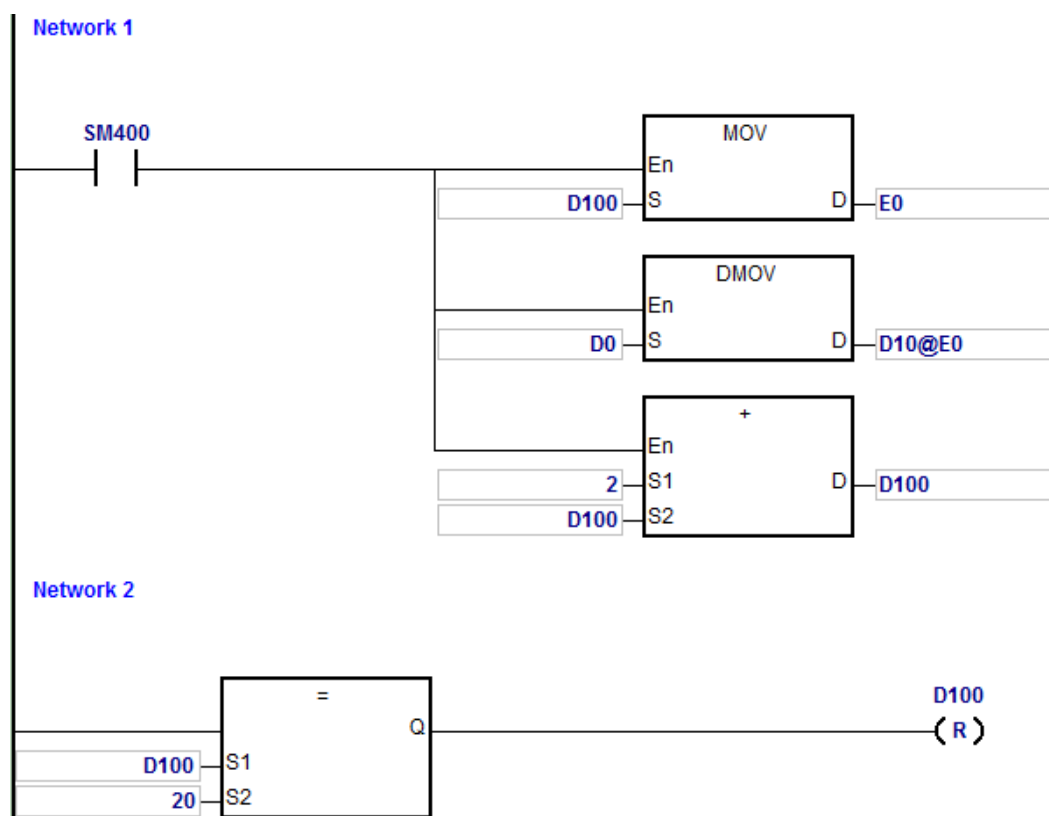


6

Внешнее прерывание активируется по переднему фронту в X0.7.



Внешнее прерывание активируется по заднему фронту в X0.7.



Примечание:

1. Когда M0 включен, выполняется инструкция DCAP. При активации внешнего прерывания на входе X0.7, значение в счетчике HC200 захватывается и сохраняется в 32-битном операнде D0.
2. Если прерывание срабатывает по переднему фронту, E0 изменяет значение на 0 путем задания значения в D100, подсчитанное значение в D0 сохраняется в D10 путем изменения E0 и значение в D100 равно 0+2.

3. Если прерывание срабатывает по заднему фронту, 0 изменяет значение на 2 путем задания значения в D100, подсчитанное значение ( $10+E0=12$ ) в D0 сохраняется в D12 путем изменения E0 и значение в D100 равно D100 равно 0+2. Когда значение в D100 равно 20, D100 сбрасывается на 0.
4. Если внешнее прерывание активируется по переднему и заднему фронту по пять раз соответственно, то захват значения будет выполняться 10 раз, а захваченные значения будут сохранены в D10, D12 ... D28.

1<sup>е</sup> захваченное значение= D10

2<sup>е</sup> захваченное значение= D12

...

10<sup>е</sup> захваченное значение=D28

11<sup>е</sup> захваченное значение=D10

API	Код инструкции				Операнды								Функция				
1011		TMRM			S <sub>1</sub> · S <sub>2</sub>								16-битный таймер (шаг: 10 мс)				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S <sub>1</sub>					○											
S <sub>2</sub>								○				○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>											●		
S <sub>2</sub>		●				●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**

TMRM
En
S1
S2

- S<sub>1</sub> : Номер таймера
- S<sub>2</sub> : Заданное значение таймера

**Описание:**

Инструкция использует в качестве шага значение 10 мс.

См. описание инструкции API1002 TMRH.

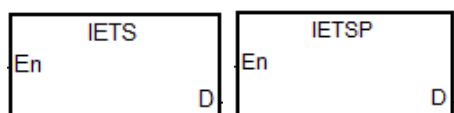
API	Код инструкции			Операнды								Функция				
1012		IETS		D								Старт инструкции измерения времени				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



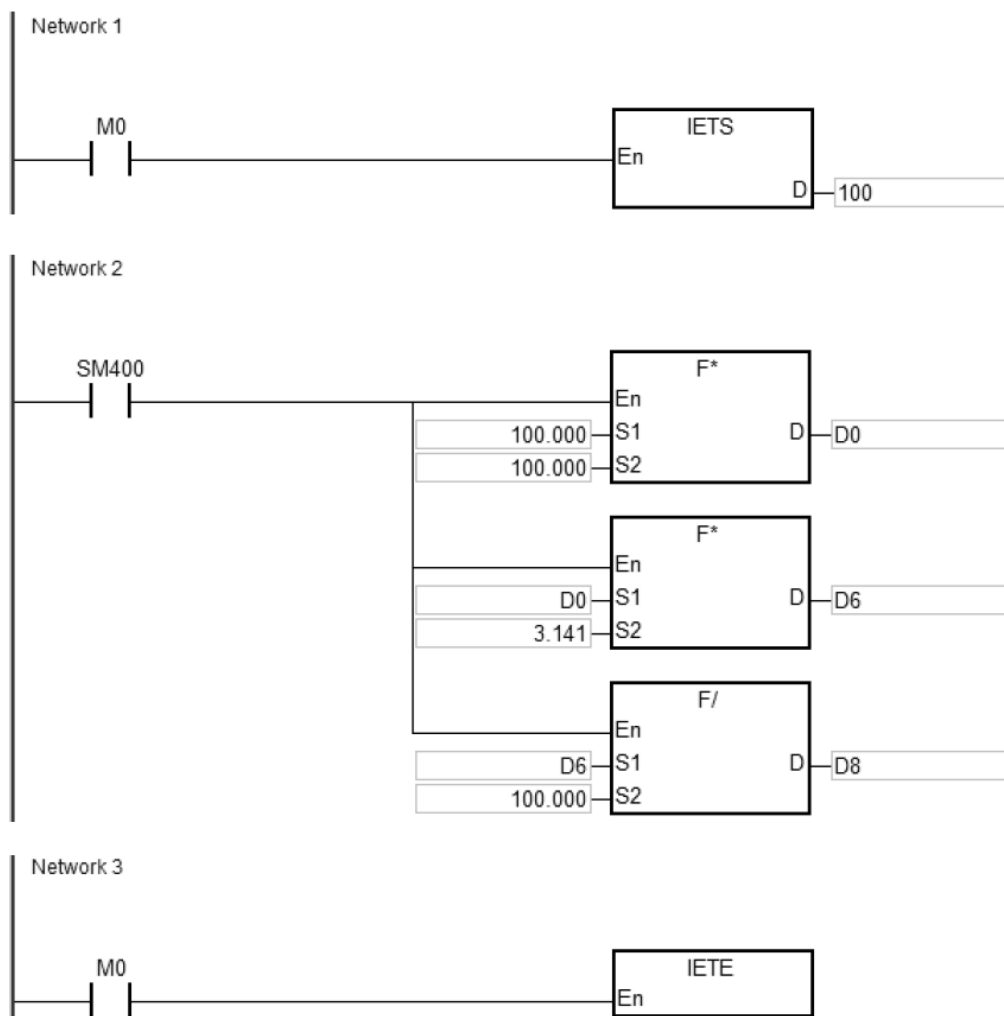
**D** : Результат измерения времени

**Описание:**

1. Инструкция IETS используется только вместе с инструкцией IEEE API1013 для измерения времени в программе ПЛК. Единица измеренного времени равна 1 мкс.
2. Когда работает инструкция IETS включена, запуск измерения времени начинается сразу же, и продолжается пока не будет выполнена инструкция IETE. Результат измерения сохраняется в **D**.
3. Диапазон измерения времени: от 0 до 32767 мкс. Когда инструкция IETS запущена, ПЛК автоматически завершит измерение времени и сохранит результат измерения в операнде **D**, при условии отсутствия активных инструкций IETE и достижения программы ПЛК команды **END**.
4. Для инструкций IETS и IETE нет ограничений на то, сколько раз они прописаны в программе. Но каждый раз при выполнении сканирования программы может быть активирован только один набор IETS и IETE, когда выполняется сканирование. Если IETS повторно используется для измерения времени, за начало измерения времени принимается время активации последней инструкции IETS. Напротив, если выполнение нескольких инструкций IETE полностью завершено, ПЛК принимает в качестве конечной точки момент, когда будет отключена первая инструкция IETE.
5. Инструкция IETS обычно используется для измерения времени работы программы ПЛК, например, функциональных блоков программы прерывания и т. д. Поскольку при включенной функции измерения времени ресурс ПЛК будет занят, предлагается удалить обе инструкции после завершения измерения, во избежание использования ресурса ПЛК во время нормальной его работы.

**Пример:**

Вычислим время выполнения инструкции на основе формулы для операции с числами с плавающей запятой, результат операции сохраняется в D100.



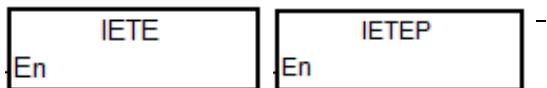
API	Код инструкции			Операнды								Функция				
1013		IETE		-								Завершение инструкции измерения времени				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
-																

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
-													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:



Описание:

Инструкция IETE должна использоваться только вместе с инструкцией IETS API1012. Для получения дополнительной информации см. описание инструкции API 1012.

## 6.12 Инструкции сдвига

### 6.12.2 Описание инструкций сдвига

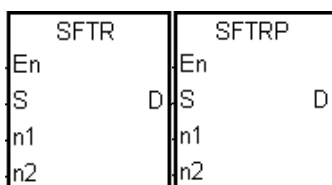
API	Код инструкции			Операнд								Функция				
1100		SFTR	P	S, D, n <sub>1</sub> , n <sub>2</sub>								Сдвиг состояния регистров вправо				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●	●	●				●								
D		●	●	●				●								
n <sub>1</sub>								●	●		○	○	○	○		
n <sub>2</sub>								●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												
D	●												
n <sub>1</sub>		●			●	●							
n <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



- S** : Начальный регистр, в котором выполняется сдвиг значения
- D** : Начальный регистр, в котором выполняется сдвиг значения
- n<sub>1</sub>** : Длина сдвигаемых данных
- n<sub>2</sub>** : Количество бит в группе

Описание:

- Состояния **n<sub>1</sub>**-битных регистров, начинающихся с операнда **D**, делятся по группам (**n<sub>2</sub>**-битов в группе), и эти группы сдвигаются вправо. Состояния бит **n<sub>2</sub>**-битных регистров, начинающихся с операнда **S**, сдвигаются к регистрам, начинающимся с операнда **D**, чтобы заполнить освободившееся место.
- Как правило, используется импульсная инструкция SFTRP.
- Операнд **n<sub>1</sub>** может иметь значение в диапазоне от 1 до 1024. Операнд **n<sub>2</sub>** может иметь значение в диапазоне от 1 до **n<sub>1</sub>**.

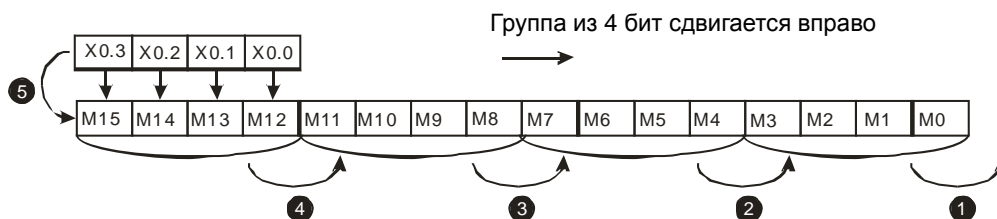
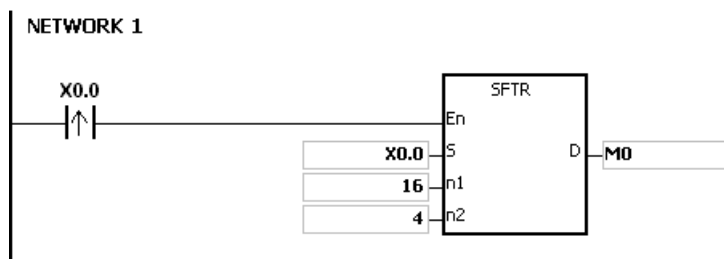
Пример 1:

- Если состояние входа X0.0 меняется с OFF на ON, состояния 16-битных регистров от M0 до M15 делятся по группам (по четыре бита в группе), и эти группы сдвигаются вправо.



2. Ниже приведен пример сдвига состояний битных регистров вправо при сканировании.

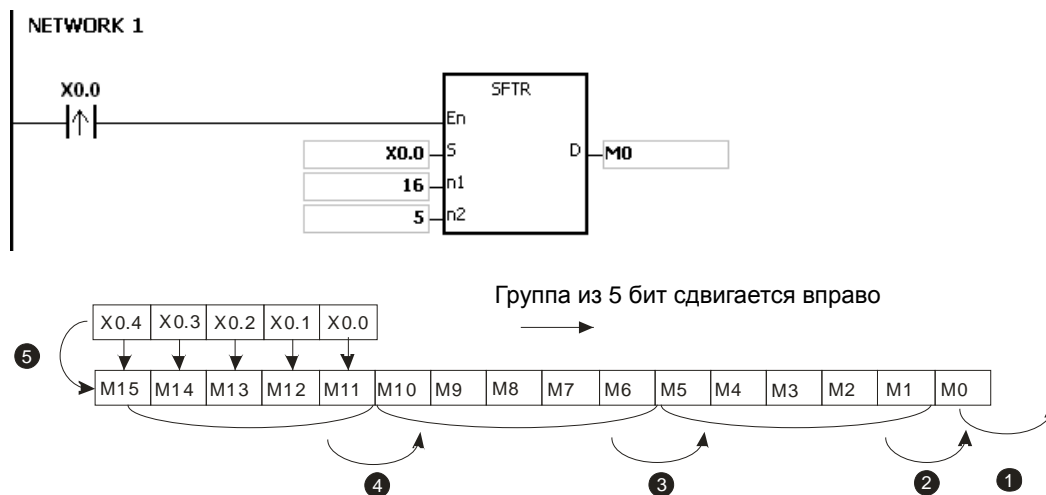
- ❶ M3~M0 → Перенос
- ❷ M7~M4 → M3~M0
- ❸ M11~M8 → M7~M4
- ❹ M15~M12 → M11~M8
- ❺ X0.3~X0.0 → M15~M12



**Пример 2:**

1. Если состояние входа X0.0 меняется с OFF на ON, состояния 16-битных регистров от M0 до M15 делятся по группам (по пять битов в группе), и эти группы сдвигаются вправо.
2. Ниже приведен пример сдвига состояний битных регистров вправо при сканировании.

- ❶ M0 → Перенос
- ❷ M5 → M0
- ❸ M10~M6 → M5~M1
- ❹ M15~M11 → M10~M6
- ❺ X0.4~X0.0 → M15~M11



**Дополнительные замечания:**

1. Если  $S+n_2-1$  или  $D+n_1-1$  больше допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n_1$  меньше 1 или значение  $n_1$  больше 1024, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
3. Если значение  $n_2$  меньше 1 или значение  $n_2$  больше, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.

API	Код инструкции			Операнд								Функция				
1101		SFTL	P	S, D, n <sub>1</sub> , n <sub>2</sub>								Сдвиг состояния регистров влево				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●	●	●				●								
D		●	●	●				●								
n <sub>1</sub>								●	●		○	○	○	○		
n <sub>2</sub>								●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												
D	●												
n <sub>1</sub>		●			●	●							
n <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

SFTL		SFTLP	
En		En	
S	D	S	D
n <sub>1</sub>		n <sub>1</sub>	
n <sub>2</sub>		n <sub>2</sub>	

- S** : Начальный регистр, в котором выполняется сдвиг значения
- D** : Начальный регистр, в котором выполняется сдвиг значения
- n<sub>1</sub>** : Длина сдвигаемых данных
- n<sub>2</sub>** : Количество бит в группе

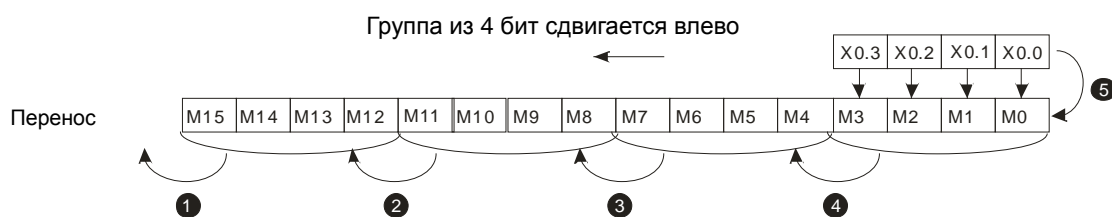
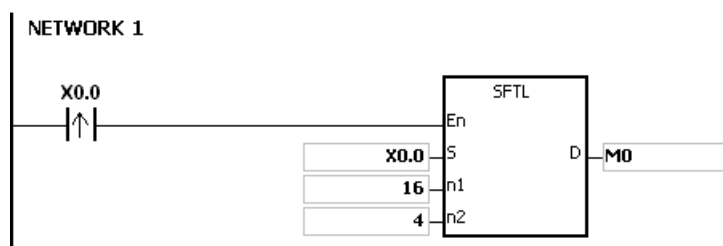
**Описание:**

1. Состояния n<sub>1</sub>-битных регистров, начинающихся с операнда **D**, делятся по группам (n<sub>2</sub>-битов в группе), и эти группы сдвигаются влево. Состояния бит n<sub>2</sub>-битных регистров, начинающихся с операнда **S**, сдвигаются к регистрам, начинающимся с операнда **D**, чтобы заполнить освободившееся место.
2. Как правило, используется импульсная инструкция SFTLP.
3. Операнд n<sub>1</sub> может иметь значение в диапазоне от 1 до 1024. Операнд n<sub>2</sub> может иметь значение в диапазоне от 1 до n<sub>1</sub>.

**Пример 1:**

1. Если состояние входа X0.0 меняется с OFF на ON, состояния 16-битных регистров от M0 до M15 делятся по группам (по четыре бита в группе), и эти группы сдвигаются влево.
2. Ниже приведен пример сдвига состояний битных регистров влево при сканировании.

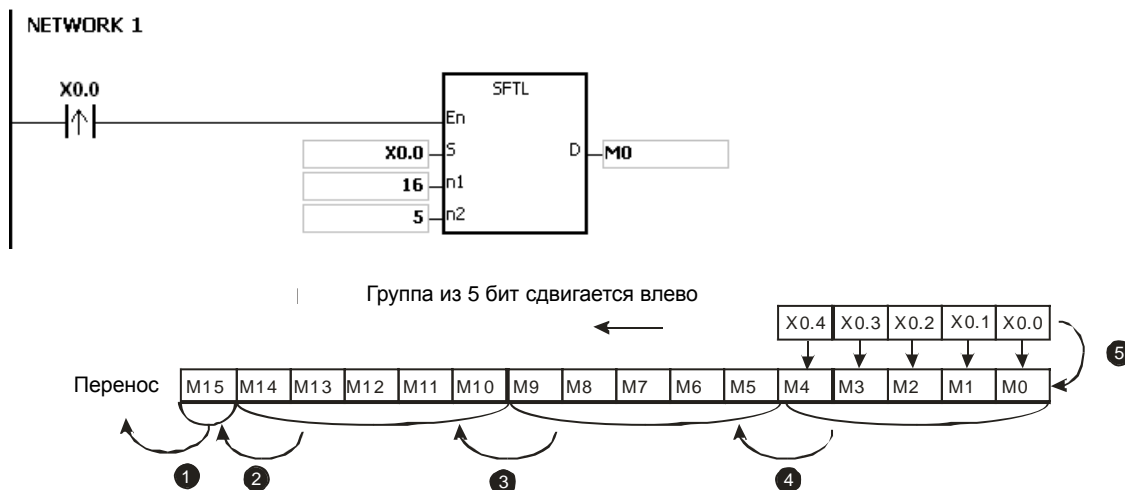
- ① M15~M12 → Перенос
- ② M11~M8 → M15~M12
- ③ M7~M4 → M11~M8
- ④ M3~M0 → M7~M4
- ⑤ X0.3~X0.0 → M3~M0



**Пример 2:**

1. Если состояние входа X0.0 меняется с OFF на ON, состояния 16-битных регистров от M0 до M15 делятся по группам (по пять бит в группе), и эти группы сдвигаются влево.
2. Ниже приведен пример сдвига состояний битных регистров влево при сканировании.

- ① M15 → Перенос
- ② M10 → M15
- ③ M9~M5 → M14~M10
- ④ M4~M0 → M9~M5
- ⑤ X0.4~X0.0 → M4~M0



**Дополнительные замечания:**

1. Если  $S+n_2-1$  или  $D+n_1-1$  больше допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n_1$  меньше 1 или значение  $n_1$  больше 1024, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
3. Если значение  $n_2$  меньше 1 или значение  $n_2$  больше, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.

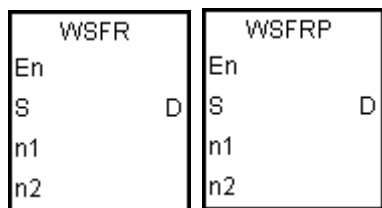
API	Код инструкции			Операнд								Функция				
1102		WSFR	P	S, D, n <sub>1</sub> , n <sub>2</sub>								Сдвиг данных в словных регистрах вправо				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
D		●			●	●		●								
n <sub>1</sub>								●	●		○	○	○	○		
n <sub>2</sub>								●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n <sub>1</sub>		●			●	●							
n <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S** : Начальный регистр, в котором выполняется сдвиг значения
- D** : Начальный регистр, в котором выполняется сдвиг значения
- n<sub>1</sub>** : Длина сдвигаемых данных
- n<sub>2</sub>** : Количество бит в группе

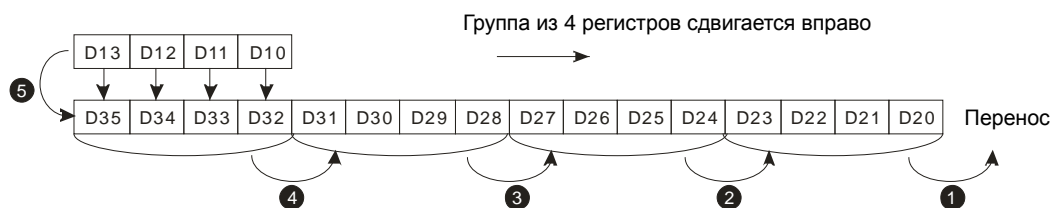
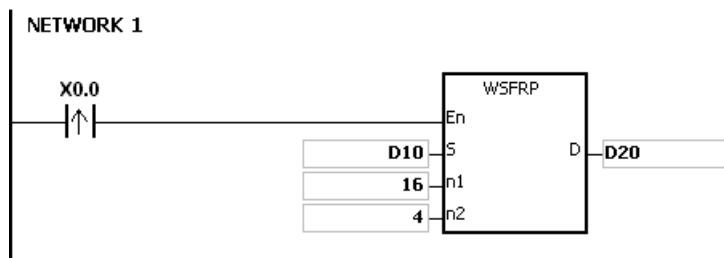
**Описание:**

- Данные в словных регистрах **n<sub>1</sub>**, начинающихся с операнда **D**, делятся по группам (**n<sub>2</sub>** слова в группе), и эти группы сдвигаются вправо. Данные в регистрах **n<sub>2</sub>** типа word, начинающихся с операнда **S**, сдвигаются к регистрам, начинающимся с операнда **D**, чтобы заполнить освободившееся место.
- Как правило, используется импульсная инструкция WSFRP.
- Операнд **n<sub>1</sub>** может иметь значение в диапазоне от 1 до 512. Операнд **n<sub>2</sub>** может иметь значение в диапазоне от 1 до **n<sub>1</sub>**.

**Пример 1:**

- Если состояние входа X0.0 меняется с OFF на ON, данные в 16-словных регистрах от D20 до D35 делятся по группам (по четыре слова в группе), и эти группы сдвигаются вправо.
- Ниже приведен пример сдвига данных регистров типа word вправо при сканировании.

- ❶ D23~D20 → Перенос
- ❷ D27~D24 → D23~D20
- ❸ D31~D28 → D27~D24
- ❹ D35~D32 → D31~D28
- ❺ D13~D10 → D35~D32

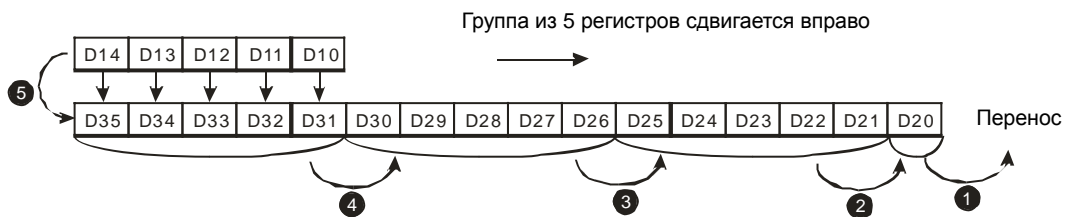
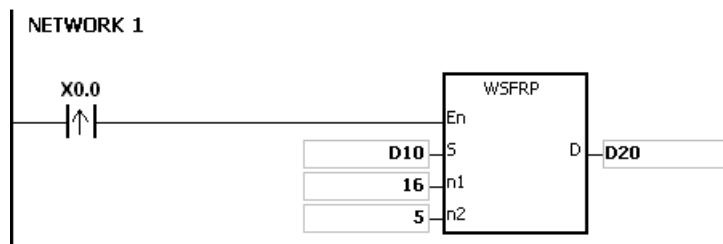


6

**Пример 2:**

1. Если состояние входа X0.0 меняется с OFF на ON, данные в 16-словных регистрах от D20 до D35 делятся по группам (по пять слов в группе), и эти группы сдвигаются вправо.
2. Ниже приведен пример сдвига данных регистров типа word вправо при сканировании.

- ❶ D20 → Перенос
- ❷ D25 → D20
- ❸ D30~D26 → D25~D21
- ❹ D35~D31 → D30~D26
- ❺ D14~D10 → D35~D31



**Дополнительные замечания:**

1. Если  $S+n_2-1$  или  $D+n_1-1$  больше допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n_1$  меньше 1 или значение  $n_1$  больше 512, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
3. Если значение  $n_2$  меньше 1 или значение  $n_2$  больше, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.



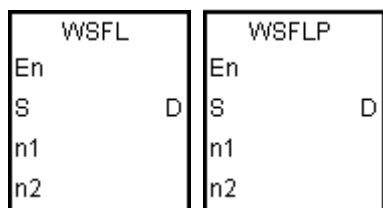
API	Код инструкции			Операнд								Функция				
1103		WSFL	P	S, D, n <sub>1</sub> , n <sub>2</sub>								Сдвиг данных в словных регистрах влево				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
D		●			●	●		●								
n <sub>1</sub>								●	●		○	○	○	○		
n <sub>2</sub>								●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n <sub>1</sub>		●			●	●							
n <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S** : Начальный регистр, в котором выполняется сдвиг значения
- D** : Начальный регистр, в котором выполняется сдвиг значения
- n<sub>1</sub>** : Длина сдвигаемых данных
- n<sub>2</sub>** : Количество бит в группе

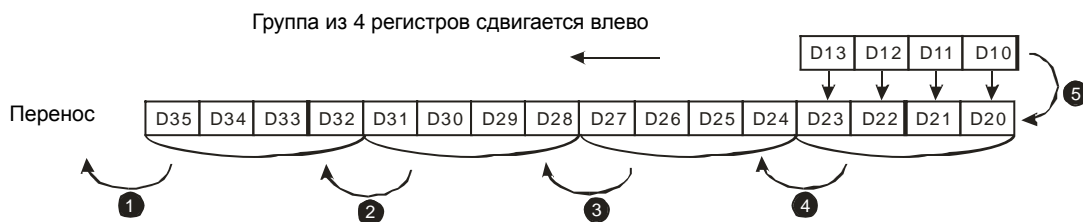
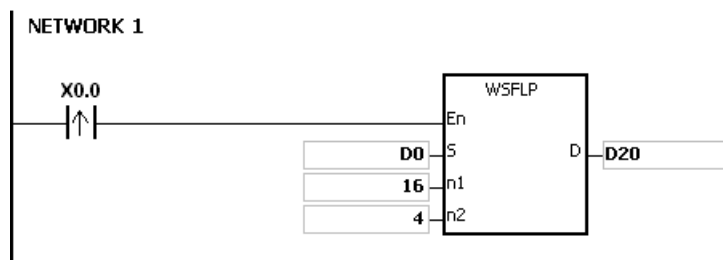
**Описание:**

- Данные в словных регистрах n<sub>1</sub>, начинающихся с операнда D, делятся по группам (n<sub>2</sub> слова в группе), и эти группы сдвигаются влево. Данные в регистрах n<sub>2</sub> типа word, начинающихся с операнда S, сдвигаются к регистрам, начинающимся с операнда D, чтобы заполнить освободившееся место.
- Как правило, используется импульсная инструкция WSFLP.
- Операнд n<sub>1</sub> может иметь значение в диапазоне от 1 до 512. Операнд n<sub>2</sub> может иметь значение в диапазоне от 1 до n<sub>1</sub>.

**Пример 1:**

- Если состояние входа X0.0 меняется с OFF на ON, данные в 16-словных регистрах от D20 до D35 делятся по группам (по четыре слова в группе), и эти группы сдвигаются влево.
- Ниже приведен пример сдвига данных регистров типа word влево при сканировании.

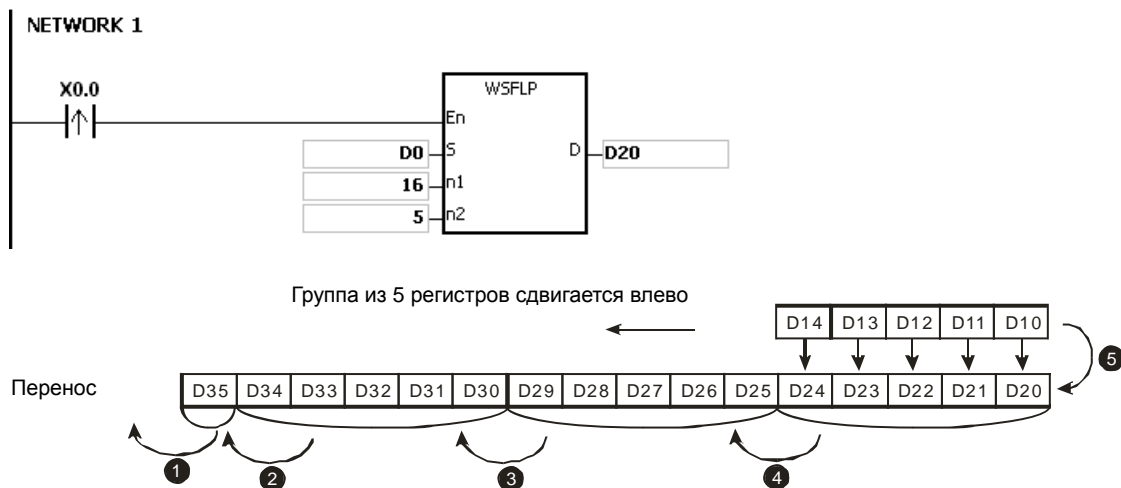
- ❶ D35~D32 → Перенос
- ❷ D31~D28 → D35~D32
- ❸ D27~D24 → D31~D28
- ❹ D23~D20 → D27~D24
- ❺ D13~D10 → D23~D20



**Пример 2:**

1. Если состояние входа X0.0 меняется с OFF на ON, данные в 16-словных регистрах от D20 до D35 делятся по группам (по пять слов в группе), и эти группы сдвигаются влево.
2. Ниже приведен пример сдвига данных регистров типа word влево при сканировании.

- ❶ D35 → Перенос
- ❷ D30 → D35
- ❸ D29~D25 → D34~D30
- ❹ D24~D20 → D29~D25
- ❺ D14~D10 → D24~D20



**Дополнительные замечания:**

1. Если  $S+n_2-1$  или  $D+n_1-1$  больше допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение  $n_1$  меньше 1 или значение  $n_1$  больше 512, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Если значение  $n_2$  меньше 1 или значение  $n_2$  больше, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

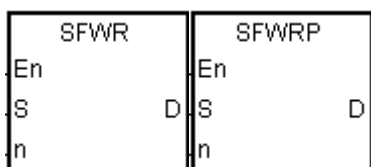
API	Код инструкции			Операнд							Функция					
1104		SFWR	P	S, D, n							Сдвиг данных и их запись в словный регистр					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



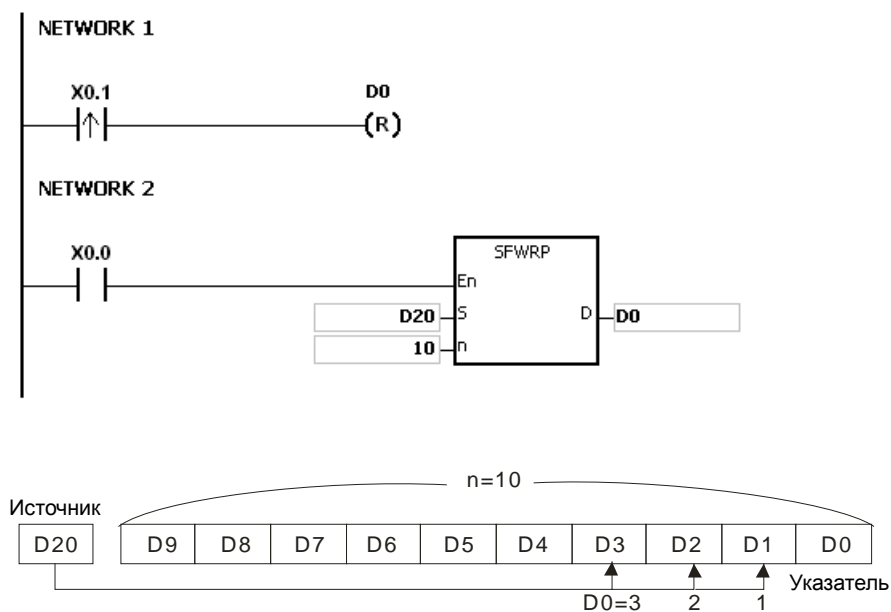
- S** : Регистр, в котором сдвигаются данные
- D** : Начальный регистр
- n** : Длина данных

**Описание:**

1. Данные в словных регистрах **n**, начинающихся с регистра, заданного операндом **D**, определяются как данные типа FIFO (first in/first out), а регистр, заданный операндом **D**, берется в качестве указателя. После выполнения инструкции значение указателя увеличивается на единицу и данные в регистре, заданном операндом **S**, записываются в регистр, заданный указателем. Если значение указателя больше или равно значению **n-1**, инструкция не выполняет запись данных и включается флаг переноса SM602.
2. Как правило, используется импульсная инструкция SFWRP.
3. Операнд **n** может иметь значение в диапазоне от 2 до 512.

**Пример:**

1. Сначала значение указателя D0 сбрасывается на 0. Если состояние входа X0.0 меняется с OFF на ON, данные из D20 записываются в D1, а D0 принимает значение 1. Если состояние входа X0.0 снова меняется с OFF на ON, данные из D20 записываются в D2, а D0 принимает значение 2.
2. Сдвиг и запись данных в регистре типа word выполняется следующим образом.
  - Данные из D20 записываются в D1.
  - D0 принимает значение 1.



**Дополнительные замечания:**

1. Если значение **D** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если **D+n-1** выходит за пределы диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если значение **n** меньше 2 или значение **n** больше 512, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
4. Инструкция SFWR может использоваться вместе с инструкцией SFRD для записи и чтения данных.

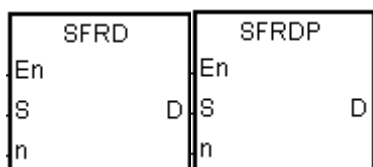
API	Код инструкции			Операнд							Функция						
1105		SFRD	P	S, D, n							Сдвиг данных и их чтение из словного регистра						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●		●			○					
D		●			●	●		●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



- S** : Начальный регистр
- D** : Регистр, в котором сдвигаются данные
- n** : Длина данных

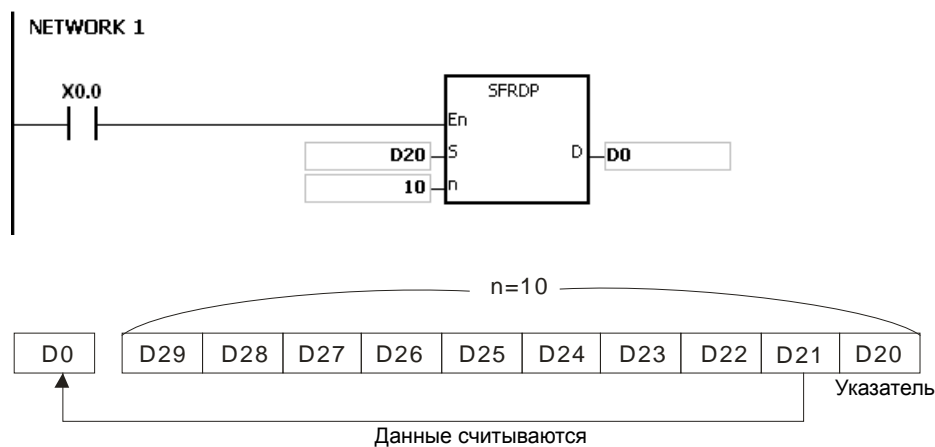
Описание:

1. Данные в словных регистрах **n**, начинающихся с регистра, заданного операндом **S**, определяются как данные типа FIFO (first in/first out), а регистр, заданный операндом **S**, берется в качестве указателя. После выполнения инструкции значение регистра, заданного операндом **S** уменьшается на единицу, данные регистра, заданного операндом **S+1** записываются в регистр, заданный операндом **D**, данные регистров, заданных операндом **S+n-1~S+2** сдвигаются вправо, а данные регистра, заданного операндом **S+n-1** не изменяются. Если значение регистра, заданного операндом **S**, равно 0, инструкция не выполняет чтение данных и включается флаг нуля SM600.
2. Как правило, используется импульсная инструкция SFRDP.
3. Операнд **n** может иметь значение в диапазоне от 2 до 512.

Пример:

1. Если состояние входа X0.0 меняется с OFF на ON, данные D21 записываются в D0, данные D29~D22 сдвигаются вправо, данные D29 не изменяются, а значение D20 уменьшается на единицу.
2. Сдвиг и чтение данных в регистре типа word выполняется следующим образом.
  - Данные D21 считываются и сдвигаются в D0.

- Данные D29~D22 сдвигаются вправо.
- Значение D20 уменьшается на единицу.



**Дополнительные замечания:**

1. Если значение **S** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **S+n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если значение **n** меньше 2 или значение **n** больше 512, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
4. Инструкция SFWR может использоваться вместе с инструкцией SFRD для записи и чтения данных.

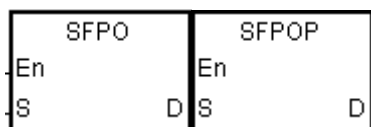
API	Код инструкции			Операнд								Функция					
1106		SFPO	P	S, D								Чтение последних данных из списка данных					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●		●								
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



**S** : Начальный регистр

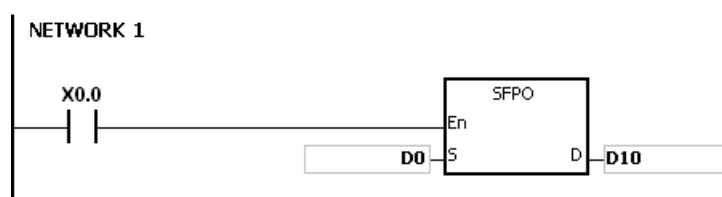
**D** : Регистр, в котором хранятся данные

Описание:

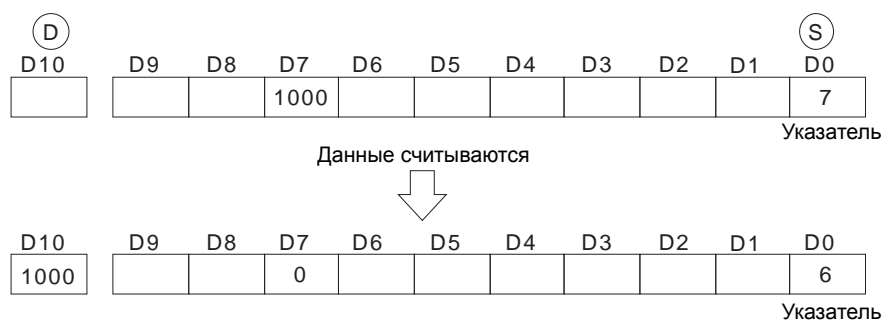
1. Регистр, заданный операндом **S**, берется в качестве указателя. После выполнения инструкции данные в регистре, заданном значением указателя, записываются в регистр, заданный операндом **D**, и сбрасываются на 0, а значение регистра, заданного операндом **S**, уменьшается на единицу. Если значение регистра, заданного операндом **S**, равно 0, инструкция не выполняет чтение данных и включается флаг нуля SM600.
2. Как правило, используется импульсная инструкция SFPOP.

Пример:

Если состояние входа X0.0 = ON, данные регистра, заданного значением D0, записываются в D10. После сдвига данных данные в регистре, заданном значением D0, сбрасываются на 0, а значение в D0 уменьшается на единицу.







**Дополнительные замечания:**

1. Если значение **S** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если **S+**(значение в операнде **S**) больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

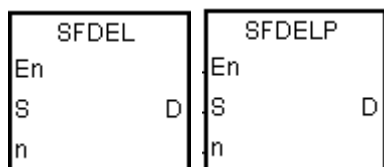
API	Код инструкции			Операнд								Функция					
1107		SFDEL	P	S, D, n								Удаление данных из списка данных					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●		●								
D		●			●	●		●			○	○				
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



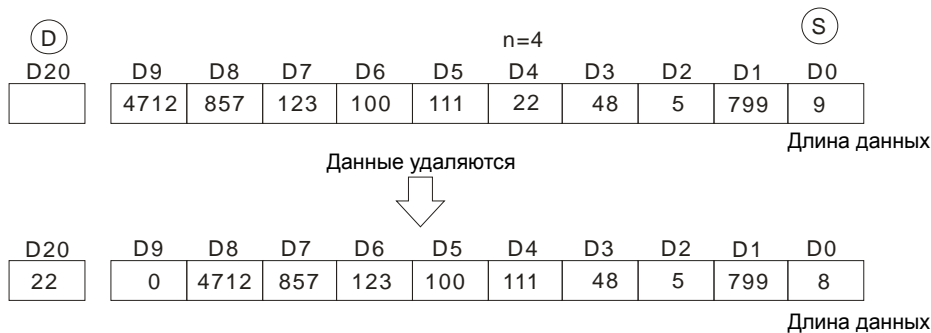
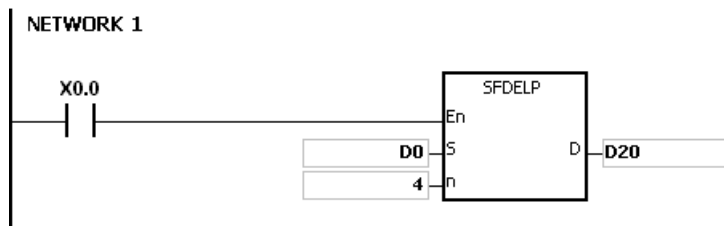
- S** : Начальный регистр
- D** : Регистр, в котором хранятся данные
- n** : Регистр, из которого удаляются данные

Описание:

1. Значение в регистре, заданном операндом **S**, обозначает длину данных, а данные находятся в регистре, заданном операндом **S+1~S+(значение операнда S)**. После выполнения инструкции данные в регистре, заданном операндом **S+n**, сохраняются в операнде **D** и удаляются; данные в регистрах, заданных операндом **S+n+1~S+(значение операнда S)**, сдвигаются вправо; данные в регистре, заданном операндом **S+(значение операнда S)**, сбрасываются на 0, а значение в регистре, заданном операндом **S**, уменьшается на единицу. Если значение регистра, заданного операндом **S**, равно 0, инструкция не выполняет удаление данных и включается флаг нуля SM600.
2. Как правило, используется импульсная инструкция SFDELP.
3. Операнд **n** может иметь значение в диапазоне от 1 до 32767.

Пример:

Предположим, что в D0 задано значение 9, а в n задано значение 4. Если состояние входа X0.0 = ON, данные D4 сохраняются в D20. После удаления данных из D4 данные D5~D9 сдвигаются вправо, а значение D0 уменьшается на единицу.



**Дополнительные замечания:**

1. Если значение **S** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **S+n** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если **S+(значение в операнде S)** больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
4. Если значение **n** больше значения параметра **S**, инструкция не выполняется, SM0=ON и у SR0 код ошибки 16#200B.
5. Если значение **n** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

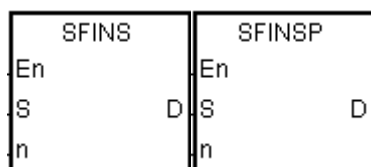
API	Код инструкции			Операнд							Функция						
1108		SFINS	P	S, D, n							Вставка данных в список данных						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●		●								
D	●	●			●	●		●	●		○	○	○	○		
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



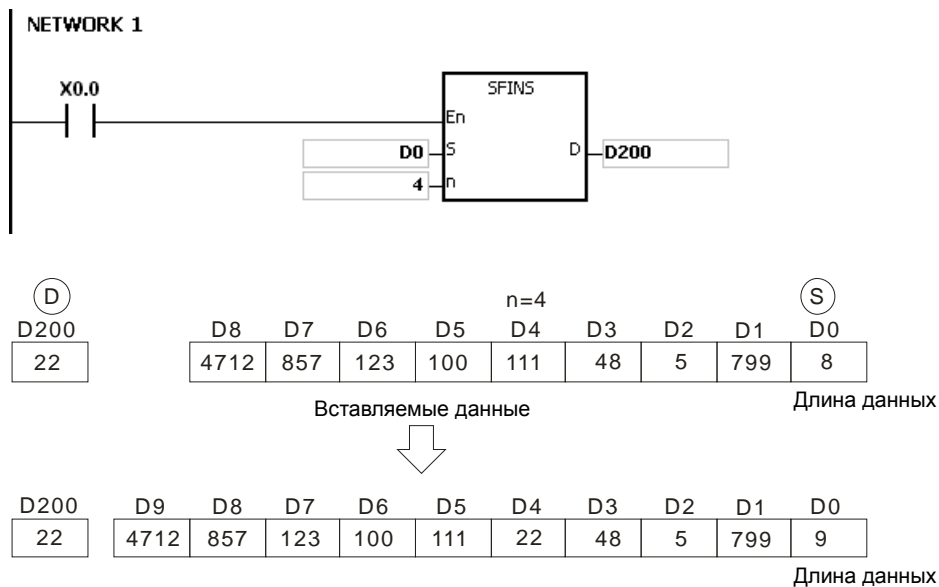
- S** : Начальный регистр
- D** : Вставляемые данные
- n** : Регистр, в который вставляются данные

**Описание:**

1. Значение в регистре, заданном операндом **S**, обозначает длину данных, а данные находятся в регистре, заданном операндом **S+1~S+(значение операнда S)**. После выполнения инструкции данные из **D** вставляются в **S+n**, исходные данные в регистре, заданном операндом **S+n~S+(значение операнда S)**, сдвигаются влево, а значение регистра, заданного операндом **S**, увеличивается на единицу. Когда значение в регистре, заданном операндом **S**, равно 32767, инструкция не выполняет запись данных, значение в регистре, заданном операндом **S**, не увеличивается и включается флаг переноса SM602.
2. Как правило, используется импульсная инструкция SFINSP.
3. Операнд **n** может иметь значение в диапазоне от 1 до 32767.

**Пример:**

Предположим, что в D0 задано значение 8, а в n задано значение 4. Если состояние входа X0.0 = ON, данные из D200 вставляются в D4, исходные данные D4~D8 сдвигаются в D5~D9, а значение D0 увеличивается на единицу.



**Дополнительные замечания:**

1. Если значение **S** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **S+n** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если **S+(значение в операнде S)+1** больше диапазона значений регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
4. Если значение **n** больше значения параметра **S**, инструкция не выполняется, SM0=ON и у SR0 код ошибки 16#200B.
5. Если значение **n** меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

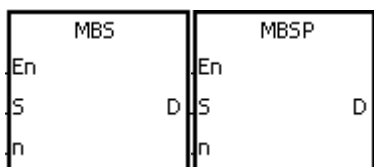
API	Код инструкции			Операнд							Функция					
1109		MBS	P	S, D, n							Сдвиг битов матрицы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●		○		○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



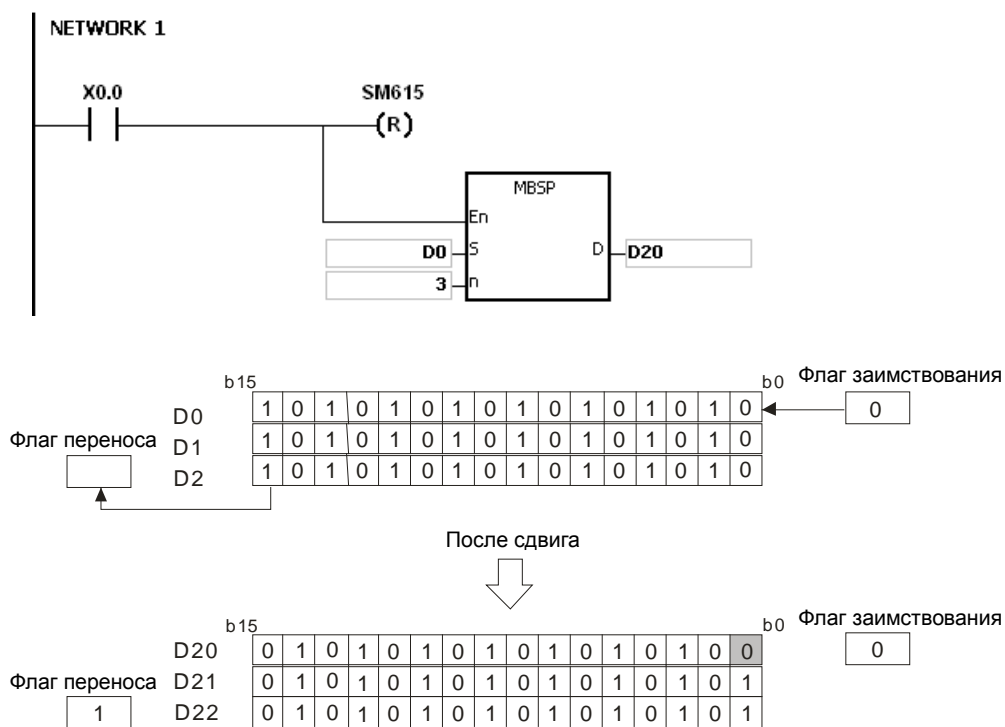
- S** : источник матрицы
- D** : результат операции
- n** : длина массива

**Описание:**

1. Значения битов n-рядов источника матрицы **S** сдвигаются вправо или влево. Если SM616 = OFF, значения битов сдвигаются влево. Если SM616 = ON, значения битов сдвигаются вправо. Место, освободившееся (сдвиг влево: b0, сдвиг вправо: b16n-1) в результате сдвига, заполняется состоянием флага заимствования SM615, значение последнего сдвинутого бита (сдвиг влево: b16n-1, сдвиг вправо: b0) передается во флаг переноса SM614, а результат выполненной операции сохраняется в **D**.
2. Операнд **n** может иметь значение в диапазоне от 1 до 256.
3. Как правило, используется импульсная инструкция MBSP.

**Пример 1:**

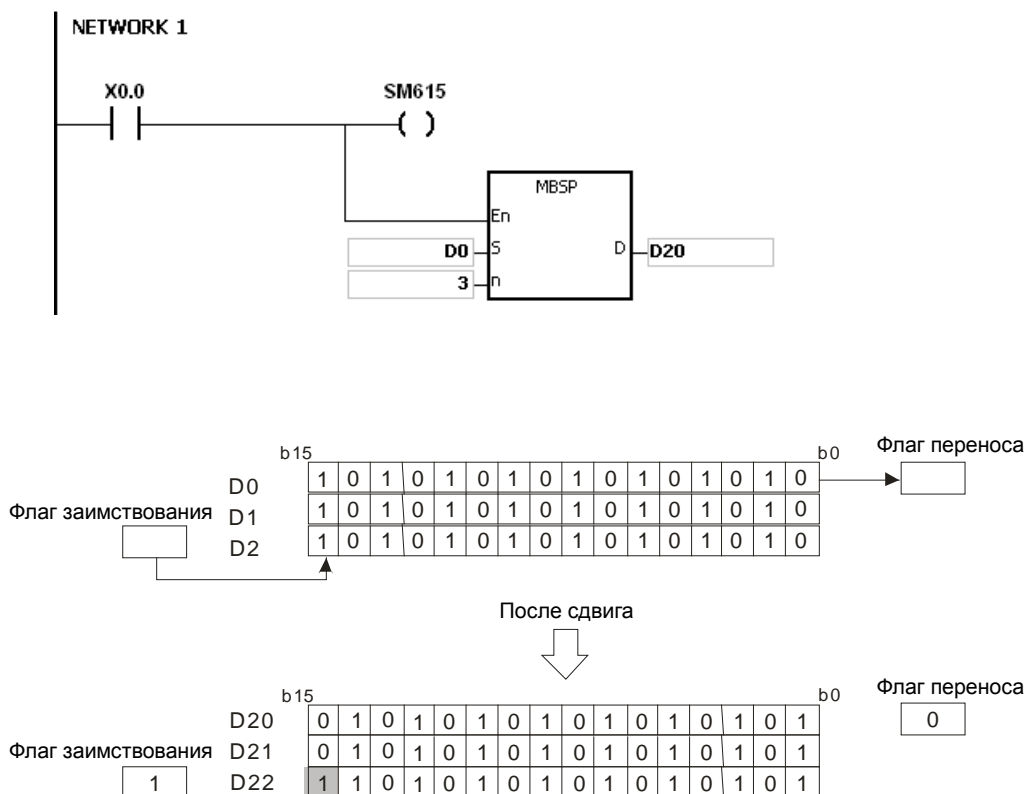
Вход X0.0 = ON и SM616 = OFF. Значения бит сдвигаются влево. Предположим, что SM615 = OFF. После сдвига значений бит в 16-битовых регистрах D0~D2 влево результат операции сохраняется в 16-битовых регистрах D20~D22, а SM614 = ON.



**Пример 2:**

Вход X0.0 = ON и SM616 = ON. Значения бит сдвигаются вправо. Предположим, что SM615 = ON. После вращения значений бит в 16-битовых регистрах D0~D2 вправо результат операции сохраняется в 16-битовых регистрах D20~D22, а SM614 = OFF.

6



**Дополнительные замечания:**

1. Если  $S+n-1$  или  $D+n-1$  больше допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Флаги:
  - SM614: флаг переноса при выдаче/сдвиге/вращении матрицы.
  - SM615: флаг заимствования при сдвиге/выдаче матрицы.
  - SM616: флаг направления при сдвиге/вращении матрицы.



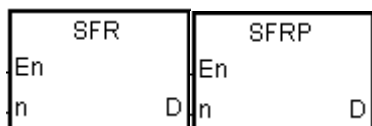
API	Код инструкции			Операнд							Функция					
1110		SFR	P	D, n							Сдвиг значений битов в 16-битном регистре на n битов вправо					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●			●	●		●			○	○				
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



**D** : Регистр, в котором происходит сдвиг  
**n** : Количество бит

**Описание:**

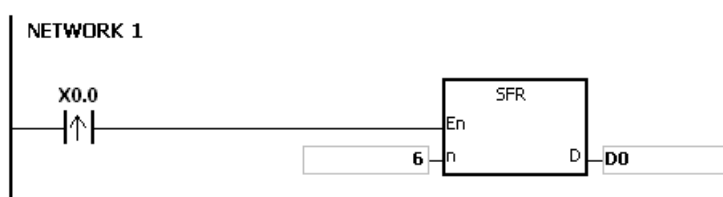
1. Значения битов в **D** сдвигаются на **n** бит вправо. Место, освободившееся (b15~b15-n+1) после сдвига, заполняется значением 0, а значение bn-1 передается в SM602.
2. Операнд **n** может иметь значение в диапазоне от 1 до 16.
3. Как правило, используется импульсная инструкция SFRP.

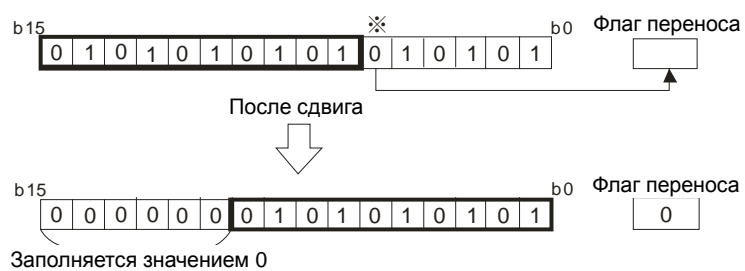
**Пример:**

Если состояние входа X0.0 = ON, значения b0~b15 в D0 сдвигаются на 6 бит вправо, а значение b5 передается в SM602. После сдвига значения b10~b15 сбрасываются на 0.

Ниже приведен пример сдвига значений бит вправо при сканировании.

- ① b5~b0 → Перенос (значение b5 передается в SM602)
- ② b15~b6 → b9~b0
- ③ 0 → b15~b10





**Дополнительные замечания:**

Если значение **n** меньше 0 или значение **n** больше 16, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200В.

API	Код инструкции			Операнд							Функция					
1111		SFL	P	D, n							Сдвиг значений бит в 16-битном регистре на n бит влево					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●			●	●		●			○	○				
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



**D** : Регистр, в котором происходит сдвиг  
**n** : Количество бит

**Описание:**

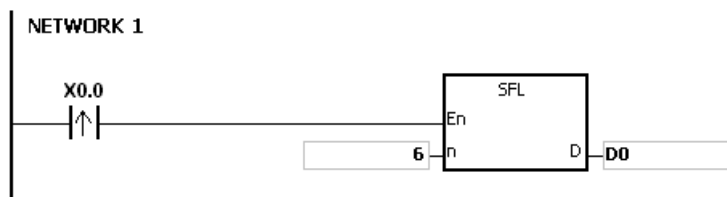
1. Значения бит в **D** сдвигаются на **n** бит влево. Место, освободившееся (b0~bn-1) после сдвига, заполняется значением 0, а значение b16-n передается в SM602.
2. Операнд **n** может иметь значение в диапазоне от 1 до 16.
3. Как правило, используется импульсная инструкция SFLP.

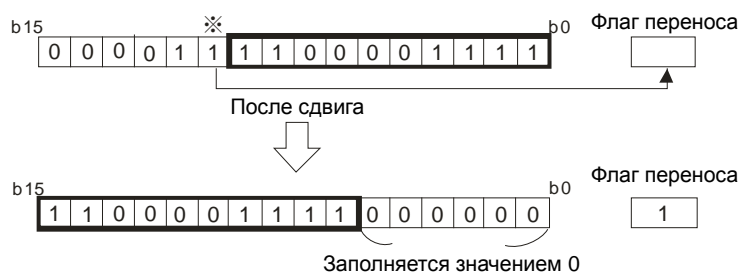
**Пример:**

Если состояние входа X0.0 = ON, значения b0~b15 в D0 сдвигаются на 6 бит влево, а значение b10 передается в SM602. После сдвига значения b0~b5 сбрасываются на 0.

Ниже приведен пример сдвига значений бит влево при сканировании.

- ① b15~b10 → Перенос (значение b10 передается в SM602)
- ② b9~b0 → b15~b6
- ③ 0 → b5~b0





**Дополнительные замечания:**

Если значение **n** меньше 0 или значение **n** больше 16, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

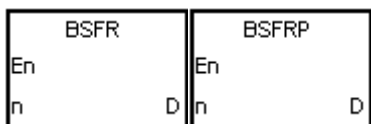
API	Код инструкции			Операнд						Функция					
1112		BSFR	P	D, n						Сдвиг состояний n-битных регистров на один бит вправо					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D		●	●	●				●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●												
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



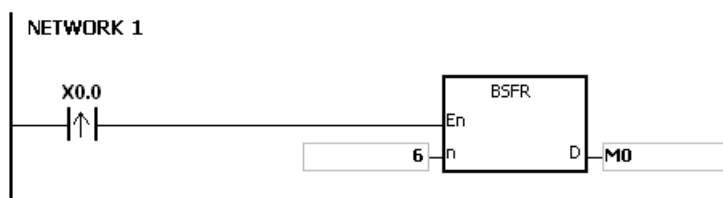
**D** : Начальный регистр, в котором происходит сдвиг  
**n** : Длина данных

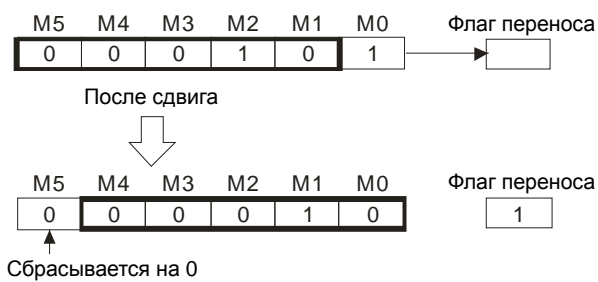
**Описание:**

1. Состояния n-битных регистров, начинающихся с операнда **D**, сдвигаются на 1 бит вправо. Состояние **D+n-1** сбрасывается на 0, а состояние **D** передается во флаг переноса SM602.
2. Как правило, используется импульсная инструкция BSFRP.
3. Операнд **n** может иметь значение в диапазоне от 1 до 1024.

**Пример:**

Если состояние входа X0.0 = ON, состояния M0~M5 сдвигаются на один бит вправо, состояние M5 сбрасывается на 0, а состояние M0 передается во флаг переноса SM602.





**Дополнительные замечания:**

1. Если значение  $D+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 1024, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

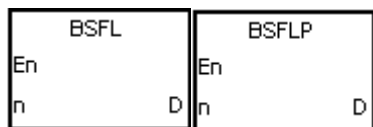
API	Код инструкции			Операнд							Функция					
1113		BSFL	P	D, n							Сдвиг состояний n-битных регистров на один бит влево					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●	●	●				●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D	●												
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



**D** : Начальный регистр, в котором происходит сдвиг

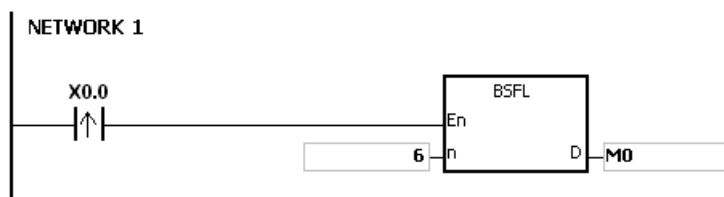
**n** : Длина данных

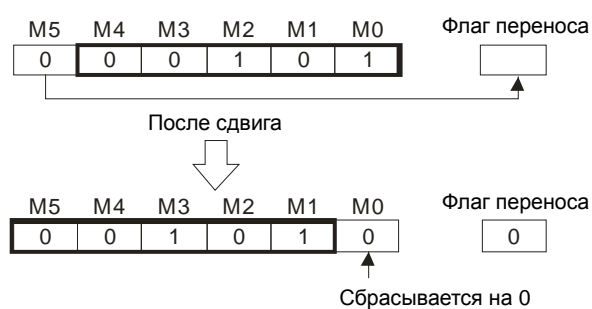
Описание:

1. Состояния n-битных регистров, начинающихся с операнда **D**, сдвигаются на 1 бит влево. Состояние **D** сбрасывается на 0, а состояние **D+n-1** передается во флаг переноса SM602.
2. Как правило, используется импульсная инструкция BSFLP.
3. Операнд **n** может иметь значение в диапазоне от 1 до 1024.

Пример:

Если состояние входа X0.0 = ON, состояния M0~M5 сдвигаются на один бит влево, состояние M0 сбрасывается на 0, а состояние M5 передается во флаг переноса SM602.





**Дополнительные замечания:**

1. Если значение  $D+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 1024, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.



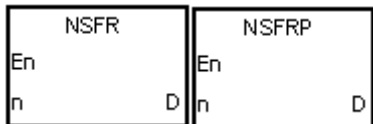
API	Код инструкции			Операнд							Функция					
1114		NSFR	P	D, n							Сдвиг n регистров вправо					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D		●			●	●		●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



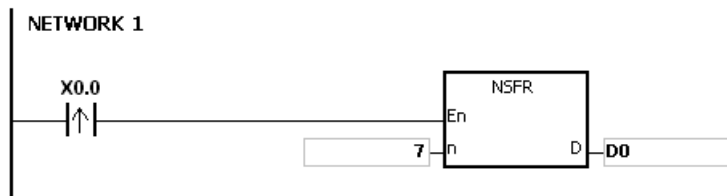
**D** : Начальный регистр, в котором происходит сдвиг  
**n** : Длина данных

**Описание:**

1. Данные в n регистрах, начинающихся с операнда D, сдвигаются вправо, а данные в D+n-1 сбрасываются на 0.
2. Как правило, используется импульсная инструкция NSFRP.
3. Операнд n может иметь значение в диапазоне от 1 до 512.

**Пример:**

Если состояние входа X0.0 = ON, данные в D1~D6 сдвигаются вправо, а данные в D6 сбрасываются на 0.



D6	D5	D4	D3	D2	D1	D0
30	2235	9578	754	28	423	11

После сдвига



D6	D5	D4	D3	D2	D1	D0
0	30	2235	9578	754	28	423

Сбрасывается на 0

**Дополнительные замечания:**

1. Если значение  $D+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 512, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.

API	Код инструкции			Операнд							Функция					
1115		NSFL	P	D, n							Сдвиг n регистров влево					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●			●	●		●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



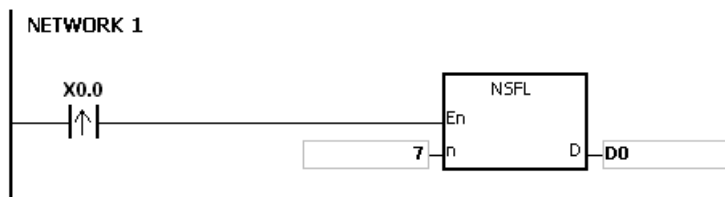
**D** : Начальный регистр, в котором происходит сдвиг  
**n** : Длина данных

**Описание:**

1. Данные в n регистрах, начинающихся с операнда D, сдвигаются влево, а данные в D сбрасываются на 0.
2. Как правило, используется импульсная инструкция NSFLP.
3. Операнд n может иметь значение в диапазоне от 1 до 512.

**Пример:**

Если состояние входа X0.0 = ON, данные в D0~D5 сдвигаются влево, а данные в D0 сбрасываются на 0.



D6	D5	D4	D3	D2	D1	D0
30	2235	9578	754	28	423	11

После сдвига



D6	D5	D4	D3	D2	D1	D0
2235	9578	754	28	423	11	0

Сбрасывается на 0

**Дополнительные замечания:**

1. Если значение  $D+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 512, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.

## 6.13 Инструкции по обработке данных

### 6.13.1 Описание инструкций по обработке данных

API	Код инструкции			Операнд							Функция					
1200	D	SER	P	$S_1, S_2, D, n$							Поиск данных					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●							
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●								
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символьное обозначение:

SER		DSER	
En		En	
S1	D	S1	D
S2		S2	
N		N	
SERP		DSERP	
En		En	
S1	D	S1	D
S2		S2	
N		N	

$S_1$  : Начальный сравниваемый регистр

$S_2$  : Сравниваемые данные

D : Начальный регистр, в котором хранится результат сравнения

n : Длина данных

Описание:

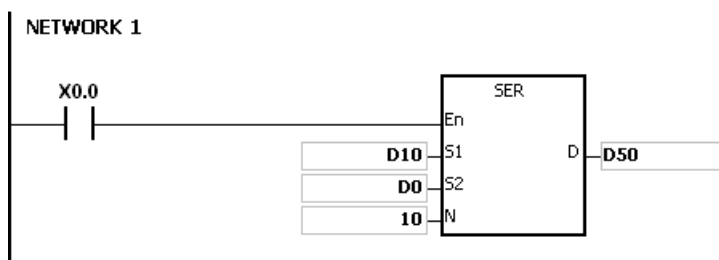
1. Знаковые десятичные значения n в регистрах, начинающихся с регистра, заданного операндом  $S_1$ , сравниваются со знаковыми десятичными значениями в регистре, заданном операндом  $S_2$ , и результат сравнения хранится в регистрах D~D+4.

Регистр	Описание
D	Количество равных значений
D+1	Номер данных первого равного значения
D+2	Номер данных последнего равного значения
D+3	Номер данных минимального равного значения
D+4	Номер данных максимального равного значения

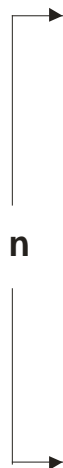
2. Операнд **n** может иметь значение в диапазоне от 1 до 256.
3. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.

**Пример:**

1. Если состояние входа X0.0 = ON, значения D10~D19 сравниваются со значением D0 и результаты сравнения сохраняются в D50~D54. Если равных значений нет, значения D50~D52 равны 0.
2. Номер данных минимального значения хранится в D53, а номер данных максимального значения хранится в D54. Если минимальных и максимальных значений больше одного, сохраняется больший номер данных.



$S_1$	Значение	Сравниваемые данные  $S_2$  $D0=100$	Номер данных	Результат	D	Значение	Описание
D10	88		0		D50	4	Количество равных значений
D11	100		1	Равно	D51	1	Номер данных первого равного значения
D12	110		2		D52	8	Номер данных последнего равного значения
D13	150		3		D53	7	Номер данных минимального равного значения
D14	100		4	Равно	D54	9	Номер данных максимального равного значения
D15	300		5				
D16	100		6	Равно			
D17	5		7	Минимум			
D18	100		8	Равно			
D19	500	9	Максимум				



6

**Дополнительные замечания:**

1. Если  $S_1+n-1$  или  $D+4$  больше допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение операнда  $n$ , используемого в 16-битной инструкции, меньше 1 или больше 256, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
3. Если значение операнда  $n$ , используемого в 32-битной инструкции, меньше 1 или больше 128, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
4. Если операнд  $D$ , используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [5] типа WORD/INT.
5. Если операнд  $D$ , используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [5] типа DWORD/DINT.

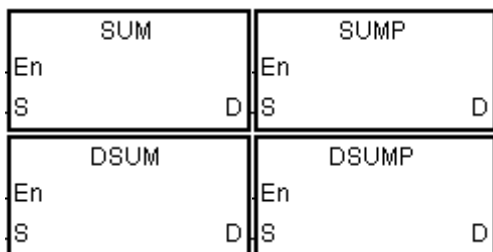
API	Код инструкции			Операнд							Функция						
1201	D	SUM	P	S, D							Количество бит с состоянием ON						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



**S** : Исходный регистр

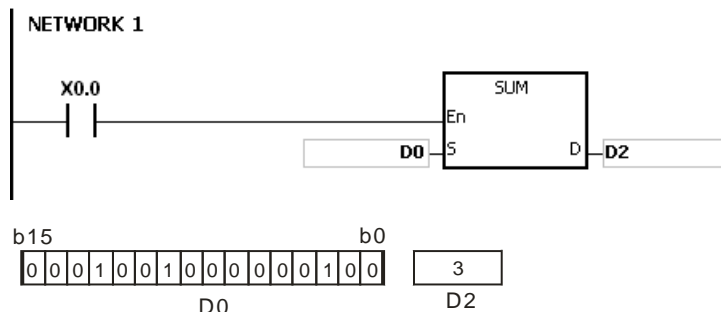
**D** : Управляемый регистр

**Описание:**

1. Количество бит со значением 1 в **S** хранится в **D**.
2. Если значения битов в исходном регистре, заданном операндом S, равны 0, включается флаг нуля SM600.
3. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.

**Пример:**

Если состояние входа X0.0 = ON, количество бит со значением 1 в D0 сохраняется в D2.



**Дополнительные замечания:**

Если значение регистра вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.



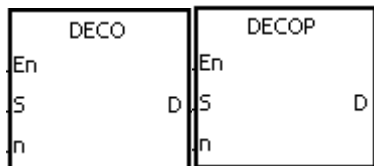
API	Код инструкции			Операнд								Функция				
1202		DECO	P	S, D, n								Дешифратор				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●	●	●	●	●		●	●	○	○	○				
D		●	●	●	●	●		●				○				
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●	●			●	●							
D	●	●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S : Исходный регистр
- D : Регистр, в котором хранятся дешифрованные значения
- n : Количество дешифрованных значений

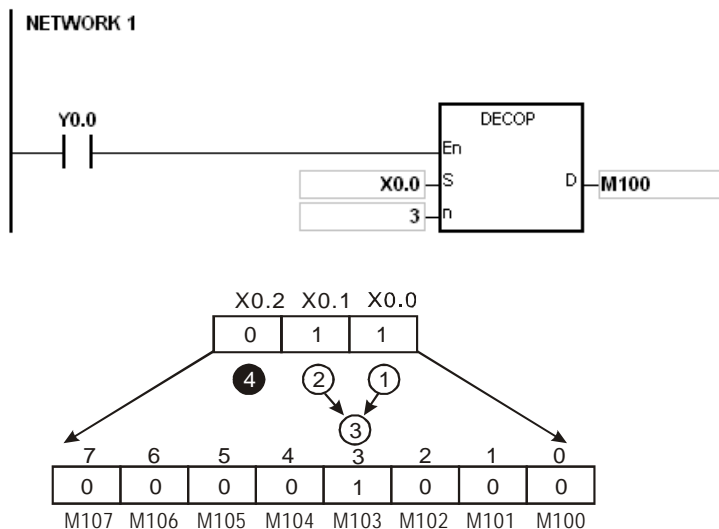
**Описание:**

1. Значения n младших битов в исходном регистре, заданном операндом S, дешифруются как значения 2<sup>n</sup> младших битов в D.
2. Значения n последовательных бит в исходном регистре, заданном операндом S, дешифруются как значения 2<sup>n</sup> младших битов в D.
3. Если исходный регистр, заданный операндом S, является таймером или счетчиком, регистр будет отображаться как регистр типа word.
4. Если D является битовым регистром, операнд n может иметь значения в диапазоне от 1 до 8. Если значение операнда n равно 8, значения 8 бит дешифруются как значения 256 бит. (Обратите внимание, что регистры, в которых хранятся дешифрованные значения, нельзя использовать повторно).
5. Если D является регистром типа word, операнд n может иметь значения в диапазоне от 1 до 4. Если значение операнда n равно 4, значения 4 бит дешифруются как значения 16 бит.
6. Как правило, используется импульсная инструкция DECOP.

**Пример 1:**

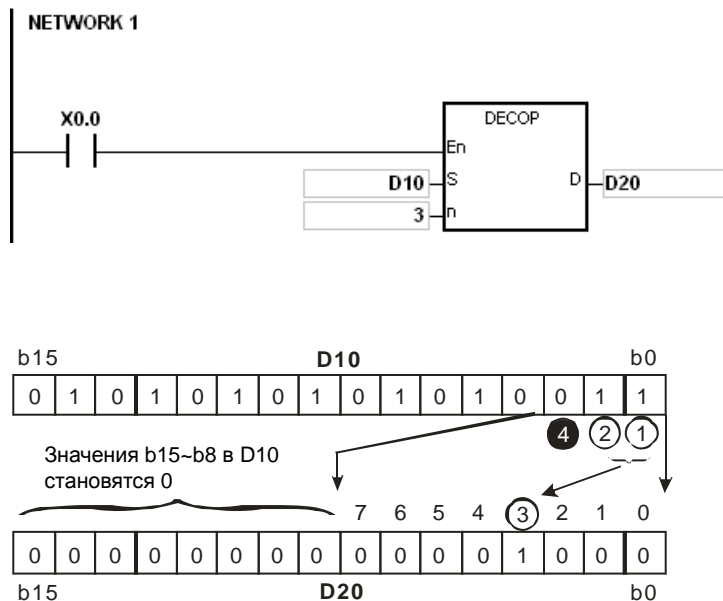
1. Если состояние входа Y0.0 меняется с OFF на ON, инструкция DECO дешифрует значения 3 бит в X0.0~X0.2 как значения 8 бит в M100~M107.

- После сложения 3 бит в X0.0–X0.2 получается значение 3. Третьему биту в M10–M1007, который является битом в M103, присваивается значение 1.
- После выполнения инструкции DECO и состояние входа Y0.0 меняется на OFF, значения 8 бит в M100–M107 не изменяются.



**Пример 2:**

- Если состояние входа X0.0 меняется с OFF на ON, инструкция DECO дешифрует значения b2–b0 в D10 как значения b7–b0 в D20, а значения b15–b8 в D10 становятся 0.
- Значения 3 младших битов в D10 дешифруются как значения 8 младших битов в D20. Значения 8 старших битов равны 0.
- После выполнения инструкции DECO состояние входа X0.0 меняется на OFF, а данные в D20 остаются без изменений.



**Дополнительные замечания:**

1. Предположим, что операнд **D** – это битовый регистр. Если значение **n** меньше 1 или значение **n** больше 8, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
2. Предположим, что операнд **D** – это регистр типа word. Если значение **n** меньше 1 или значение **n** больше 4, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Предположим, что операнд **S** – это битовый регистр. Если значение **S+n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
4. Предположим, что операнд **D** – это битовый регистр. Если  $D+(2^n)-1$  больше допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

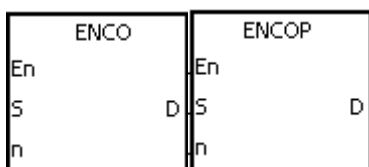
API	Код инструкции			Операнд							Функция						
1203		ENCO	P	S, D, n							Шифратор						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●	●	●	●	●		●	●	○	○	○				
D		●			●	●		●			○	○				
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●	●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S** : Исходный регистр
- D** : Регистр, в котором хранятся зашифрованные значения
- n** : Количество зашифрованных значений

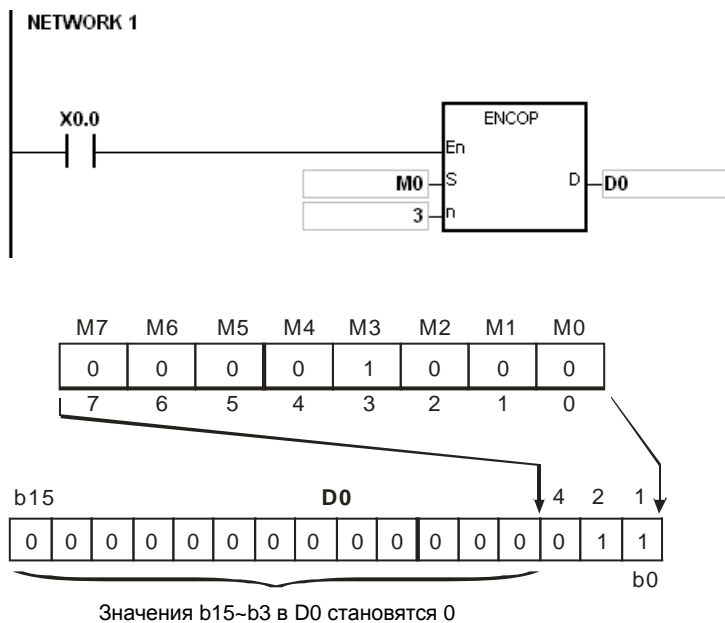
**Описание:**

1. Если операнд **S** является регистром типа word, значения  $2^n$  младших битов в исходном регистре, заданном операндом **S**, зашифровываются как значения **n** младших битов в **D**.
2. Если операнд **S** является битовым регистром, обрабатывается старший бит со значением  $S+(n-1)$  из  $2^n$  младших битов и результат сохраняется в **D**.
3. Если исходный регистр, заданный операндом **S**, является таймером или счетчиком, регистр будет отображаться как регистр типа word.
4. Если операнд **S** является битовым регистром, операнд **n** может иметь значения в диапазоне от 1 до 8. Если значение операнда **n** равно 8, значения 256 бит шифруются как значения 8 бит.
5. Если операнд **S** является регистром типа word, операнд **n** может иметь значения в диапазоне от 1 до 4. Если значение операнда **n** равно 4, значения 16 бит шифруются как значения 4 бит.
6. Как правило, используется импульсная инструкция ENCOP.

**Пример 1:**

1. Если состояние входа X0.0 меняется с OFF на ON, инструкция ENCO зашифровывает значения 8 бит в M0~M7 как значения 3 младших битов в D0, а значения b15~b3 в D0 становятся 0.

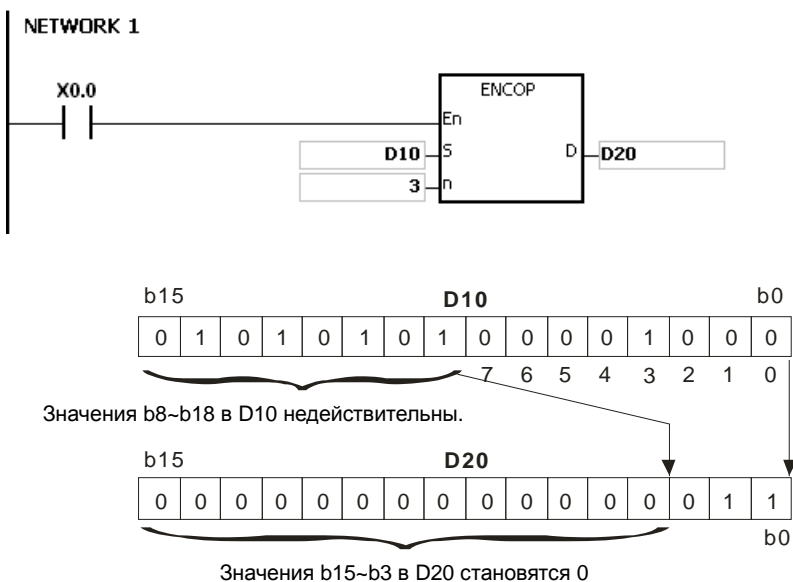
- После выполнения инструкции ENCO состояние входа X0.0 меняется на OFF, а данные в D остаются без изменений.



**Пример 2:**

- Если состояние входа X0.0 меняется с OFF на ON, инструкция ENCO шифрует значения b0~b7 в D10 как значения b2~b0 в D20, а значения b15~b3 в D20 становятся 0 (значения b8~b18 в D10 недействительны).
- После выполнения инструкции ENCO состояние входа X0.0 меняется на OFF, а данные в D остаются без изменений.

6



**Дополнительные замечания:**

- При отсутствии бита со значением 1 в исходном регистре, заданном операндом S, инструкция не выполняется, SM0 = ON, а у SR0 код ошибки 16#2003.

2. Предположим, что операнд **S** – это битовый регистр. Если значение **n** меньше 1 или значение **n** больше 8, инструкция не выполняется, **SM0 = ON** и у **SR0** код ошибки 16#200B.
3. Предположим, что операнд **S** – это регистр типа word. Если значение **n** меньше 1 или значение **n** больше 4, инструкция не выполняется, **SM0 = ON** и у **SR0** код ошибки 16#200B.
4. Предположим, что операнд **S** – это битовый регистр. Если **S+(2^n)-1** больше допустимого диапазона регистра, инструкция не выполняется, **SM0 = ON** и у **SR0** код ошибки 16#2003.
5. Предположим, что операнд **D** – это битовый регистр. Если значение **D+n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, **SM0 = ON** и у **SR0** код ошибки 16#2003.

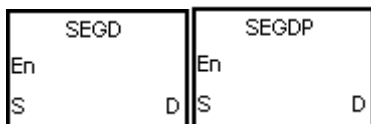
API	Код инструкции			Операнд								Функция				
1204		SEGD	P	S, D								Дешифратор для семисегментного индикатора				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



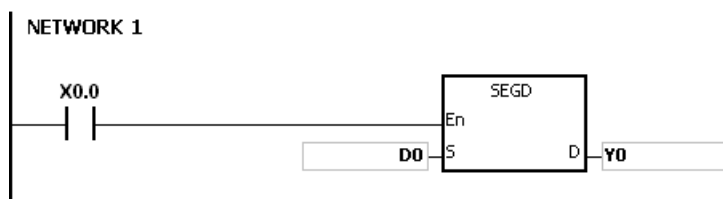
- S** : Исходный регистр
- D** : Регистр, в котором хранятся данные семисегментного индикатора

**Описание:**


Значения 4 младших битов (b0~b3) в исходном регистре, заданном операндом **S**, дешифруются как данные семисегментного индикатора, хранимые в **D**.

**Пример:**

Если состояние входа X0.0 = ON, значения b0~b3 в D0 дешифруются как данные семисегментного индикатора, хранимые в Y0.0~Y0.15. Если данные в исходном регистре превышают 4 бит, дешифруются значения 4 младших битов.



В таблице ниже приводится соотношение между данными семисегментного индикатора и комбинацией битов исходных данных.

Значение Hex	Комбинация бит	Назначение сегментов	Состояние каждого сегмента							Данные на дисплее
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ON	ON	ON	ON	ON	ON	OFF	0
1	0001		OFF	ON	ON	OFF	OFF	OFF	OFF	1
2	0010		ON	ON	OFF	ON	ON	OFF	ON	2
3	0011		ON	ON	ON	ON	OFF	OFF	ON	3
4	0100		OFF	ON	ON	OFF	OFF	ON	ON	4
5	0101		ON	OFF	ON	ON	OFF	ON	ON	5
6	0110		ON	OFF	ON	ON	ON	ON	ON	6
7	0111		ON	ON	ON	OFF	OFF	ON	OFF	7
8	1000		ON	ON	ON	ON	ON	ON	ON	8
9	1001		ON	ON	ON	ON	OFF	ON	ON	9
A	1010		ON	ON	ON	OFF	ON	ON	ON	A
B	1011		OFF	OFF	ON	ON	ON	ON	ON	B
C	1100		ON	OFF	OFF	ON	ON	ON	OFF	C
D	1101		OFF	ON	ON	ON	ON	OFF	ON	D
E	1110		ON	OFF	OFF	ON	ON	ON	ON	E
F	1111		ON	OFF	OFF	OFF	ON	ON	ON	F



API	Код инструкции			Операнд								Функция				
1205	D	SORT	P	S, m <sub>1</sub> , m <sub>2</sub> , D, n								Сортировка данных				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S								●								
m <sub>1</sub>								●					○	○		
m <sub>2</sub>								●					○	○		
D								●								
n								●					○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
m <sub>1</sub>		●	●		●	●	●						
m <sub>2</sub>		●	●		●	●	●						
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**

SORT		DSORT	
En		En	
S	D	S	D
m1		m1	
m2		m2	
n		n	

SORTP		DSORTP	
En		En	
S	D	S	D
m1		m1	
m2		m2	
n		n	

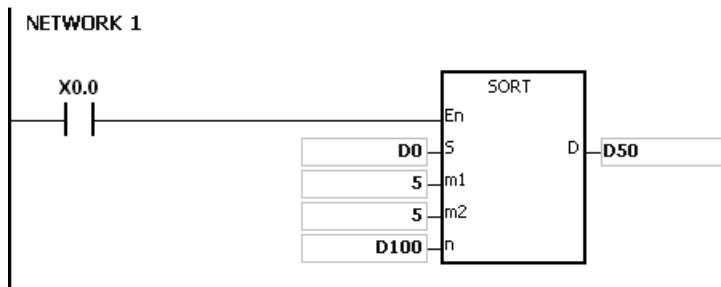
- S** : Начальный регистр, в котором хранятся исходные данные
- m<sub>1</sub>** : Количество строк данных
- m<sub>2</sub>** : Количество столбцов данных
- D** : Начальный регистр, в котором хранятся отсортированные данные
- n** : Контрольное значение для сортировки данных

**Описание:**

1. Отсортированные данные, которые хранятся в регистрах m<sub>1</sub>xm<sub>2</sub>, начиная с регистра, заданного операндом **D**. Если операнды **S** и **D** задают один и тот же регистр, отсортированные данные совпадают с исходными данными в регистре, заданном операндом **S**.
2. Операнд **m<sub>1</sub>** может иметь значение в диапазоне от 1 до 32. Операнд **m<sub>2</sub>** может иметь значение в диапазоне от 1 до 6. Операнд **n** может иметь значение в диапазоне от 1 до **m<sub>2</sub>**.
3. Если SM604 = OFF, данные сортируются в порядке возрастания. Если SM604 = ON, данные сортируются в порядке убывания.
4. Вместо повторной сортировки рекомендуется использовать инструкцию импульсного типа.
5. 32-битный счетчик может использоваться только в 32-битных инструкциях, а в регистре E – не может.

**Пример:**

1. Предположим, что SM604 = OFF. Если состояние входа X0.0 меняется с OFF на ON, данные сортируются в порядке возрастания.



2. Ниже приводятся данные, которые будут сортироваться.

		← столбцы данных m <sub>2</sub> →				
		Столбец				
Столбец		1	2	3	4	5
Строка		Номер студента	Китайский	Английский	Математика	Физика
↑ Строки данных m <sub>1</sub> ↓	1	(D0) 1	(D5) 90	(D10) 75	(D15) 66	(D20) 79
	2	(D1) 2	(D6) 55	(D11) 65	(D16) 54	(D21) 63
	3	(D2) 3	(D7) 80	(D12) 98	(D17) 89	(D22) 90
	4	(D3) 4	(D8) 70	(D13) 60	(D18) 99	(D23) 50
	5	(D4) 5	(D9) 95	(D14) 79	(D19) 75	(D24) 69

3. Если значение в D100 равно 3, данные сортируются следующим образом.

		← столбцы данных $m_2$ →				
		Столбец				
Столбец	1	2	3	4	5	
	Номер студента	Китайский	Английский	Математика	Физика	
↑ Строки данных $m_1$ ↓	1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
	2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
	3	(D52) 1	(D57) 90	(D62) 75	(D67) 66	(D72) 79
	4	(D53) 5	(D58) 95	(D63) 79	(D68) 75	(D73) 69
	5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

4. Если значение в D100 равно 5, данные сортируются следующим образом.

		← столбцы данных $m_2$ →				
		Столбец				
Столбец	1	2	3	4	5	
	Номер студента	Китайский	Английский	Математика	Физика	
↑ Строки данных $m_1$ ↓	1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
	2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
	3	(D52) 5	(D57) 95	(D62) 79	(D67) 75	(D72) 69
	4	(D53) 1	(D58) 90	(D63) 75	(D68) 66	(D73) 79
	5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

**Дополнительные замечания:**

1. Если значение регистра вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение  $m_1$ ,  $m_2$  или  $n$  вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

API	Код инструкции			Операнд							Функция						
1206		ZRST	P	D <sub>1</sub> , D <sub>2</sub>							Сброс зоны						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D <sub>1</sub>		●	●	●	●	●	●	●		○	○	○				
D <sub>2</sub>		●	●	●	●	●	●	●		○	○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D <sub>1</sub>	●	●			●	●							
D <sub>2</sub>	●	●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

ZRST	ZRSTP
En	En
D1	D1
D2	D2

**D<sub>1</sub>** : Начальный регистр, для которого выполняется сброс

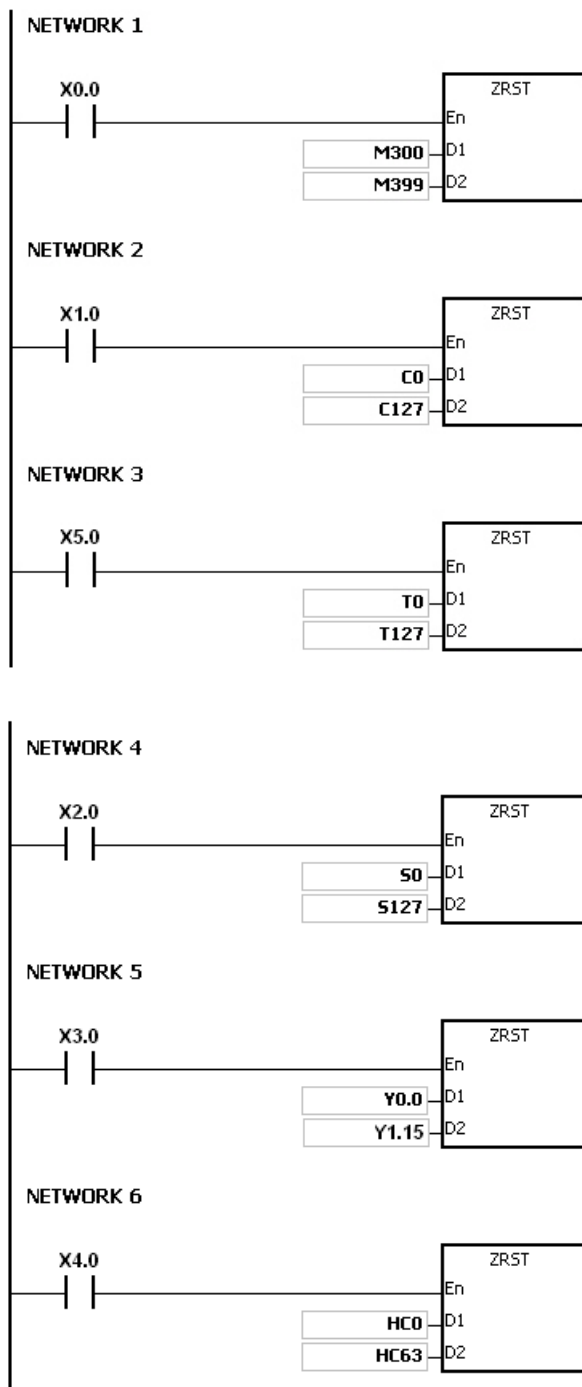
**D<sub>2</sub>** : Конечный регистр, для которого выполняется сброс

**Описание:**

1. После выполнения инструкции значения в **D<sub>1</sub>~D<sub>2</sub>** сбрасываются. Для выполнения данной инструкции тип регистра должен быть таким же.
2. Если номер регистра **D<sub>1</sub>** больше номера регистра **D<sub>2</sub>**, сбрасывается только **D<sub>2</sub>**.
3. 32-битный счетчик может использоваться в инструкции ZRST.

**Пример:**

1. Если состояние входа X0.0 = ON, состояние вспомогательных реле M300~M399 меняется на OFF.
2. Если состояние входа X1.0 = ON, выполняется сброс 16-битных счетчиков C0~C127 (значения C0~C127 сбрасываются на 0, а состояние контакта и катушки меняется на OFF).
3. Если состояние входа X2.0 = ON, состояние шаговых реле S0~S127 меняется на OFF.
4. Если состояние входа X3.0 = ON, состояние выходных реле Y0.0~Y1.15 меняется на OFF.
5. Если состояние входа X4.0 = ON, выполняется сброс 32-битных счетчиков HC0~HC63 (значения HC0~HC63 сбрасываются на 0, а состояние контакта и катушки меняется на OFF).
6. Если состояние входа X5.0 = ON, выполняется сброс таймеров T0~T127 (значения T0~T127 сбрасываются на 0, а состояние контакта и катушки меняется на OFF).



**Дополнительные замечания:**

1. Если  $D_1$  и  $D_2$  представлены разными регистрами, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2007.
2. Если  $D_1$  и  $D_2$  содержат данные разных форматов, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2007.

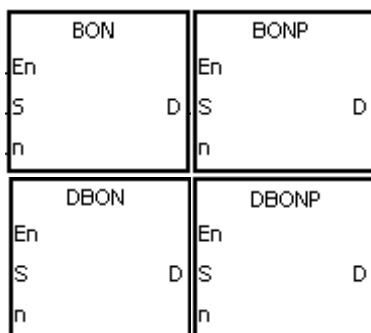
API	Код инструкции			Операнд							Функция					
1207	D	BON	P	S, D, n							Проверка состояния битов					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●	●	●				●		○						
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D	●												
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



**S** : Исходный регистр

**D** : Регистр, в котором хранится результат проверки

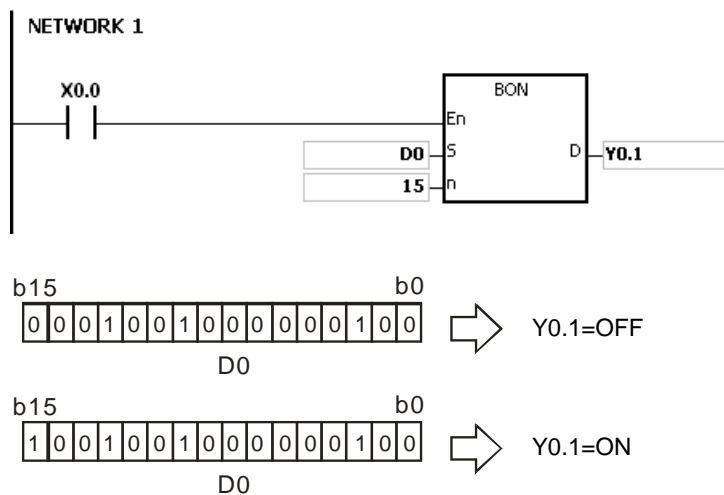
**n** : Бит, состояние которого оценивается

**Описание:**

1. Выполняется проверка состояния **n**-го бита в **S**, а результат проверки сохраняется в **D**.
2. Значение операнда **n**, используемого в 16-битной инструкции, должно находиться в диапазоне от 0 до 15, а значение операнда **n**, используемого в 32-битной инструкции, должно находиться в диапазоне от 0 до 31.
3. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре **E** – не может.

**Пример:**

1. Если состояние входа  $X0.0 = ON$ , состояние выхода  $Y0.1 = ON$ , если значение 15-го бита в  $D0$  равно 1. Если состояние входа  $X0.0 = ON$ , состояние выхода  $Y0.1 = OFF$ , если значение 15-го бита в  $D0$  равно 0.
2. Если состояние входа  $X0.0 = OFF$ , состояние выхода  $Y0.1$  остается таким же, как и было до изменения состояния входа  $X0.0$  на  $OFF$ .



**Дополнительные замечания:**

Если значение **n** вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.



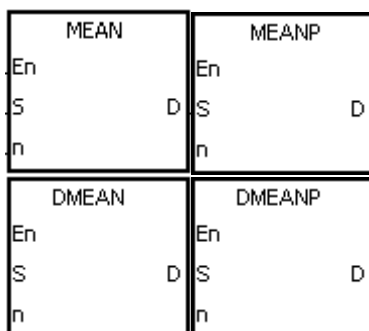
API	Код инструкции			Операнд							Функция					
1208	D	MEAN	P	S, D, n							Среднее арифметическое					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S	●	●			●	●	●	●	●							
D		●			●	●	●	●			○	○				
n	●	●			●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
D		●	●		●	●	●						
n		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



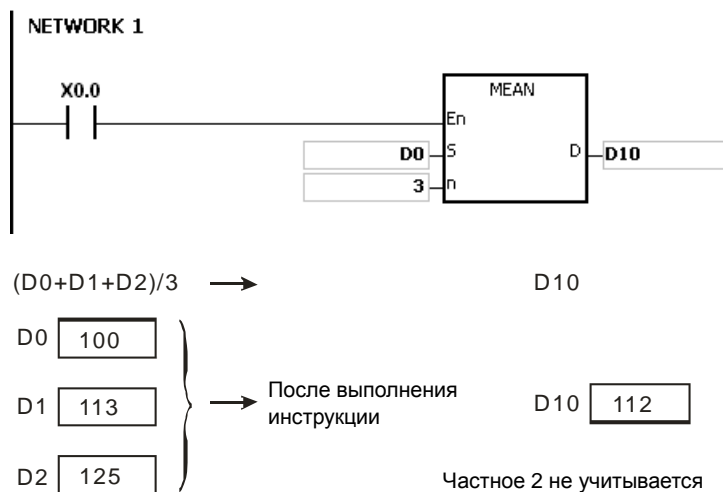
- S** : Начальный регистр
- D** : Регистр, в котором хранится среднее арифметическое
- n** : Количество регистров

**Описание:**

1. После сложения значений в регистре **n**, начиная с регистра, заданного операндом **S**, среднее арифметическое от суммы сохраняется в **D**.
2. Если при вычислении появляется остаток, он не учитывается.
3. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 256.
4. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре **E** – не может.

**Пример:**

Если состояние входа X0.0 = ON, выполняется сложение значений трех регистров, начиная с D0. После сложения значений их сумма делится на 3. Частное хранится в D10, а остаток не учитывается.



**Дополнительные замечания:**

1. Если значение операнда **n**, используемого в 16-битной инструкции, меньше 1 или больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
2. Если значение операнда **n**, используемого в 32-битной инструкции, меньше 1 или больше 128, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Если значение **S+n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

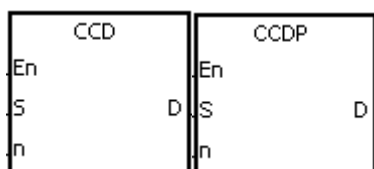
API	Код инструкции			Операнд							Функция					
1209		CCD	P	S, D, n							Проверка суммы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



- S** : Начальный регистр
- D** : Регистр, в котором хранится сумма
- n** : Количество блоков данных

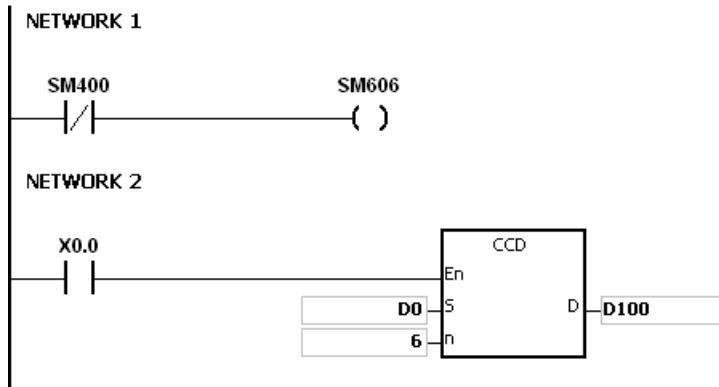
6

Описание:

1. При обмене данными проверка суммы выполняется для сравнения контрольных сумм одних и тех же данных в разных случаях или при разном представлении данных для проверки их целостности.
2. 16-битный режим преобразования: Если SM606 = OFF, инструкция работает в режиме 16-битного преобразования. Выполняется сложение блоков данных **n** в регистрах, начинающихся с регистра, заданного операндом **S** (8 бит в группе). Сумма хранится в регистре, заданном операндом **D**, а значения битов четности хранятся в **D+1**.
3. 8-битный режим преобразования: Если SM606 = ON, инструкция работает в режиме 8-битного преобразования. Выполняется сложение блоков данных **n** в регистрах, начинающихся с регистра, заданного операндом **S** (8 бит в группе, но действительны только восемь младших битов). Сумма хранится в регистре, заданном операндом **D**, а значения битов четности хранятся в **D+1**.
4. Операнд **n** может иметь значение в диапазоне от 1 до 256.

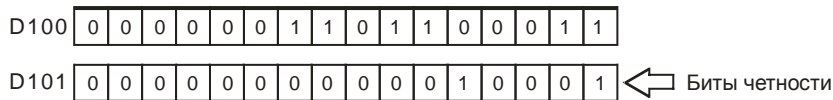
**Пример 1:**

1. Если SM606 = OFF, инструкция работает в режиме 16-битного преобразования.
2. Если состояние входа X0.0 = ON, в D0~D2 (8 бит в группе), выполняется сложение шести блоков данных. Сумма хранится в D100, а значения битов четности хранятся в D101.



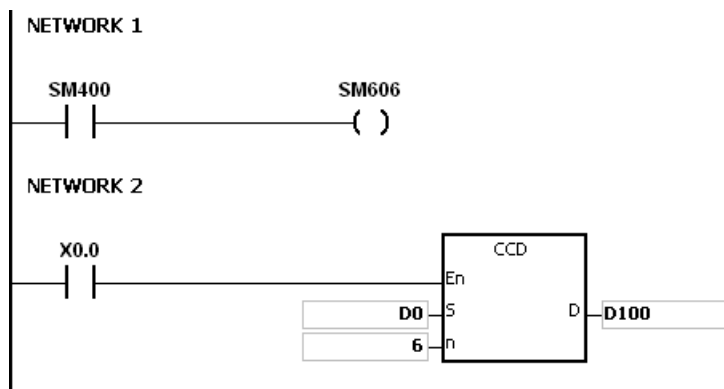
S	Данные
D0 младший	100 = 0 1 1 0 0 1 0 0
D0 старший	111 = 0 1 1 0 1 1 1 ① ←
D1 младший	120 = 0 1 1 1 1 0 0 0
D1 старший	202 = 1 1 0 0 1 0 1 0
D2 младший	123 = 0 1 1 1 1 0 1 ① ←
D2 старший	211 = 1 1 0 1 0 0 1 ① ←
D 100	867
D 101	0 0 0 1 0 0 0 ① ←

Sum  
 Если количество единиц нечетное, бит четности равен 1.  
 Если количество единиц четное, бит четности равен 0.



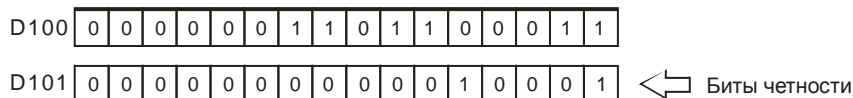
**Пример 2:**

1. Если SM606 = ON, инструкция работает в режиме 8-битного преобразования.
2. Если состояние входа X0.0 = ON, в D0~D5 (8 бит в группе), выполняется сложение шести блоков данных. Сумма хранится в D100, а значения битов четности хранятся в D101.



S	Данные
D0 младший	100 = 0 1 1 0 0 1 0 0
D1 младший	111 = 0 1 1 0 1 1 1 1
D2 младший	120 = 0 1 1 1 1 0 0 0
D3 младший	202 = 1 1 0 0 1 0 1 0
D4 младший	123 = 0 1 1 1 1 0 1 1
D5 младший	211 = 1 1 0 1 0 0 1 1
D100	867
D101	0 0 0 1 0 0 0 1

↑↑ Если количество единиц нечетное, бит четности равен 1.  
 Если количество единиц четное, бит четности равен 0.



6

**Дополнительные замечания:**

1. Предположим, что SM606 = ON. Если значение S+n-1 вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Предположим, что SM606 = OFF. Если значение S+n/2-1 вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если значение n меньше 1 или значение n больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
4. Если операнд D объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.

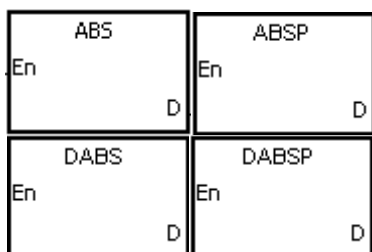
API	Код инструкции			Операнд							Функция						
1210	D	ABS	P	D							Абсолютное значение						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



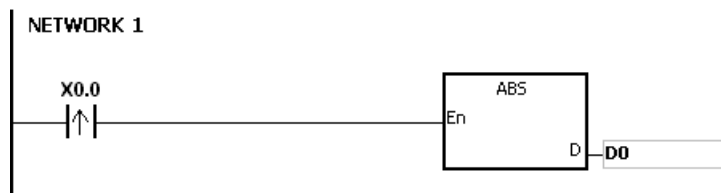
**D** : Регистр, используемый для получения абсолютного значения

**Описание:**

1. После выполнения инструкции ABS, получается абсолютное значение от значения в регистре, заданном операндом **D**.
2. Как правило, используется импульсная инструкция ABSP.
3. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре **E** – не может.

**Пример:**

Предположим, что значение в **D0** до выполнения инструкции равно 1234. Если состояние входа **X0.0** меняется с OFF на ON, получается абсолютное значение от 1234 в **D0**. То есть после выполнения инструкции значение в **D0** становится 1234.



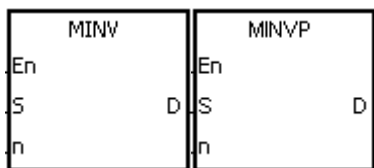
API	Код инструкции			Операнд							Функция					
1211		MINV	P	S, D, n							Инверсия битов матрицы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
D		●			●	●		●								
n	●	●			●	●		●	●		○		○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



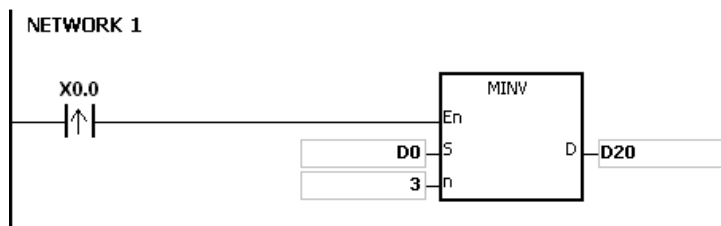
- S** : источник матрицы
- D** : результат операции
- n** : длина массива

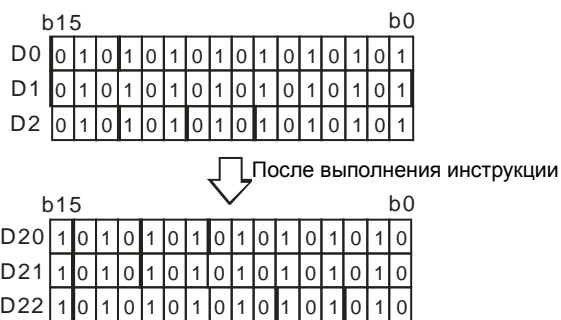
**Описание:**

1. Выполняется инверсия битов в регистрах **n**, начиная с регистра, заданного операндом **S**, а результат инверсии сохраняется в **D**.
2. Операнд **n** может иметь значение в диапазоне от 1 до 256.

**Пример:**

Если состояние входа X0.0 = ON, выполняется инверсия битов в трех 16-битных регистрах D0–D2, а результаты инверсии сохраняются в 16-битных регистрах D20–D22.





**Дополнительные замечания:**

1. Если **S+n-1** или **D+n-1** больше допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **n** меньше 1 или значение **n** больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.



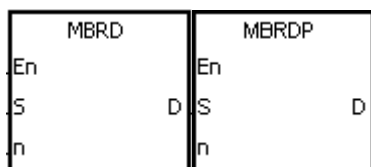
API	Код инструкции			Операнд								Функция				
1212		MBRD	P	S, n, D								Чтение битов в матрице				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●							
n	●	●			●	●		●	●		○		○	○		
D		●			●	●		●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



- S : Источник матрицы
- n : Длина массива
- D : Указатель

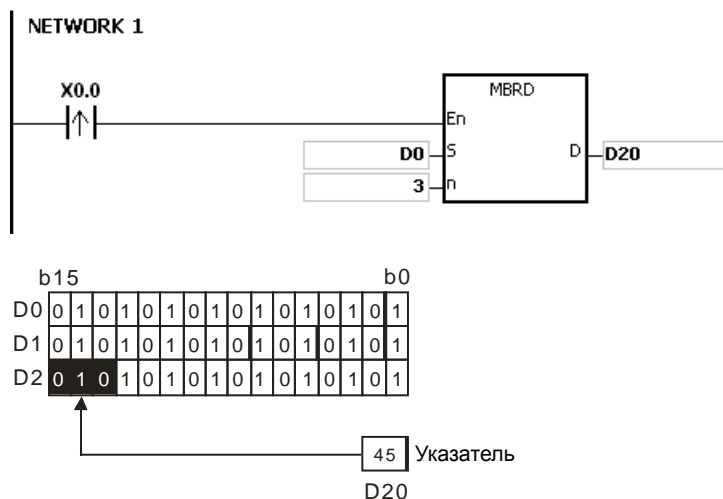
## 6 Описание:

- После выполнения инструкции проверяется состояние SM613. Если состояние SM613 = ON, значение указателя **D** сбрасывается на 0. Значение бита, заданного значением указателя **D**, считывается в SM614. После считывания значения битов, выполняется проверка состояния SM612. Если состояние SM612 = ON, значение указателя **D** увеличивается на единицу.
- Если считывается значение последнего бита, SM608 = ON, а номер бита записывается в указатель **D**.
- Операнд **n** может иметь значение в диапазоне от 1 до 256.
- Значение указателя задается пользователем. Значения задаются в диапазоне от 0 до 16n-1, который соответствует диапазону от b0 до b16n-1. Если значение указателя выходит за пределы допустимого диапазона, SM611 присваивается значение 1, а инструкция не выполняется.

## Пример:

- Предположим, что состояние SM613 = OFF и SM612 = ON, тогда состояние входа X0.0 меняется с OFF на ON.
- Предположим, что текущее значение в D20 равно 45. Если состояние входа X0.0 меняется с OFF на ON три раза, выдаются следующие результаты выполнения.

- ❶ Значение в D20 равно 46, SM614 = OFF, SM608 = OFF.
- ❷ Значение в D20 равно 47, SM614 = ON, SM608 = OFF.
- ❸ Значение в D20 равно 47, SM614 = OFF, SM608 = ON.



**Дополнительные замечания:**

1. Если значение **S+n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **n** меньше 1 или значение **n** больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Флаги:
  - SM608: Сравнение матриц завершено. После сравнения последних битов состояние SM608 = ON.
  - SM611: Флаг ошибки указателя матрицы. Когда значение указателя выходит за пределы допустимого диапазона сравнения, состояние SM611 = ON.
  - SM612: Флаг увеличения указателя матрицы. Текущее значение указателя увеличивается на единицу.
  - SM613: Флаг сброса указателя матрицы. Текущее значение указателя сбрасывается на 0.
  - SM614: Флаг переноса при выдаче/сдвиге/вращении матрицы.

API	Код инструкции			Операнд								Функция				
1213		MBWR	P	S, n, D								Запись битов в матрице				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●		●								
n	●	●			●	●		●	●		○		○	○		
D		●			●	●		●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



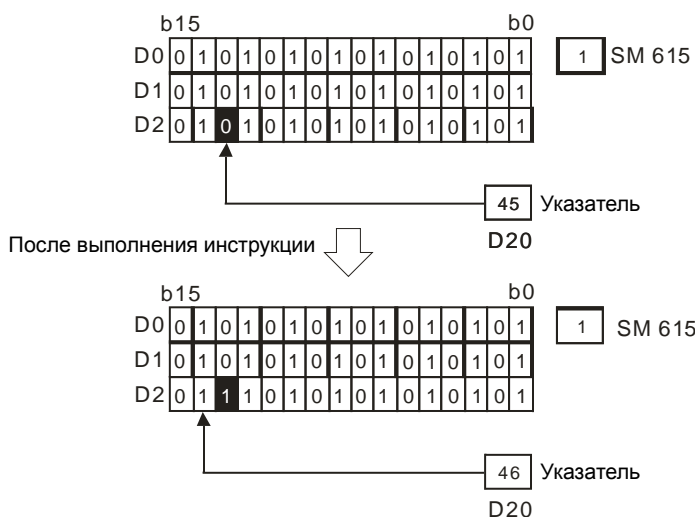
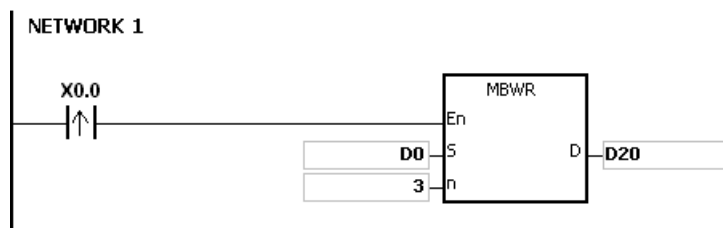
- S** : Источник матрицы
- n** : Длина массива
- D** : Указатель

**Описание:**

1. После выполнения инструкции проверяется состояние SM613. Если состояние SM613 = ON, значение указателя **D** сбрасывается на 0. Состояние SM615 записывается в бит, заданный значением указателя **D**. После записи состояния SM615 в бит выполняется проверка состояния SM612. Если состояние SM612 = ON, значение в указателе **D** увеличивается на единицу.
2. Если состояние SM615 записывается в последний бит, SM608 = ON, а номер бита записывается в указатель **D**. Если значение указателя **D** выходит за пределы допустимого диапазона сравнения, состояние SM611 = ON.
3. Операнд **n** может иметь значение в диапазоне от 1 до 256.
4. Значение указателя задается пользователем. Значения задаются в диапазоне от 0 до 16n-1, который соответствует диапазону от b0 до b16n-1. Если значение указателя выходит за пределы допустимого диапазона, SM611 присваивается значение 1, а инструкция не выполняется.

**Пример:**

1. Предположим, что состояние SM613 = OFF и SM612 = ON, тогда состояние входа X0.0 меняется с OFF на ON.
2. Предположим, что текущее значение в D20 равно 45. Если состояние входа X0.0 меняется с OFF на ON один раз, выдаются следующие результаты выполнения. Если значение в D20 равно 45, SM615 = OFF, SM608 = OFF.



**Дополнительные замечания:**

1. Если значение **S+n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение **n** меньше 1 или значение **n** больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Флаги:
  - SM608: Сравнение матриц завершено. После сравнения последних битов состояние SM608 = ON.
  - SM611: Флаг ошибки указателя матрицы. Когда значение указателя выходит за пределы допустимого диапазона сравнения, состояние SM611 = ON.
  - SM612: Флаг увеличения указателя матрицы. Текущее значение указателя увеличивается на единицу.
  - SM613: Флаг сброса указателя матрицы. Текущее значение указателя сбрасывается на 0.
  - SM615: Флаг заимствования при сдвиге/выдаче матрицы.

API	Код инструкции			Операнд							Функция						
1214		MBC	P	S, n, D							Подсчет битов со значением 0 или 1						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S		●			●	●		●								
n	●	●			●	●		●	●		○		○	○		
D		●			●	●		●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



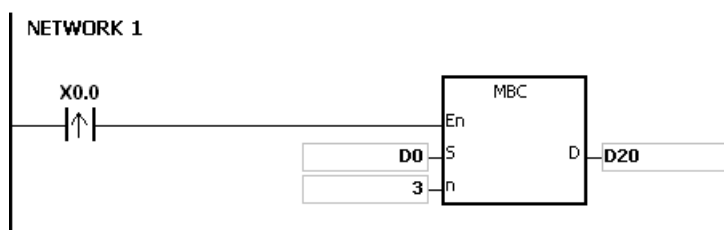
- S** : источник матрицы
- n** : длина массива
- D** : результат операции

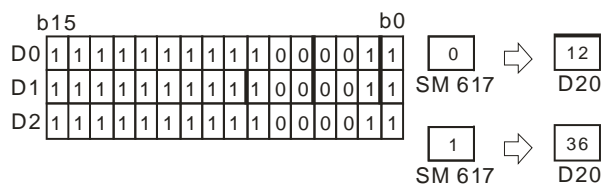
**Описание:**

1. Данная инструкция служит для подсчета бит со значением 1 или 0 в регистрах **n**, начиная с регистра, заданного операндом **S**. Результат операции сохраняется в **D**.
2. Если состояние SM617 = ON, выполняется подсчет битов со значением 1. Если состояние SM617 = OFF, выполняется подсчет битов со значением 0. Если результат операции 0, состояние SM618 = ON.
3. Операнд **n** может иметь значение в диапазоне от 1 до 256.

**Пример:**

Предположим, что SM617 = ON. Если состояние входа X0.0 = ON, выполняется подсчет битов со значением 1, а результат операции сохраняется в D20. Предположим, что SM617 = OFF. Если состояние входа X0.0 = ON, выполняется подсчет битов со значением 0, а результат операции сохраняется в D20.





**Дополнительные замечания:**

1. Если значение  $S+n-1$  вне пределов допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 256, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.
3. Флаги:
  - SM617: Подсчет битов со значением 0 или 1.
  - SM618: Если результат подсчета в матрице равен 0, состояние  $SM618 = ON$ .

API	Код инструкции			Операнд							Функция						
1215		DIS	P	S, n, D							Разделение 16-битных данных						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●		○	○				
n	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

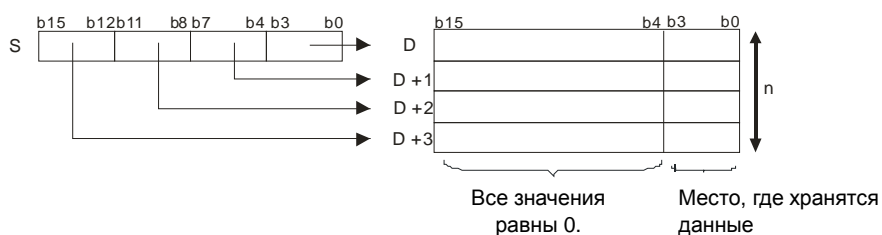
Символьное обозначение:



- S** : Источник данных
- n** : Количество регистров
- D** : Результат операции

### 6 Описание:

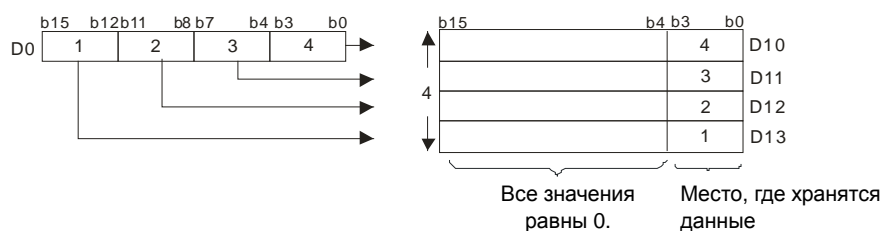
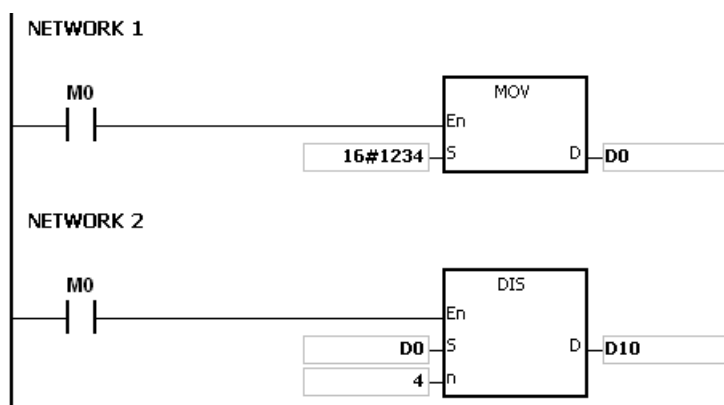
- 16-битное значение в регистре, заданном операндом **S** делится на четыре группы (по четыре бита в группе), и эти группы сохраняются в четырех младших битах в каждом регистре (регистры в диапазоне от **D** до **D+(n-1)**).



- Операнд **n** может иметь значение в диапазоне от 1 до 4.

### Пример:

Предположим, что значение в D0 равно 16#1234. Когда M0 включен, инструкция DIS начинает исполняться. Значение в D0 делится на 4 группы (по четыре бита в группе), и эти группы сохраняются в четырех младших битах в каждом регистре (регистры в диапазоне от D10 до D13.).



**Дополнительные замечания:**

1. Если  $D \sim D+(n-1)$  больше допустимого диапазона регистра, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 4, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#200B.



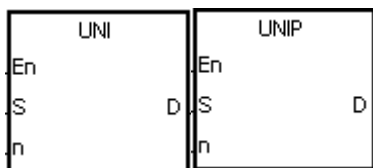
API	Код инструкции			Операнд							Функция				
1216		UNI	P	<b>S, n, D</b>							Объединение 16-битных данных				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S</b>	●	●			●	●		●	●							
<b>n</b>	●	●			●	●		●	●		○	○	○	○		
<b>D</b>		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●			●	●							
<b>n</b>		●			●	●							
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

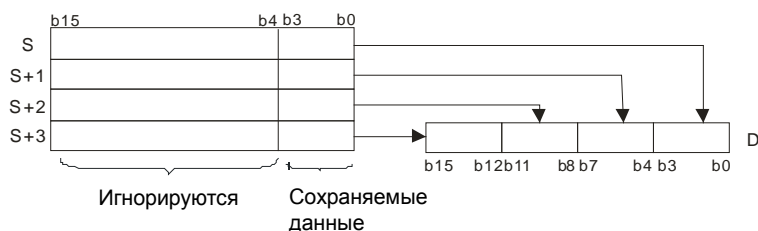
Символьное обозначение:



- S** : Источник данных
- n** : Длина данных
- D** : Результат операции

Описание:

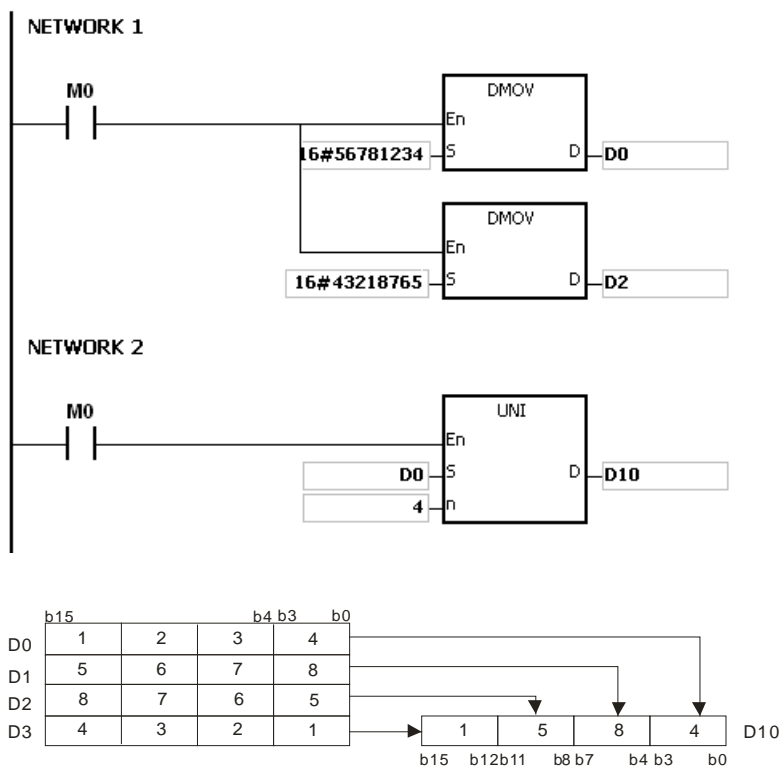
- 16-битные значения в регистрах, заданных **S~S+(n-1)** делятся на группы (по четыре бита в группе), и каждая группа, состоящая из **b0~b3**, сохраняется в регистре, заданном операндом **D** (**b0~b15**).



- Операнд **n** может иметь значение в диапазоне от 1 до 4.

Пример:

Предположим, значения в **D0~D3** равны **16#1234**, **16#5678**, **16#8765** и **16#4321**, соответственно. Когда **M0** включен, инструкция **UNI** начинает исполняться. Значения в **D0~D3** делятся на группы (по четыре бита в группе), и каждая группа, состоящая из **b0~b3**, сохраняется в **D10** (**b0~b15**).



**Дополнительные замечания:**

1. Если  $S \sim S+(n-1)$  больше допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение  $n$  меньше 1 или значение  $n$  больше 4, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.

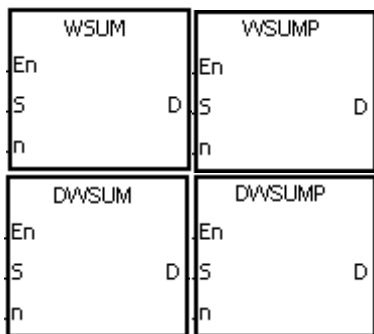
API	Код инструкции			Операнд							Функция					
1217	D	WSUM	P	S, n, D							Получение суммы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●							
n	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●						
n		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

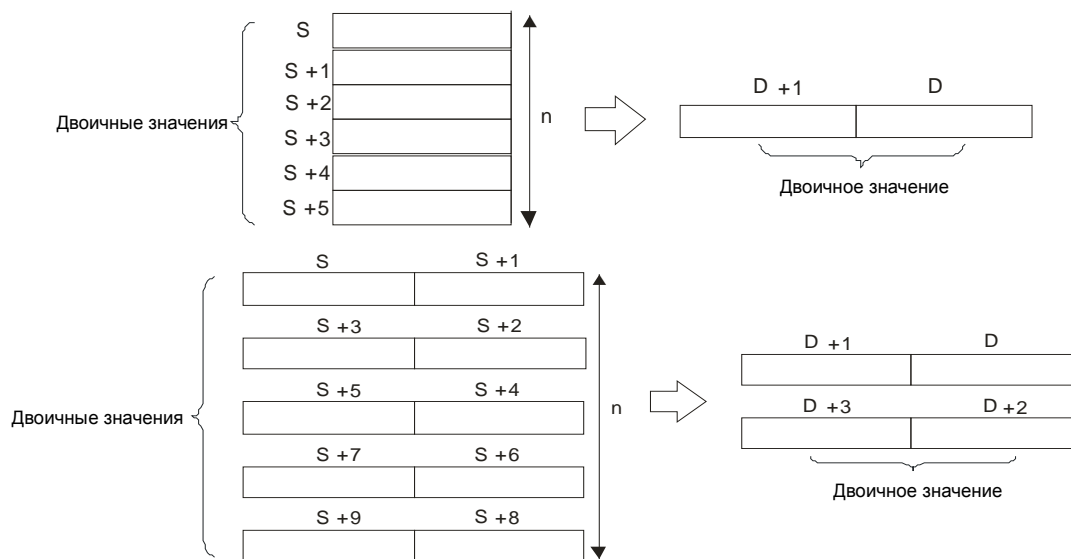
**Символьное обозначение:**



- S** : Источник данных
- n** : Длина данных
- D** : Результат операции

**Описание:**

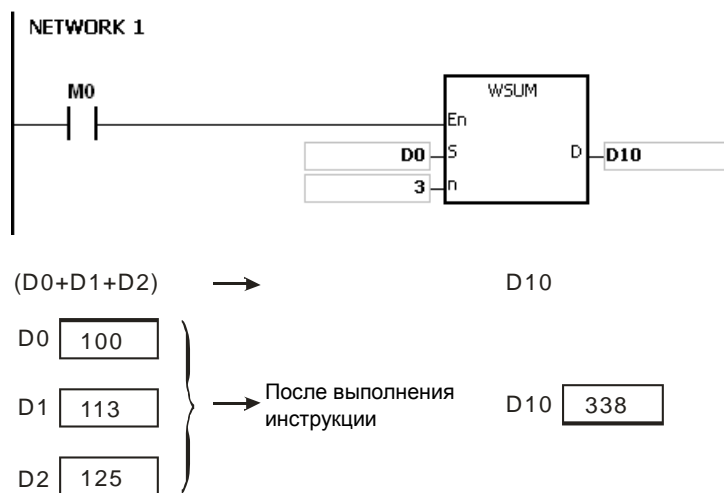
1. Выполняется сложение знаковых десятичных значений в **S~S+n-1** и сумма сохраняется в регистре, заданном операндом **D**.



2. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 256.
3. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.

**Пример:**

После выполнения инструкции WSUM выполняется сложение значений в D0~D2, а их сумма сохраняется в D10.



**Дополнительные замечания:**

1. Если значение операнда **n**, используемое в 16-битной инструкции, меньше 1 или больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
2. Если значение операнда **n**, используемое в 32-битной инструкции, меньше 1 или больше 128, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
3. Если значение **S+n-1** или **D** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
4. Если операнд **D**, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет DWORD или МАССИВ [2] типа WORD.
5. Если операнд **D**, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа DWORD.

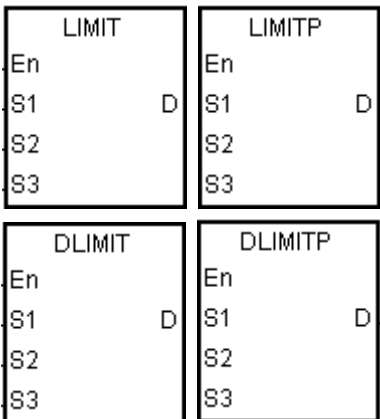
API	Код инструкции			Операнд							Функция					
1221	D	LIMIT	P	S <sub>1</sub> , S <sub>2</sub> , S <sub>3</sub> , D							Ограничение значений					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●	●	●	●		○	○	○	○		
S <sub>3</sub>	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●	●		●	●	●						
S <sub>2</sub>		●	●		●	●	●						
S <sub>3</sub>		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



- S<sub>1</sub> : Минимальное выходное значение
- S<sub>2</sub> : Максимальное выходное значение
- S<sub>3</sub> : Входное значение
- D : Выходное значение

**Описание:**

1. Входное значение в операнде S<sub>3</sub> сравнивается с минимальным выходным значением в операнде S<sub>1</sub> и максимальным выходным значением в операнде S<sub>2</sub>, а результат сравнения сохраняется в операнде D.

Если минимальное выходное значение в операнде S<sub>1</sub> больше входного значения в операнде S<sub>3</sub>, выходное значение, сохраненное в операнде D, равно минимальному выходному значению в операнде S<sub>1</sub>.

Если максимальное выходное значение в операнде S<sub>2</sub> меньше входного значения в операнде S<sub>3</sub>, выходное значение, сохраненное в операнде D, равно максимальному выходному значению в операнде S<sub>2</sub>.

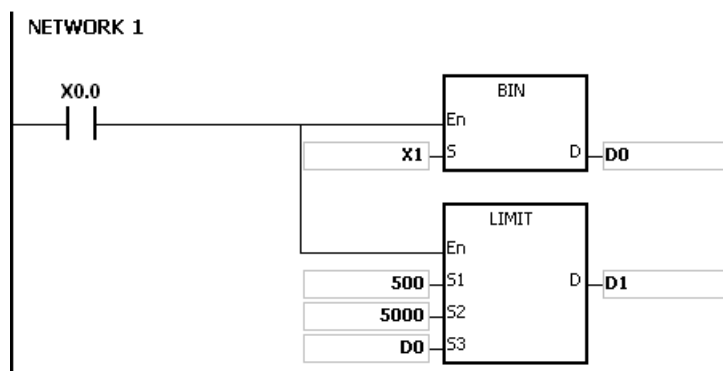
Если входное значение в операнде S<sub>3</sub> находится в пределах диапазона между минимальным выходным значением в операнде S<sub>1</sub> и максимальным выходным значением в операнде S<sub>2</sub>, выходное значение, сохраненное в операнде D, равно входному значению в операнде S<sub>3</sub>.

Если минимальное выходное значение в операнде **S<sub>1</sub>** больше максимального выходного значения в операнде **S<sub>2</sub>**, инструкция не выполняется.

- 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.

**Пример:**

- Если состояние входа X0.0 = ON, состояние X1 преобразуется в двоичное значение, а результат преобразования сохраняется в D0. Кроме этого, значение, сохраненное в D0, сравнивается с 500 и 5000, а результат сравнения сохраняется в D1.



Минимальное выходное значение	Максимальное выходное значение	Выходное значение в D0	Функция	Выходное значение в D1
500	5000	499	$D0 < 500$	500
		5001	$D0 > 5000$	5000
		600	$500 \leq D0 \leq 5000$	600

**Дополнительные замечания:**

Если минимальное выходное значение в операнде **S<sub>1</sub>** больше максимального выходного значения в операнде **S<sub>2</sub>**, инструкция не выполняется, SM0 = ON, а у SR0 код ошибки 16#2003.

API	Код инструкции			Операнд							Функция					
1222	D	BAND	P	$S_1, S_2, S_3, D$							Регулирование зоны нечувствительности					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
$S_3$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
$S_3$		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символьное обозначение:

BAND		BANDP	
En		En	
$S_1$	D	$S_1$	D
$S_2$		$S_2$	
$S_3$		$S_3$	

DBAND		DBANDP	
En		En	
$S_1$	D	$S_1$	D
$S_2$		$S_2$	
$S_3$		$S_3$	

- $S_1$  : Минимальное значение зоны нечувствительности
- $S_2$  : Максимальное значение зоны нечувствительности
- $S_3$  : Входное значение
- D : Выходное значение

Описание:

- Минимальное значение зоны нечувствительности в операнде  $S_1$  или максимальное значение зоны нечувствительности в операнде  $S_2$  вычитается из входного значения в операнде  $S_3$ , и разница сохраняется в операнде D.

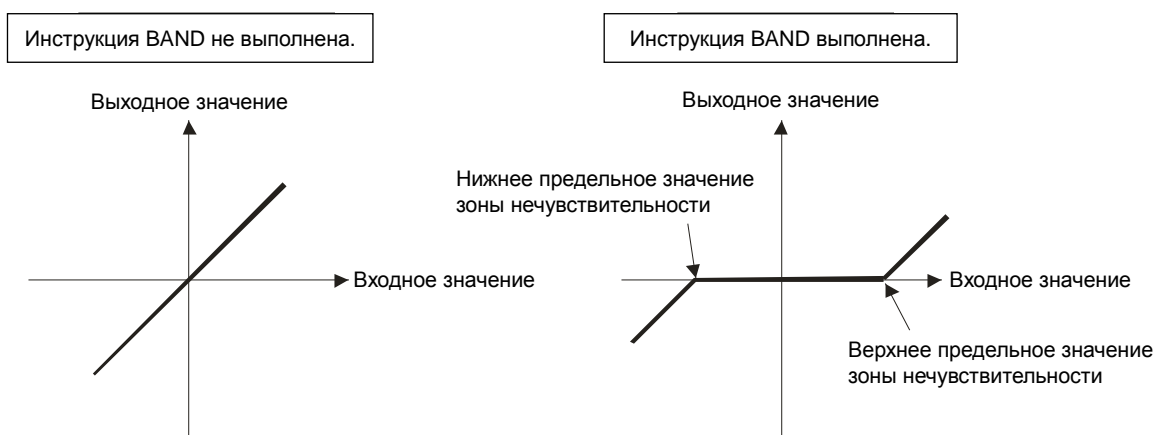
Если минимальное значение зоны нечувствительности в операнде  $S_1$  больше входного значения в операнде  $S_3$ , минимальное значение зоны нечувствительности в операнде  $S_1$  вычитается из входного значения в операнде  $S_3$ , а разница сохраняется в операнде D.

Если максимальное значение зоны нечувствительности в операнде  $S_2$  меньше входного значения зоны нечувствительности в операнде  $S_3$ , максимальное значение зоны нечувствительности в операнде  $S_2$  вычитается из входного значения в операнде  $S_3$ , а разница сохраняется в операнде D.

Если входное значение в операнде  $S_3$  находится в пределах диапазона между минимальным значением зоны нечувствительности в операнде  $S_1$  и максимальным значением зоны нечувствительности в операнде  $S_2$ , выходное значение, сохраняемое в операнде  $D$ , равно 0.

Если минимальное значение зоны нечувствительности в операнде  $S_1$  больше максимального значения зоны нечувствительности в операнде  $S_2$ , инструкция не выполняется.

2. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.
3. Рисунки:



4. Минимальное значение зоны нечувствительности в операнде  $S_1$ , максимальное значение зоны нечувствительности в операнде  $S_2$ , входное значение в операнде  $S_3$  и выходное значение в операнде  $D$  должны находиться в пределах диапазона, указанного ниже.

- После выполнения инструкции BAND минимальное значение зоны нечувствительности в операнде  $S_1$ , максимальное значение зоны нечувствительности в операнде  $S_2$ , входное значение в операнде  $S_3$  и выходное значение в операнде  $D$  находятся в пределах диапазона от -32768 до 32767. Предположим, что минимальное значение зоны нечувствительности в операнде  $S_1$  равно 10, а максимальное значение зоны нечувствительности в операнде  $S_3$  равно -32768. Тогда выходное значение в операнде  $D$  рассчитывается следующим образом.

$$\text{Выходное значение в } D = -32768 - 10 = 16\#8000 - 16\#000A = 16\#7FF6 = 32758$$

- После выполнения инструкции DBAND минимальное значение зоны нечувствительности в операнде  $S_1$ , максимальное значение зоны нечувствительности в операнде  $S_2$ , входное значение в операнде  $S_3$  и выходное значение в операнде  $D$  находятся в пределах диапазона от -2147483648 до 2147483647. Предположим, что минимальное значение зоны нечувствительности в  $(S_1+1, S_1)$  равно 1000, а максимальное значение зоны нечувствительности в  $(S_3+1, S_3)$  равно -2147483648. Тогда выходное значение в  $(D+1, D)$  рассчитывается следующим образом.

$$\text{Выходное значение в } (D+1, D)$$

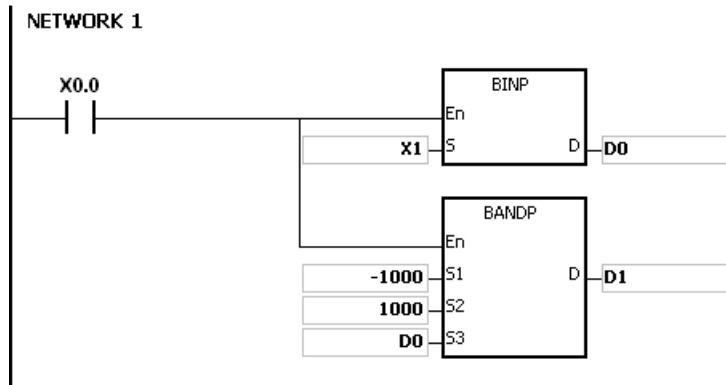
$$= -2147483648 - 1000 = 16\#80000000 - 16\#000003E8 = 16\#7FFFC18$$

$$= 2147482648$$



**Пример 1:**

Если состояние входа X0.0 = ON, значение -1000 или 1000 вычитается из двоично-кодированного десятичного значения в X1, а разница сохраняется в D1.

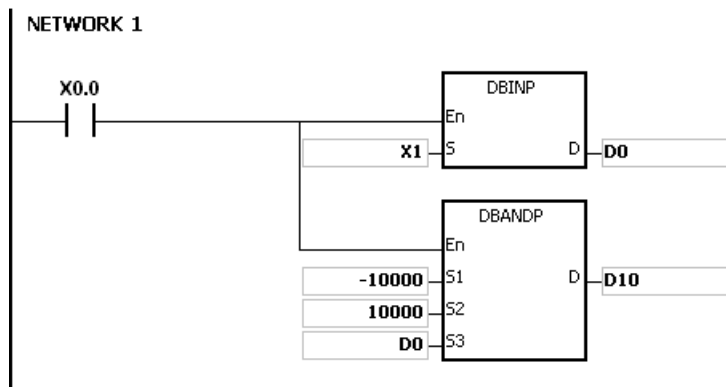


Результаты выполнения:

Минимальное значение зоны нечувствительности	Максимальное значение зоны нечувствительности	Входное значение в D0	Функция	Выходное значение в D1
-1000	1000	-1200	$D0 < -1000 \Rightarrow D1 = D0 - (-1000)$	-200
		1200	$D0 > 1000 \Rightarrow D1 = D0 - 1000$	200
		500	$-1000 \leq D0 \leq 1000 \Rightarrow D0 = 0$	0

**Пример 2:**

Если состояние входа X0.0 = ON, значение -10000 или 10000 вычитается из двоично-кодированного десятичного значения в (X2, X1), а разница сохраняется в (D11, D10).



Результаты выполнения:

Минимальное значение зоны нечувствительности	Максимальное значение зоны нечувствительности	Входное значение в (D1, D0)	Функция	Выходное значение в (D11, D10)
-10000	10000	-12000	$(D1, D0) < -10000$ $\Rightarrow (D11, D10)$ $= (D1, D0) - (-10000)$	-2000
		12000	$(D1, D0) > 10000$ $\Rightarrow (D11, D10)$ $= (D1, D0) - 10000$	2000
		5000	$-10000 \leq (D1, D0) \leq 10000$ $\Rightarrow (D1, D0) = 0$	0

**Дополнительные замечания:**

Если минимальное значение зоны нечувствительности в операнде S1 больше максимального значения зоны нечувствительности в операнде S2, инструкция не выполняется, SM0 = ON, а у SR0 код ошибки 16#2003.

API	Код инструкции			Операнд							Функция						
1223	D	ZONE	P	$S_1, S_2, S_3, D$							Контроль зоны						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●	●	●	●		○	○	○	○		
$S_2$	●	●			●	●	●	●	●		○	○	○	○		
$S_3$	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
$S_3$		●	●		●	●	●						
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**

ZONE		ZONEP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	

DZONE		DZONEP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	

- $S_1$  : Отрицательное отклонение
- $S_2$  : Положительное отклонение
- $S_3$  : Входное значение
- D : Выходное значение

**Описание:**

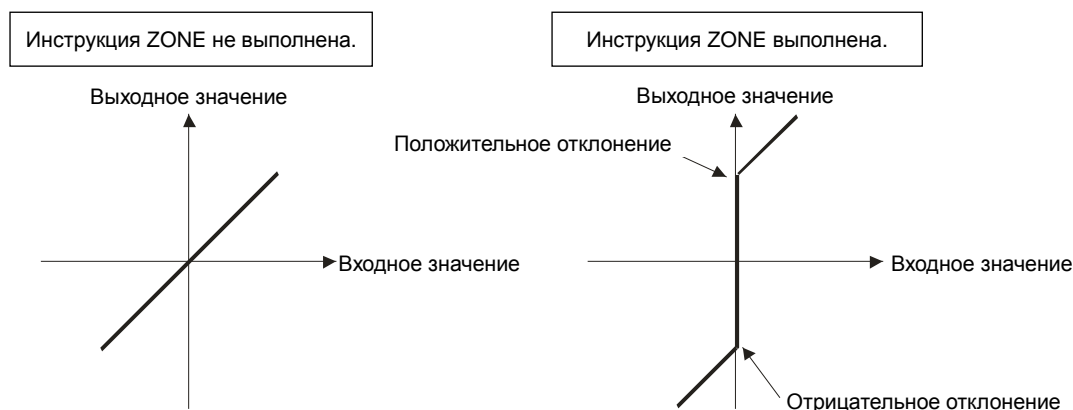
- Отрицательное отклонение в операнде  $S_1$  и положительное отклонение в операнде  $S_2$  добавляются к входному значению в операнде  $S_3$ , а сумма сохраняется в операнде D.

Если входное значение в операнде  $S_3$  меньше 0, отрицательное отклонение в операнде  $S_1$  добавляется к входному значению в операнде  $S_3$ , а сумма сохраняется в операнде D.

Если входное значение в операнде  $S_3$  больше 0, положительное отклонение в операнде  $S_2$  добавляется к входному значению в операнде  $S_3$ , а сумма сохраняется в операнде D.

Если входное значение в операнде  $S_3$  равно 0, выходное значение, сохраненное в операнде D, равно 0.

2. Рисунки:



3. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.

4. Отрицательное отклонение в операнде **S<sub>1</sub>**, положительное отклонение в операнде **S<sub>2</sub>**, входное значение в операнде **S<sub>3</sub>** и выходное значение в операнде **D** должны находиться в указанных ниже пределах.

- После выполнения инструкции ZONE отрицательное отклонение в операнде **S<sub>1</sub>**, положительное отклонение в операнде **S<sub>2</sub>**, входное значение в операнде **S<sub>3</sub>** и выходное значение в операнде **D** находятся в пределах диапазона от -32768 до 32767. Предположим, что отрицательное отклонение в операнде **S<sub>1</sub>** равно -100, а входное значение в операнде **S<sub>3</sub>** равно -32768. Тогда выходное значение в операнде **D** рассчитывается следующим образом.

$$\text{Выходное значение в D} = (-32768) + (-100) = 16\#8000 + 16\#FF9C = 16\#7F9C = 32668$$

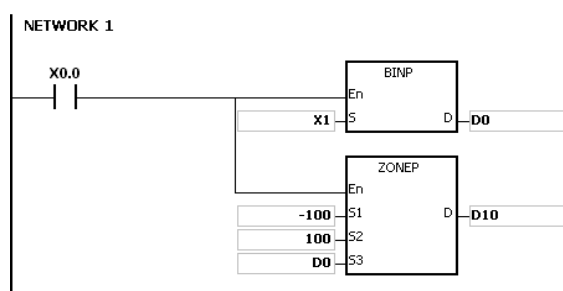
- После выполнения инструкции DZONE отрицательное отклонение в операнде **S<sub>1</sub>**, положительное отклонение в операнде **S<sub>2</sub>**, входное значение в операнде **S<sub>3</sub>** и выходное значение в операнде **D** находятся в пределах диапазона от -2147483648 до 2147483647. Предположим, что отрицательное отклонение в (**S<sub>1</sub>+1, S<sub>1</sub>**) равно -1000, а входное значение в (**S<sub>3</sub>+1, S<sub>3</sub>**) равно -2147483648. Тогда выходное значение в (**D+1, D**) рассчитывается следующим образом.

$$\text{Выходное значение в (D+1, D)}$$

$$= -2147483648 + (-1000) = 16\#80000000 + 16\#FFFFFFC18 = 16\#7FFFFFFC18 = 2147482648$$

**Пример 1:**

Если состояние входа X0.0 = ON, значение -100 или 100 прибавляется к двоично-кодированному десятичному значению в X1, а сумма сохраняется в D10.

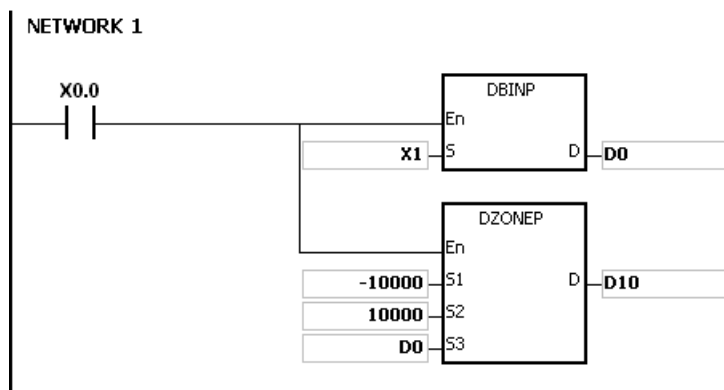


Результаты выполнения:

Отрицательное отклонение	Положительное отклонение	Входное значение в D0	Функция	Выходное значение в D10
-100	100	-10	$D0 < 0 \Rightarrow D10 = (-10) + (-100)$	-110
		0	$D0 = 0 \Rightarrow D10 = 0$	0
		50	$D0 > 0 \Rightarrow D10 = 50 + 100$	150

**Пример 2:**

Если состояние входа X0.0 = ON, значение -10000 или 10000 прибавляется к двоично-кодированному десятичному значению в (X2, X1), а сумма сохраняется в (D11, D10).



6

Отрицательное отклонение	Положительное отклонение	Входное значение в (D1, D0)	Функция	Выходное значение в (D11, D10)
-10000	10000	-10	$(D1, D0) < 0 \Rightarrow (D11, D10) = (-10) + (-10000)$	-10010
		0	$(D1, D0) = 0 \Rightarrow (D11, D10) = 0$	0
		50	$(D1, D0) > 0 \Rightarrow (D11, D10) = 50 + 10000$	10050

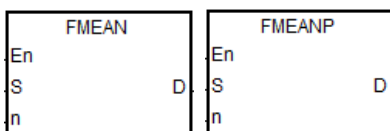
API	Код инструкции			Операнд								Функция					
1224		FMEAN	P	S, D, n								Среднее арифметическое значение с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S							●	●								
n							●	●					○	○		
D							●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
n			●				●						
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



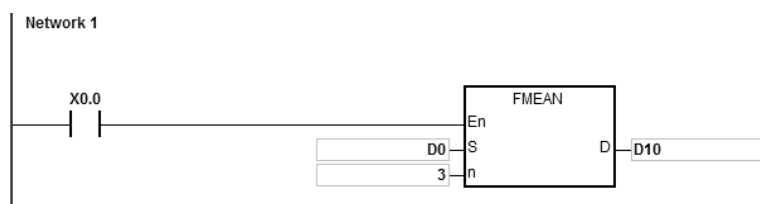
- S** : Начальный регистр
- D** : Регистр, в котором хранится среднее арифметическое
- n** : Количество регистров

**Описание:**

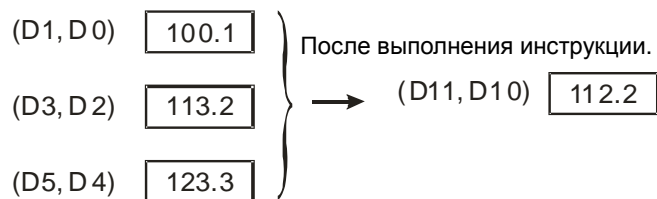
1. После сложения значений одинарной точности с плавающей запятой в регистрах **n**, начиная с регистра, заданного операндом **S**, среднее арифметическое от суммы сохраняется в **D**.
2. Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 256.
3. Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), SM602 (флаг переноса):
  - Если результат операции 0, состояние SM600 = ON. В остальных случаях состояние OFF.
  - Если значение при сложении или абсолютный результат операции меньше значения с плавающей запятой, D=16#FF800000 и включается флаг заимствования SM601.
  - Если значение при сложении или абсолютный результат операции больше значения с плавающей запятой, D=16#7F800000 и включается флаг переноса SM602.

**Пример:**

Если состояние входа X0.0 = ON, прибавьте 3 значения одинарной точности с плавающей запятой к (D1, D0), (D3, D2), (D5, D4) и разделите полученную сумму на 3, затем сохраните результат в (D11, D10).



$$[(D1, D0) + (D3, D2) + (D5, D4)] / 3 \rightarrow (D11, D10)$$



**Дополнительные замечания:**

1. Если используемый операнд **n** меньше 1 или больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
2. Если значение **S+2\*n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON, а у SR0 код ошибки 16#2013.

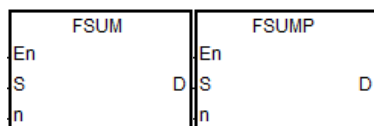
API	Код инструкции			Операнд				Функция				
1225		FSUM	P	<b>S, n, D</b>				Сумма значений с плавающей запятой				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S</b>							●	●								
<b>n</b>							●	●					○	○		
<b>D</b>							●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>									●				
<b>n</b>			●				●						
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

Символьное обозначение:



**S** : Источник данных

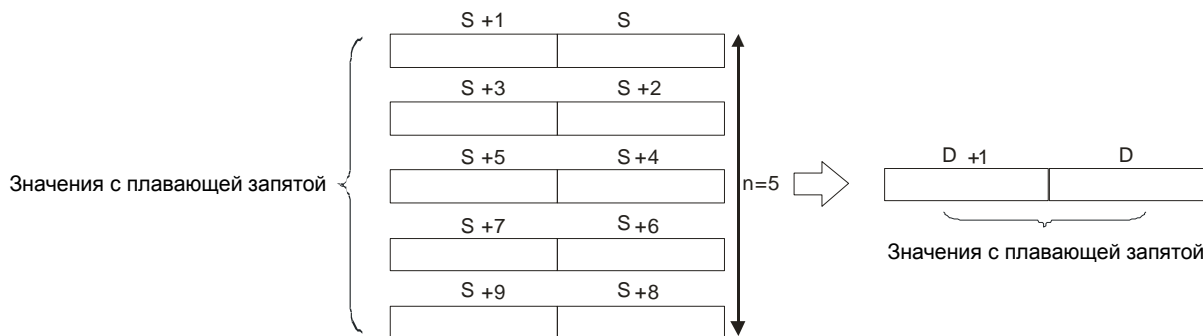
**n** : Длина данных

**D** : Результат операции

Описание:

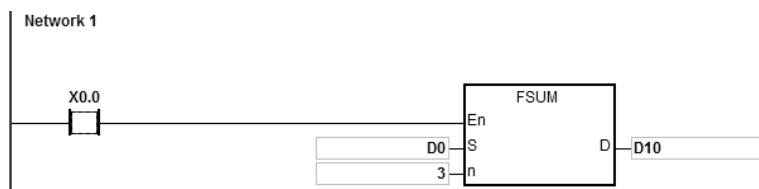
- После сложения значений одинарной точности с плавающей запятой в регистрах **n**, начиная с регистра, заданного операндом **S**, сумма сохраняется в **D**.
- Операнд **n**, используемый в 16-битной инструкции, может иметь значение в диапазоне от 1 до 256.
- Флаги: SM600 (флаг нуля), SM601 (флаг заимствования), SM602 (флаг переноса):
  - Если результат операции 0, состояние SM600 = ON. В остальных случаях состояние OFF.
  - Если значение при сложении или абсолютный результат операции меньше значения с плавающей запятой, D=16#FF800000 и включается флаг заимствования SM601.
  - Если значение при сложении или абсолютный результат операции больше значения с плавающей запятой, D=16#7F800000 и включается флаг переноса SM602.



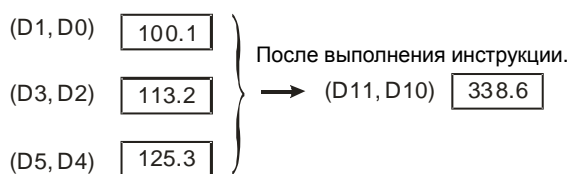


**Пример:**

После выполнения инструкции FSUM выполняется сложение 3 значений одинарной точности с плавающей запятой в (D1, D0), (D3, D2), (D5, D4) и результат сохраняется в (D11, D10).



$$[(D1, D0) + (D3, D2) + (D5, D4)] \rightarrow (D11, D10)$$



**Дополнительные замечания:**

1. Если используемый операнд **n** меньше 1 или больше 256, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
2. Если значение **S+2\*n-1** вне пределов допустимого диапазона регистра, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
3. Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON, а у SR0 код ошибки 16#2013.

API	Код инструкции			Операнд								Функция				
1226		DTM	P	S, D, m, n								Передача и перемещение данных				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●								
D		●			●	●		●								
m								●					○	○		
n								●					○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D		●			●	●							
m		●			●	●							
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S** : Источник данных
- D** : Результат операции
- m** : Выбор метода преобразования
- n** : Длина используемых данных

**Описание:**

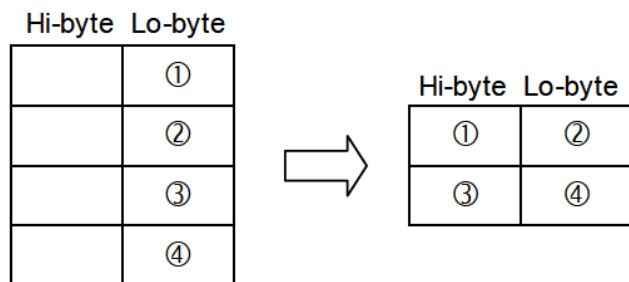
1. Параметр **m** предназначен для выбора режима преобразования согласно следующей таблице. Подробнее о режимах см. ниже. Если значение параметра не является одним из значений в таблице ниже, преобразование или перемещение данных производиться не будет, сообщение об ошибке появляться также не будет.

Значение m	Описание
0	8-битные данные преобразуются в 16-битные данные (старшие 8 бит, младшие 8 бит)
1	8-битные данные преобразуются в 16-битные данные (младшие 8 бит, старшие 8 бит)
2	16-битные данные (старшие 8 бит, младшие 8 бит) преобразуются в 8-битные данные
3	16-битные данные (младшие 8 бит, старшие 8 бит) преобразуются в 8-битные данные
4	8-битные шестнадцатеричные данные (старшие 4 бита, младшие 4 бита) преобразуются в данные формата ASCII
5	8-битные шестнадцатеричные данные (младшие 4 бита, старшие 4 бита) преобразуются в данные формата ASCII
6	8-битные данные формата ASCII преобразуются в шестнадцатеричные данные (старшие 4 бита, младшие 4 бита)
7	8-битные данные формата ASCII преобразуются в шестнадцатеричные данные (младшие 4 бита, старшие 4 бита)

2.  $n$  – значение настройки длины данных. Диапазон настройки составляет 1 ~ 256. Если значение превышает диапазон, ПЛК будет выполнять инструкцию с минимальным или максимальным значением.
3. Методы преобразования и перемещения данных показаны ниже (на схемах: Lo-byte – младший байт, Hi-byte – старший байт).

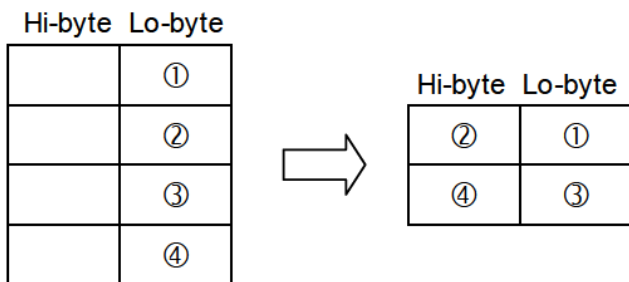
- При  $m=0$ :

Когда  $n=4$ , 8-битные данные преобразуются в 16-битные данные (старшие 8 бит, младшие 8 бит), преобразование показано на рис. ниже:



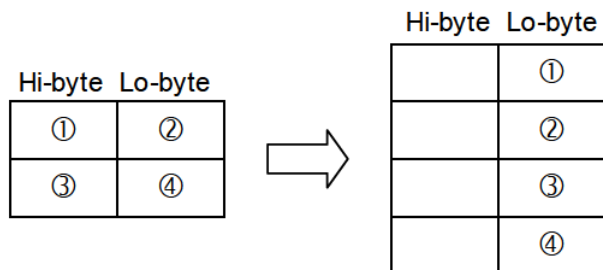
- При  $m=1$ :

Когда  $n=4$ , 8-битные данные преобразуются в 16-битные данные (младшие 8 бит, старшие 8 бит), преобразование показано на рис. ниже:



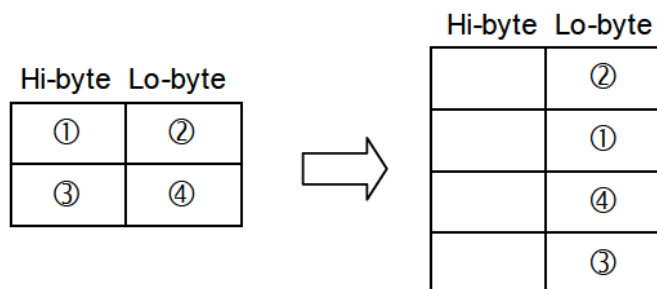
- При  $m=2$ :

Когда  $n=4$ , 16-битные данные (старшие 8 бит, младшие 8 бит) преобразуются в 8-битные данные, преобразование показано на рис. ниже:



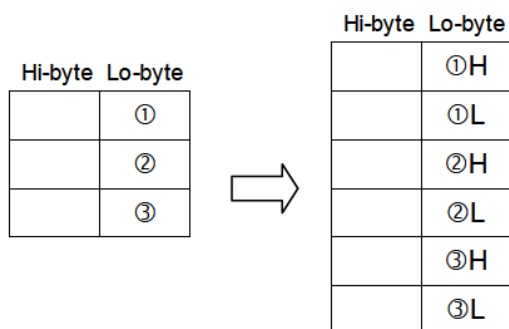
- При  $m=3$ :

Когда  $n=2$ , 16-битные данные (младшие 8 бит, старшие 8 бит) преобразуются в 8-битные данные, преобразование показано на рис. ниже:



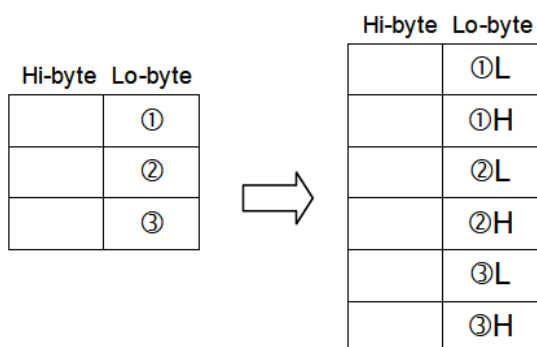
- При  $m=4$ :

Когда  $n=3$ , 8-битные шестнадцатеричные данные (старшие 4 бита, младшие 4 бита) преобразуются в данные формата ASCII, преобразование показано на рис. ниже:



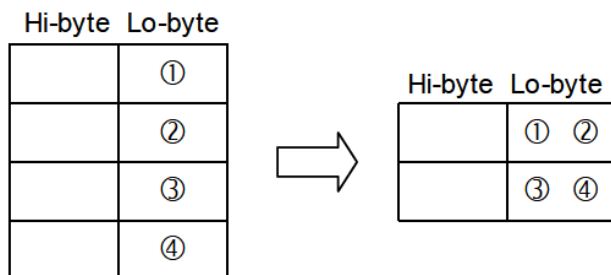
- При  $m=5$ :

Когда  $n=4$ , 8-битные шестнадцатеричные данные (младшие 4 бита, старшие 4 бита) преобразуются в данные формата ASCII, преобразование показано на рис. ниже:



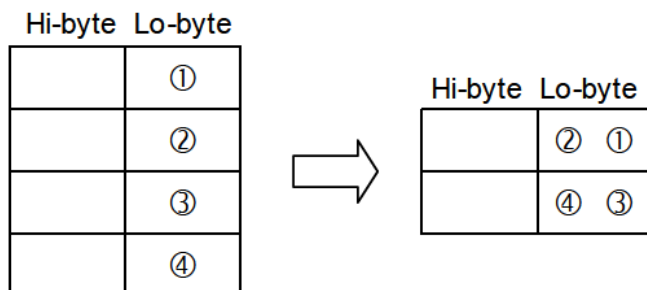
- При  $m=6$ :

Когда  $n=4$ , 8-битные данные формата ASCII преобразуются в шестнадцатеричные данные (старшие 4 бита, младшие 4 бита), преобразование показано на рис. ниже:



- При  $m=7$ :

Когда  $n=4$ , 8-битные данные формата ASCII преобразуются в шестнадцатеричные данные (младшие 4 бита, старшие 4 бита), преобразование показано на рис. ниже:



**Примечание:**

**6**

1. Для **S** и **D** рекомендуется использовать матричные переменные, тип данных Word.

## 6.14 Инструкции по созданию структуры

### 6.14.1 Описание инструкций по созданию структуры

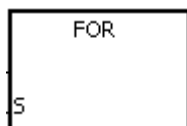
API	Код инструкции			Операнд								Функция					
1300		FOR		S								Начало вложенного цикла					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●			●	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●				●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:



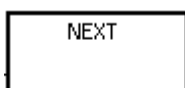
**S** : Количество повторений цикла

Описание:

Подробнее см. описание инструкции NEXT в параграфе, посвященном API1301.

API	Код инструкции	Операнд	Функция
1301	NEXT	-	Конец вложенного цикла
		Импульсная инструкция	16-битная инструкция
		-	AS
			32-битная инструкция
			-

**Символьное обозначение:**

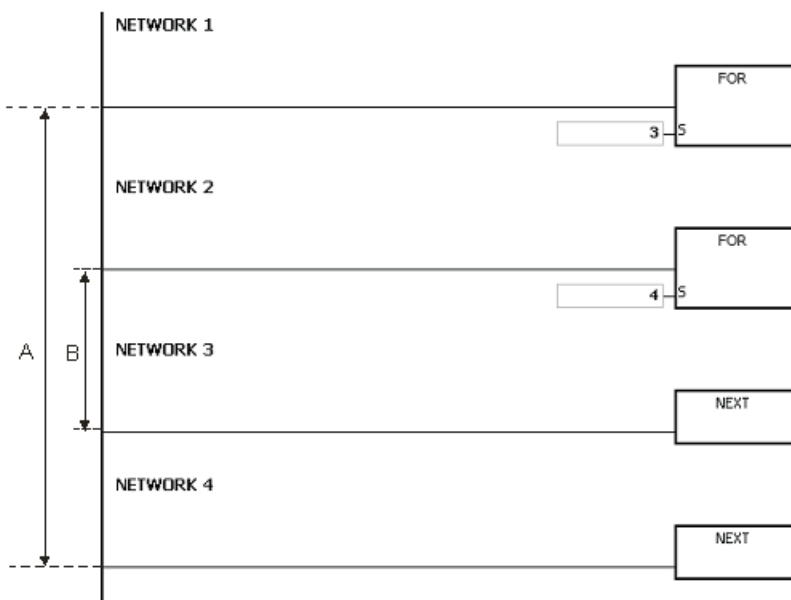


**Описание:**

1. Программа между инструкциями FOR и NEXT выполняется N раз. После выполнения программы между инструкциями FOR и NEXT N раз, выполняется программа, следующая за инструкцией NEXT. Инструкция FOR указывает, сколько раз выполняется программа между инструкциями FOR и NEXT.
2. N может принимать значение в диапазоне от 1 до 32,767. Если значение N меньше 1, оно считается равным 1.
3. Если программа между инструкциями FOR и NEXT не выполняется, ее можно пропустить с помощью инструкции CJ.
4. Следующие условия приводят к возникновению ошибки.
  - Инструкция NEXT идет перед инструкцией FOR.
  - Инструкция FOR присутствует, а инструкция NEXT отсутствует.
  - Инструкция NEXT следует за инструкцией FEND или END.
  - Количество раз, когда используется инструкция FOR, отличается от количества раз, когда используется инструкция NEXT.
5. Инструкция FOR/NEXT поддерживает вложенную программную структуру. Существует не более 32 уровней вложенных программных структур. Если цикл выполняется много раз, на сканирование программы в ПЛК уходит больше времени и выдается ошибка сторожевого таймера. Для решения данной проблемы используется инструкция WDT.

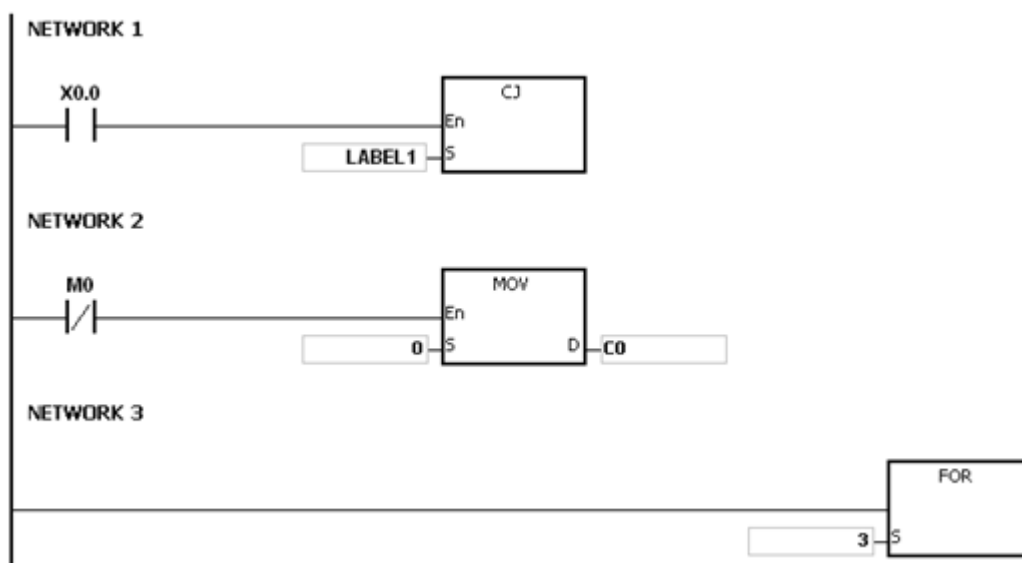
**Пример 1:**

После выполнения программы А три раза, выполняется программа, следующая за инструкцией NEXT. Программа В выполняется четыре раза каждый раз, когда выполняется программа. Поэтому программа В суммарно выполняется двенадцать раз.

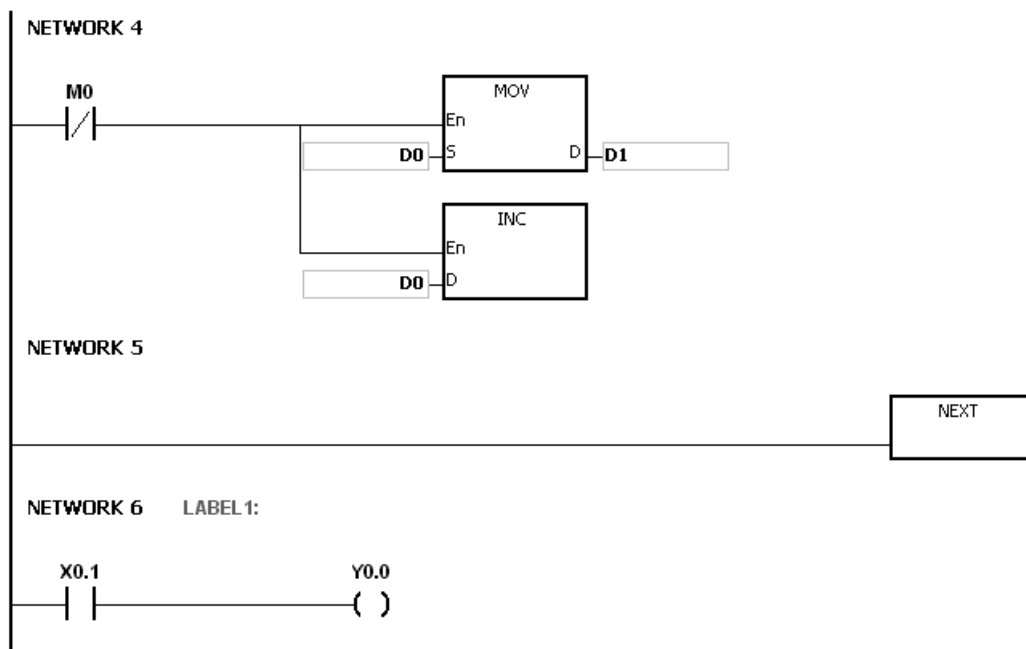


**Пример 2:**

Если состояние входа X0.0 = OFF, выполняется программа между инструкциями FOR и NEXT. Когда состояние входа X0.0 = ON, выполняется инструкция CJ. При выполнении программа перейдет к пункту LABEL 1:, т. е. цепочка 6 и цепочки 4~5 не выполняются.



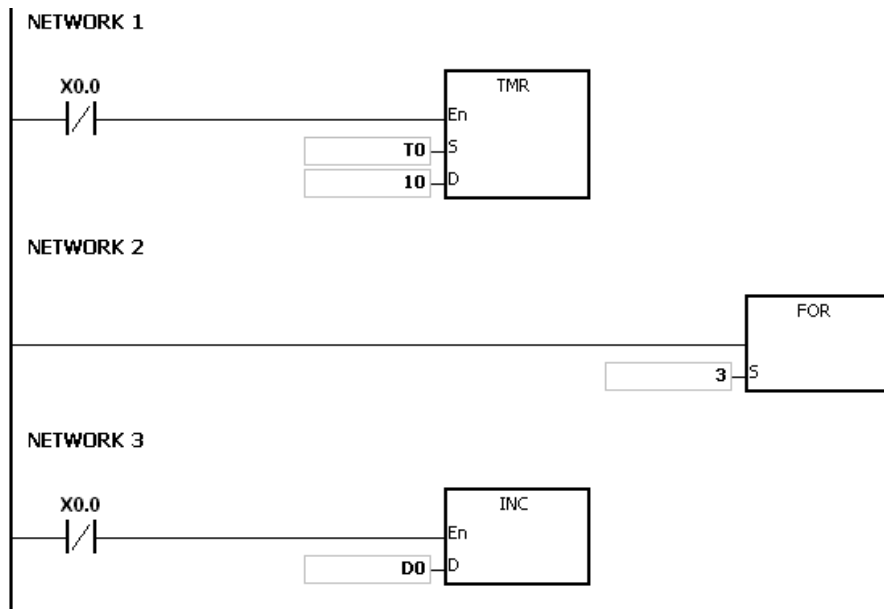


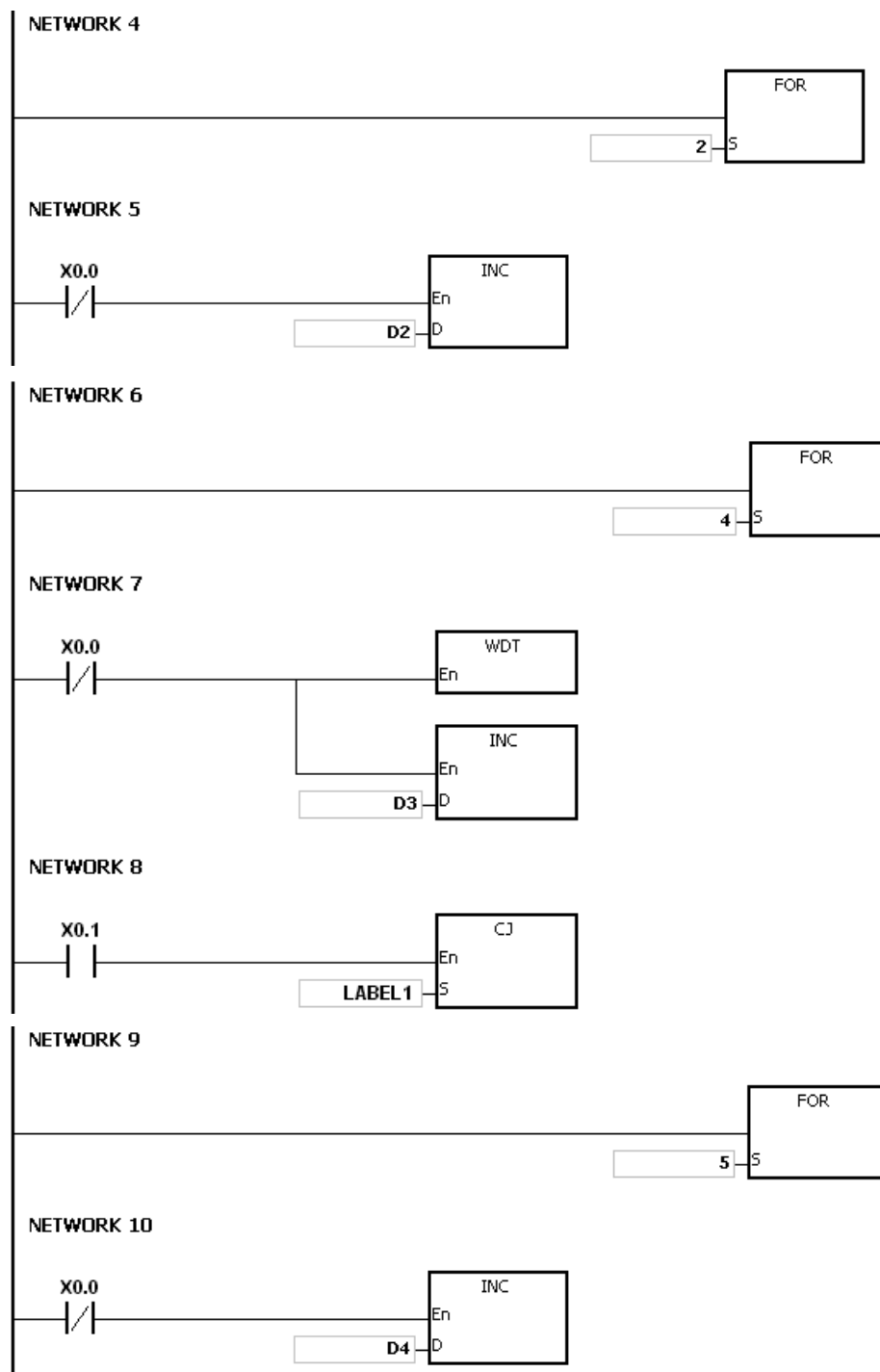


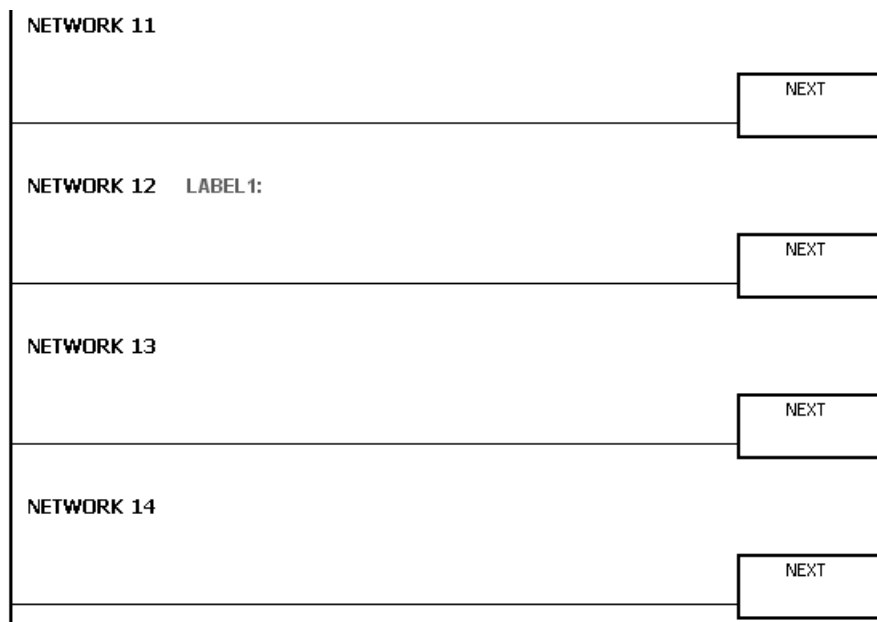
**Пример 3:**

Если программа между инструкциями FOR и NEXT не выполняется, ее можно пропустить с помощью инструкции CJ. Когда состояние входа X0.1 8 = ON в цепочке 8, выполняется инструкция CJ. При выполнении программа перейдет к пункту LABEL 1:, т. е. цепочка 12 и цепочки 9~11 не выполняются.

6







**Дополнительные замечания:**

Подробнее см. данные по использованию метки в руководстве по эксплуатации редактора ISPSOft.

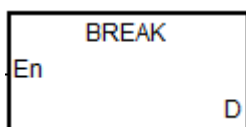
API	Код инструкции			Операнд							Функция					
1302		BREAK		D							Конец цикла FOR-NEXT					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●				●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



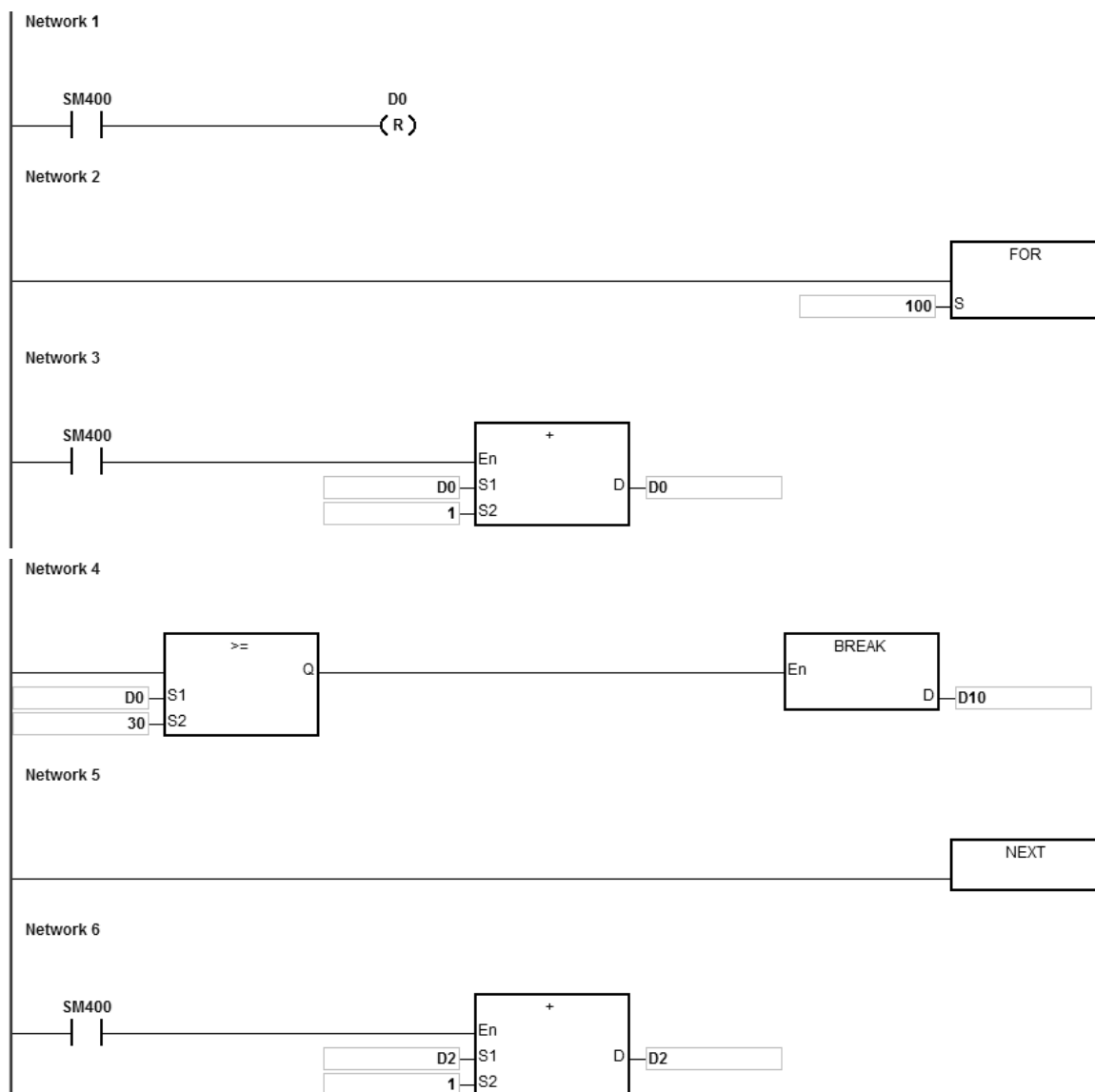
**D** : Регистр, в котором хранится оставшееся количество циклов, которое может быть выполнено

**Описание:**

1. Инструкция BREAK используется для окончания цикла FOR/NEXT. Оставшееся количество циклов FOR/NEXT хранится в операнде D, а программа переходит к инструкции NEXT и начинается выполнение следующей инструкции.
2. После выполнения инструкции BREAK оставшееся количество циклов FOR/NEXT сохраняется в операнде D, включая время, в течение которого выполняется инструкция BREAK.
3. После выполнения инструкции BREAK в первый раз для окончания цикла FOR/NEXT программа не может перейти к инструкции NEXT для выполнения следующей инструкции. Однако, если инструкция BREAK выполняется больше одного раза для окончания цикла FOR/NEXT, программа может перейти к инструкции NEXT для выполнения следующей инструкции.

**Пример:**

По завершении цикла FOR/NEXT к значению в D0 прибавляется 1. Если значение в D0 равно 30, цикл FOR/NEXT заканчивается и оставшееся количество циклов FOR/NEXT, т. е. 71, сохраняется в D10. При выполнении программа перейдет к пункту LABEL 1; т. е. цепочки 6 и 1 прибавятся к значению в D2.



**Дополнительные замечания:**

1. Если инструкция BREAK не входит в цикл FOR/NEXT, выдается ошибка операции, инструкция не выполняется, SM0 = ON, а у SR0 код ошибки 16#2017.
2. Подробнее см. данные по использованию метки в руководстве по эксплуатации редактора ISPSOft.

## 6.15 Инструкции модулей

### 6.15.1 Описание инструкций модулей

API	Код инструкции			Операнд							Функция						
1400	D	FROM	P	$m_1, m_2, m_3, D_1, D_2, n$							Чтение данных из регистра управления модуля расширения						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$m_1$					●	●	●	●	●		○	○	○	○		
$m_2$					●	●	●	●	●		○	○	○	○		
$m_3$					●	●	●	●	●		○	○	○	○		
$D_1$					●	●	●	●								
$D_2$					●	●	●	●								
$n$					●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$m_1$		●	●		●	●	●						
$m_2$		●	●		●	●	●						
$m_3$		●	●		●	●	●						
$D_1$		●	●		●	●	●						
$D_2$		●	●		●	●	●						
$n$		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

Символьное обозначение:

FROM	FROMP
En	En
$m_1$	$m_1$
$m_2$	$m_2$
$m_3$	$m_3$
$n$	$n$
D <sub>1</sub>	D <sub>1</sub>
D <sub>2</sub>	D <sub>2</sub>

DFROM	DFROMP
En	En
$m_1$	$m_1$
$m_2$	$m_2$
$m_3$	$m_3$
$n$	$n$
D <sub>1</sub>	D <sub>1</sub>
D <sub>2</sub>	D <sub>2</sub>

- $m_1$  : Номер процессорного модуля или удаленного модуля
- $m_2$  : Порядковый номер дополнительного модуля
- $m_3$  : Номер регистра управления
- $D_1$  : Регистр, в котором хранятся данные
- $D_2$  : Регистр, в котором хранится код ошибки
- $n$  : Длина данных

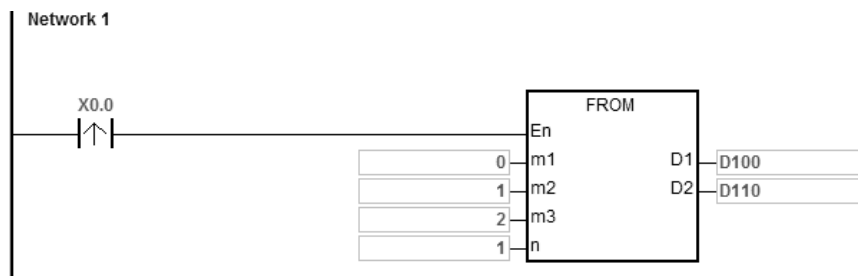
Описание:

1. Данная инструкция предназначена для чтения данных из регистра управления модуля расширения.
2. Операнд  $m_1$  может иметь значение в диапазоне от 0 до 16. 0 обозначает процессорный модуль, а 1~16 обозначает модуль расширения.

3. Операнд  $m_2$  представляет порядковый номер удаленных модулей или модулей расширения, подсоединенных к процессорному модулю по порядку, начиная справа. У первого модуля порядковый номер 1, у второго порядковый номер 2 и т. д. Всего может подсоединяться до 32 модулей, и номер получает каждый модуль независимо от его типа.
4. Операнд  $m_3$  показывает номер регистра управления.
5. При выполнении инструкции FROM операнд  $D_2$  становится равным 0. Если появляется ошибка, значение операнда  $D_2$  перестает быть равным 0. Подробнее по кодам ошибок см. замечания ниже. Если инструкция не исполняется, у операнда  $D_2$  нет никакого кода ошибки.
6. Операнд  $m_1$  может иметь значение в диапазоне от 1 до 8.
7. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.

**Пример:**

Когда состояние входа X0.0 меняется с OFF на ON, выполняется инструкция FROM. Производится чтение данных, хранящихся в CR#2 первого модуля с правой стороны и запись этих данных в D100. Если при этом ошибки нет, в регистре D110 записывается код 16#0000.



**Дополнительные замечания:**

1. Если значения операндов  $m_1$  и  $m_2$  превышают допустимый диапазон значений, это означает ошибку, инструкция не исполняется, SM0 = ON, а в регистр SR0 записывается код ошибки 16#2003.
2. Если  $D_1 \sim D_1+n-1$  больше допустимого диапазона значений регистра, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.
3. Если значение операнда  $n$  превышает допустимый диапазон значений, это означает ошибку, инструкция не исполняется, SM0 = ON, а в регистр SR0 записывается код ошибки 16#200B.
4. Применение инструкции FROM снижает производительность процессорного модуля и соответствующего модуля ввода/вывода, рекомендуется пользоваться этой инструкцией как импульсной, как показано на примере ниже.

5. Если при выполнении инструкции появляется любая ошибка от модуля, в регистр **D<sub>2</sub>** записывается код ошибки. Описание кодов ошибок см. ниже:

Код ошибки	Описание
16#1400	Чтение данных из регистра управления (CR) модуля, у которого нет регистра управления с таким номером.
16#1401	Значение, вводимое в модуль, недействительное.
16#1402	Нет ответа от модуля. Истекло время ожидания ответа (таймаут).



API	Код инструкции			Операнд							Функция					
1401	D	TO	P	$m_1, m_2, m_3, S, D, n$							Запись данных в регистр управления модуля расширения					

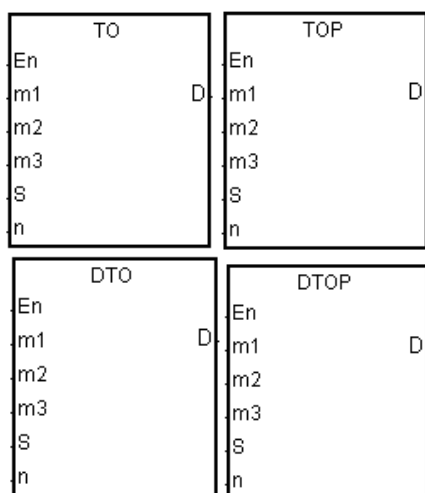
Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$m_1$					●	●	●	●	●		○	○	○	○		
$m_2$					●	●	●	●	●		○	○	○	○		
$m_3$					●	●	●	●	●		○	○	○	○		
<b>S</b>					●	●	●	●	●				○	○		
<b>D</b>					●	●	●	●								
<b>n</b>					●	●	●	●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$m_1$		●	●		●	●	●						
$m_2$		●	●		●	●	●						
$m_3$		●	●		●	●	●						
<b>S</b>		●	●		●	●	●						
<b>D</b>		●	●		●	●	●						
<b>n</b>		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

6

**Символьное обозначение:**



- $m_1$  : Номер процессорного модуля или удаленного модуля
- $m_2$  : Порядковый номер дополнительного модуля
- $m_3$  : Номер регистра управления
- S** : Регистр, в котором хранятся данные
- D** : Регистр, в котором хранится код ошибки
- n** : Длина данных

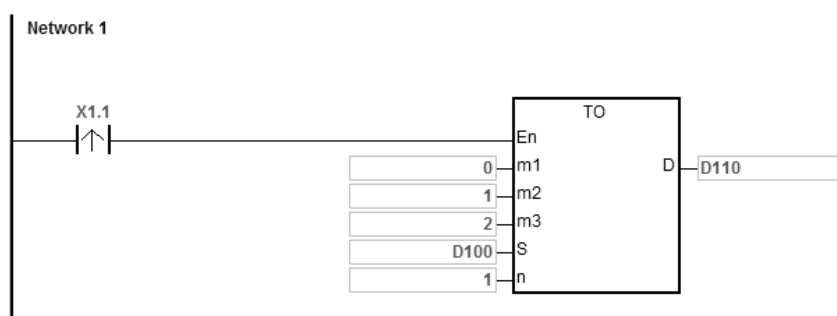
**Описание:**

1. Данная инструкция предназначена для записи данных в регистр управления модуля расширения.
2. Операнд  $m_1$  может иметь значение в диапазоне от 0 до 16. 0 обозначает процессорный модуль, а 1-16 обозначает модуль расширения.

3. Операнд  $m_2$  представляет порядковый номер удаленных модулей или модулей расширения, подсоединенных к процессорному модулю по порядку, начиная справа. У первого модуля порядковый номер 1, у второго порядковый номер 2 и т. д. Всего может подсоединяться до 32 модулей, и номер получает каждый модуль независимо от его типа.
4. Операнд  $m_3$  показывает номер регистра управления.
5. При выполнении инструкции FROM операнд  $D_2$  становится равным 0. Если появляется ошибка, значение операнда  $D_2$  перестает быть равным 0. Подробнее по кодам ошибок см. замечания ниже. Если инструкция не исполняется, у операнда  $D_2$  нет никакого кода ошибки.
6. Операнд  $m_1$  может иметь значение в диапазоне от 1 до 8.
7. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре E – не может.
8. Если у операнда  $S$  шестнадцатеричное значение, шестнадцатеричные значения  $n$  передаются в модуль ввода/вывода. Предположим, что  $S = 16\#0001$  и  $n = 3$ . Три кода  $16\#0001$  будут переданы в модуль ввода/вывода.

**Пример:**

Когда состояние входа X0.0 меняется с OFF на ON, выполняется инструкция TO. Данные, хранящиеся в регистре D100, записываются в регистр CR#2 первого модуля справа. Если при этом ошибки нет, в регистре D110 записывается код  $16\#0000$ .



**Дополнительные замечания:**

1. Если значения операндов  $m_1$  и  $m_2$  превышают допустимый диапазон значений, это означает ошибку, инструкция не исполняется,  $SM0 = ON$ , а в регистр SR0 записывается код ошибки  $16\#2003$ .
2. Если  $D_1 \sim D_{1+n-1}$  больше допустимого диапазона значений регистра, инструкция не выполняется,  $SM0 = ON$  и в регистр SR0 записывается код ошибки  $16\#2003$ .
3. Если значение операнда  $n$  превышает допустимый диапазон значений, это означает ошибку, инструкция не исполняется,  $SM0 = ON$ , а в регистр SR0 записывается код ошибки  $16\#200B$ .

4. Применение инструкции TO снижает производительность процессорного модуля и соответствующего модуля ввода/вывода, рекомендуется пользоваться этой инструкцией как импульсной, как показано на примере ниже.
5. Если при выполнении инструкции появляется любая ошибка от модуля, в регистр D<sub>2</sub> записывается код ошибки. Описание кодов ошибок см. ниже:

Код ошибки	Описание
16#1400	Чтение данных из регистра управления (CR) модуля, у которого нет регистра управления с таким номером.
16#1401	Значение, вводимое в модуль, недействительное.
16#1402	Нет ответа от модуля. Истекло время ожидания ответа (таймаут).

API	Код инструкции			Операнд								Функция					
1402		PUCONF	P	Module ~ Error, ErrCode								Настройка параметров управления выходами модуля PU					
Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F	
Module								●					○	○			
Axis								●					○	○			
Mode								●					○	○			
SSpeed								●					○	○			
Atime								●					○	○			
Dtime								●					○	○			
MSpeed								●					○	○			
Z_no								●					○	○			
Offset								●					○	○			
Done		●	●	●													
Error		●	●	●													
ErrCode								●									

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
Mode		●			●	●							
SSpeed		●			●	●							
Atime		●			●	●							
Dtime		●			●	●							
MSpeed			●				●						
Z_no		●			●	●							
Offset		●			●	●							
Done	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

PUCONF		PUCONFP	
En		En	
Module	Done	Module	Done
Axis	Error	Axis	Error
Mode	ErrCode	Mode	ErrCode
SSpeed		SSpeed	
Atime		Atime	
Dtime		Dtime	
MSpeed		MSpeed	
Z_no		Z_no	
Offset		Offset	

- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- Mode** : Выходной режим
- SSpeed** : Скорость для начальной / конечной частоты
- Atime** : Время разгона
- Dtime** : Время замедления
- MSpeed** : Максимальная выходная частота
- Z\_no** : Число сигналов Z-фазы для определения возвращения в нулевую точку
- Offset** : Смещение положения после поиска Z-фазы и возврата в нулевую точку
- Done** : Флаг выполнения
- Error** : Флаг ошибки
- ErrCode** : Код ошибки

**Описание:**

- Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
- Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.

Ниже представлена комбинация номеров осей и соответствующих выходов модулей PU.

Модуль PU	Ось 1	Ось 2	Ось 3	Ось 4
AS02PU	Y0.0 / Y0.1	Y0.2 / Y0.3	Нет	Нет
AS04PU	Y0.0 / Y0.1	Y0.2 / Y0.3	Y0.4 / Y0.5	Y0.6 / Y0.7

- Mode** задает режим на выходе для оси, значения параметров объясняются в таблице ниже.

Выходной режим	Описание	Примечание
1	Одиарный импульсный выход (точка с четным номером только для применения в качестве выхода)	Например, выход Y0.0 или Y0.2
2	Импульсный выход (точка с четным номером) + направление (точка с нечетным номером)	Например, выход Y0.0 + направление Y0.1
3	CW (точка с четным номером) + CCW (точка с нечетным номером)	Например, CW Y0.0 + CCW Y0.1
4	Фаза A (точка с четным номером) + фаза B (точка с нечетным номером)	Например, фаза A Y0.0 + фаза B Y0.1
Другое значение	Автоматически переключается в режим 2 (режим по умолчанию)	

4. **SSpeed~ Offset**

См. описание ниже следующих энергозависимых параметров и значений уставок. Если значения уставок выходят за допустимые пределы, инструкция будет автоматически выполняться с минимальным или максимальным значением.

Параметр	Функция	Диапазон	Знач. по умолч.	Примечание
<b>SSpeed</b>	Скорость для начальной / конечной частоты	0 ~ 10,000 (Ед.изм: Гц)	100	
<b>Atime</b>	Время разгона	0 ~ 10,000 (Ед.изм: Гц)	100	
<b>Dtime</b>	Время замедления	0 ~ 10,000 (Ед.изм: Гц)	100	
<b>MSpeed</b>	Максимальная выходная частота	100 ~ 200,000 (Ед.изм: Гц)	100К	32-битное значение
<b>Z_no</b>	Число сигналов Z-фазы для определения возвращения в нулевую точку	-100 ~ 100 (Ед.изм.: разы)	0	0: отключено
<b>Offset</b>	Смещение положения после поиска Z-фазы и возврата в нулевую точку	-10,000 ~ 10,000 (Ед.изм.: импульсы)	0	0: отключено

- Done:** флаг завершения для выхода указанного модуля PU. Когда флаг включен, это означает успешное завершение настройки параметра. Можно продолжать выполнять вывод сигнала позиционирования в зависимости от состояния включения флага завершения. Сброс флага Done должен проводиться вручную. Флаг Done включается только после завершения настройки.
- Error:** флаг ошибки параметра выхода указанного модуля PU. Большинство диапазонов параметров автоматически фильтруются ПЛК. Таким образом, если флаг ошибки включен, это означает, что не модуль PU указан, номер модуля PU неверен или неверен номер выходной оси.
- Инструкция представляет собой импульсную инструкцию. Даже если контакт А принимается как контакт состояния, параметры модуля PU активируются только при запуске инструкции. Поэтому, если значение параметра должно быть обновлено, необходимо перезапустить инструкцию.
- Поскольку заданные параметры доставляются через инструкцию коммуникации модуля, подтвердите состояние на выходе (Done или Error) до изменения значения параметра, а затем выполните соответствующие операции.
- ErrCode** показывает коды ошибок. См. описание ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации

API	Код инструкции			Операнд								Функция					
1403		PUSTART		Module ~ Error								Чтение состояния выходов модуля PU					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Module								●					○	○		
Axis								●					○	○		
ZeroS	●	●	●	●												
C_Pos								●								
Execute		●	●	●												
Pause		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
ZeroS	●												
C_Pos			●				●						
Execute	●												
Pause	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:

PUSTAT	
En	
Module	C_Pos
Axis	Execute
ZeroS	Pause
	Error
	ErrCode

- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- ZeroS** : Сброс текущей позиции на выходе на нулевую
- C\_Pos** : Текущая позиция на выходе
- Execute** : Флаг выполнения
- Pause** : Флаг паузы
- Error** : Флаг ошибки
- ErrCode** : Код ошибки

Описание:

- Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция

относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.

2. **Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.
3. **C\_Pos1** устанавливает текущее положение выходной оси для указанного модуля PU. Значение параметра является фиксированным и сохраняется в модуле PU. Если значение должно быть сброшено до 0, включите **ZeroS** при старте работы инструкции.
4. **Execute** – это флаг только для чтения, который показывает наличие выходного сигнала на оси модуля PU. Когда **Execute** включен, это означает, что выходной сигнал есть. Когда **Execute** отключен, это означает, что выход на ось не используется и может принимать следующий выходной сигнал.
5. **Pause** – это флаг только для чтения, который показывает приостановку вывода выходного сигнала на оси модуля PU. Когда **Pause** включен, это означает, что вывод сигнала приостановлен, текущая скорость равна 0, а текущий выход не достиг определенной заданной выходной позиции. Если восстановить вывод сигнала, флаг будет сброшен автоматически.

Примечание. В то время как включен флаг **Pause**, флаг **Execute** также постоянно включен.

6. **Error** – это флаг ошибки (только для чтения), он показывает, что в процессе чтения состояния указанного модуля PU произошла ошибка. См. коды ошибок в **ErrCode**.
7. После того, как инструкция PUSTAT выдаст команду паузы, флаги **Execute**, **Pause** и **Error** станут флагами только для чтения, и их состояние не может быть изменено. Флаги **Execute**, **Pause** и **Error** могут быть установлены или сброшены только при отключенной инструкции PUSTAT.
8. **ErrCode** показывает коды ошибок. См. описание ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено



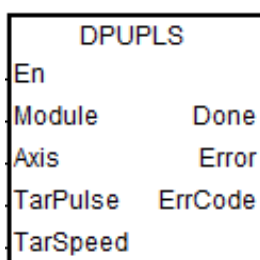
API	Код инструкции			Операнд								Функция				
1404	D	PUPLS		Module ~ ErrCode								Импульсный выход модуля PU (без разгона)				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Module								●					○	○		
Axis								●					○	○		
TarPulse								●					○	○		
TarSpeed								●					○	○		
Done		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
TarPulse			●				●						
TarSpee			●				●						
Done	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символьное обозначение:



**Module** : Номер модуля

**Axis** : Номер оси на выходе модуля

**TarPulse** : Заданное число выходных импульсов

**TarSpeed** : Заданная выходная частота

**Done** : Флаг выполнения

**Error** : Флаг ошибки

**ErrCode** : Код ошибки

Описание:

- Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
- Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход

на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.

3. **TarPulse** задает количество выходных импульсов. Номер импульса представляет собой положительное 32-битное значение. Когда значение равно 0, это означает, что выход всегда выполняется, количество выходных импульсов не ограничено, и выход не отключается до тех пор, пока инструкция не будет остановлена. Когда значение меньше 0, ПЛК автоматически использует 2-е дополнение для преобразования значения в положительное целое число как количество выходных импульсов.
4. **TarSpeed** устанавливает заданную выходную скорость (единица измерения: Гц). Входное значение представляет собой 32-битное значение со знаком в диапазоне от -100 000 (-100K) ~ 100 000 (100K). Вы можете изменить заданную частоту в любое время после включения инструкции, и модуль PU автоматически переключится на вновь установленную заданную частоту после вывода всех импульсов.

Примечание. Перед изменением заданной частоты учитывайте, соответствуют ли друг другу измененная скорость и время сканирования ПЛК.

5. Когда **TarSpeed** является положительным числом (> 0), это означает, что выход «положительного направления» выключен. Когда **TarSpeed** является отрицательным числом (<0), это означает, что выход «отрицательного направления» включен. Когда значение **TarSpeed** равно 0, это означает, что выходной сигнал будет приостановлен после вывода всех импульсов.
6. Инструкция не поддерживает функцию ускорения и замедления. Если необходима функция ускорения и замедления, используйте инструкцию DPUDRI.
7. Эта инструкция может использоваться для изменения скорости. Пока инструкция выполняется, вы можете изменить значение **TarSpeed**, чтобы изменить выходную скорость.
8. Когда достигнуто количество выходных импульсов, заданное **TarPulse**, флаг **Done** включается. Сбрасывается флаг **Done** вручную. Инструкция устанавливает этот флаг только в том случае, если вывод выходных импульсов завершен.
9. Если при работе инструкции в процессе вывода выходных импульсов возникает ошибка, включается флаг ошибки. Коды ошибок отображаются в **ErrCode**.
10. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено

API	Код инструкции			Операнд								Функция			
1405	D	PUDRI		Module ~ ErrCode								Относительная выходная позиция модуля PU (с разгоном и замедлением)			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Module								●					○	○		
Axis								●					○	○		
RTarPosi								●					○	○		
TarSpeed								●					○	○		
Done		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
RTarPosi			●				●						
TarSpeed			●				●						
Done	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символьное обозначение:

DPUDRI	
En	
Module	Down
Axis	Error
RTarPosi	ErrCode
TarSpeed	

- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- RTarPosi** : Число выходных импульсов для относительного позиционирования
- TarSpeed** : Заданная выходная частота
- Done** : Флаг выполнения
- Error** : Флаг ошибки
- ErrCode** : Код ошибки

Описание:

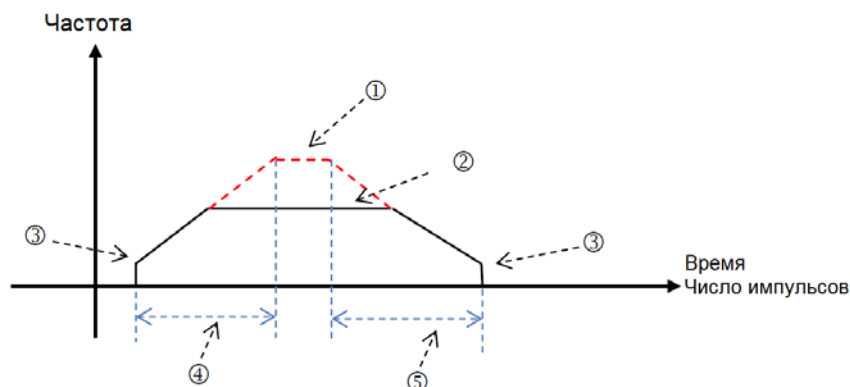
- Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
- Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход

на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.

3. **RTarPosi** устанавливает положение для относительного позиционирования. Число импульсов – 32-битное значение. Когда значение больше 0, сигнал на выходе будет определять положительное направление (и выход, определяющий направление, отключен). Когда значение меньше 0, сигнал на выходе будет определять отрицательное направление (и выход, определяющий направление, включен). Когда значение равно 0, включается флаг выполнения **Done**.
4. **TarSpeed** устанавливает заданную выходную частоту (единица измерения: Гц). Значением частоты является положительное 32-битное целое число. Когда значение меньше 0, инструкция автоматически использует 2-е дополнение для преобразования значения в положительное целое число. Когда значение равно 0, инструкция будет уведомлять модуль о входе в режим паузы. Устройство на выходе замедляется на скорости торможения до 0, при этом включается флаг паузы.
5. После запуска выхода заданная частота может меняться в любое время. При фактическом изменении частоты ПЛК автоматически изменяет частоту, основанную на установленном ускорении и замедлении в инструкции PUCONF.
6. Когда достигнуто число импульсов импульсов для относительного позиционирования, заданное в **RTarPosi**, флаг **Done** включается. Сбрасывается флаг **Done** вручную. Инструкция устанавливает этот флаг только в том случае, если вывод выходных импульсов завершен.
7. Если при работе инструкции в процессе вывода выходных импульсов возникает ошибка, включается флаг ошибки. Коды ошибок отображаются в **ErrCode**.
8. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено

9. График кривой разгона / замедления для инструкции DPUDRI



① Максимальное значение выходной частоты. Обратитесь к настройке в инструкции PUCONF для настройки параметра.

В качестве альтернативы, установите значение параметра через HWCONFIG.

② Заданная частота, указанная в инструкции для выходов модуля PU. Выходной сигнал не должен превышать максимальную выходную частоту. Если максимальная выходная частота превышена, максимальная выходная частота рассматривается как выходная частота.

③ Начальная / конечная выходная частота. Обратитесь к инструкции PUCONF для настройки параметра.

В качестве альтернативы, установите значение параметра через HWCONFIG.

④ Время разгона. Обратитесь к инструкции PUCONF для настройки параметра.

В качестве альтернативы, установите значение параметра через HWCONFIG.

⑤ Время торможения. Обратитесь к инструкции PUCONF для настройки параметра.

В качестве альтернативы, установите значение параметра через HWCONFIG.

Разгон и торможение, управляемые модулем PU, выполняются с фиксированным наклоном. Таким образом, фактическое время разгона и торможения изменяется в зависимости от заданной выходной частоты. Формулы для расчета разгона и торможения соответственно показаны ниже:

$(\text{Максимальная выходная частота} - \text{начальная частота}) / \text{время разгона};$

$(\text{Максимальная выходная частота} - \text{конечная частота}) / \text{время торможения}.$

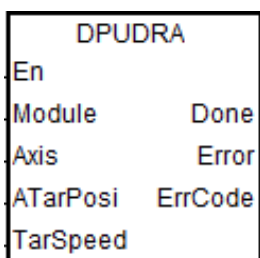
API	Код инструкции			Операнд								Функция					
1406	D	PUDRA		Module ~ ErrCode								Абсолютный адресный выход модуля PU (с разгоном и замедлением)					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Module								●					○	○		
Axis								●					○	○		
RTarPosi								●					○	○		
TarSpeed								●					○	○		
Done		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
RTarPosi			●				●						
TarSpeed			●				●						
Done	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символьное обозначение:



- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- RTarPosi** : Число выходных импульсов для абсолютной адресации
- TarSpeed** : Заданная выходная частота
- Done** : Флаг выполнения
- Error** : Флаг ошибки
- ErrCode** : Код ошибки

Описание:

- ATarPosi** – это позиционирование для абсолютной адресации. Число импульсов представляет собой знаковое 32-битное значение. Модуль PU будет автоматически сравнивать его с текущей позицией. Когда значение больше 0, сигнал на выходе будет определять положительное направление (и выход, определяющий направление, отключен). Когда значение меньше 0, сигнал на выходе будет определять отрицательное направление (и выход, определяющий направление, включен). Когда значение равно 0,

включается флаг выполнения **Done**.

2. См. описание инструкции DPUDRI для настройки других параметров.

API	Код инструкции				Операнд								Функция				
1407		PUZRN			Module ~ ErrCode								Возврат в нулевую точку для модуля PU				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Module								●					○	○		
Axis								●					○	○		
Mode								●					○	○		
TarSpeed								●					○	○		
JogSpeed								●					○	○		
Done		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
Mode		●			●	●							
TarSpeed			●				●						
JogSpeed		●			●	●							
Done	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:

PUZRN	
En	
Module	Done
Axis	Error
Mode	ErrCode
TarSpeed	
JogSpeed	

- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- Mode** : Выбор режима возврата в нулевую точку
- TarSpeed** : Заданная выходная частота
- JogSpeed** : Частота Jog режима для возврата в нулевую точку
- Done** : Флаг выполнения
- Error** : Флаг ошибки
- ErrCode** : Код ошибки



**Описание:**

1. **Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
2. **Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.
3. **Mode** задает режим возврата в нулевую точку. Описание режимов показано в таблице ниже:

Режим	Функция	Точка входа, используемая вместе с другими настройками	Примечание
0	Непосредственный сброс текущей позиции до 0	Нет	
1	Задание точки, в которой точка DOG коснется нулевой точки; движение по оси начинается в отрицательном направлении, оставляя позицию точки DOG	DOG	Использование настроек HWCONFIG
2	Задание точки, в которой точка DOG коснется нулевой точки; движение по оси начинается в положительном направлении, оставляя позицию точки DOG	DOG	Использование настроек HWCONFIG
3	После завершения режима Mode = 1, поиск заданного номера Z-фазы	DOG и вход Z-фазы	Использование инструкции PUCONF
4	После завершения режима Mode = 2, поиск заданного номера Z-фазы	DOG и вход Z-фазы	Использование инструкции PUCONF
5	После завершения режима Mode = 1, вывод заданного числа выходных импульсов	DOG	Использование инструкции PUCONF
6	После завершения режима Mode = 2, вывод заданного числа выходных импульсов	DOG	Использование инструкции PUCONF
7	Выполнение режима Mode = 1 + положительный и отрицательный пределы	DOG, положительный предел, отрицательный предел	Использование настроек HWCONFIG
8	Выполнение режима Mode = 2 + положительный и отрицательный пределы	DOG, положительный предел, отрицательный предел	Использование настроек HWCONFIG
9	После завершения режима Mode = 1 и отработки положительного и отрицательного пределов, поиск заданного номера Z-фазы	DOG, положительный предел, отрицательный предел и вход Z-фазы	Использование настроек HWCONFIG или инструкции PUCONF
10	После завершения режима Mode = 2 и отработки положительного и отрицательного пределов, поиск заданного номера Z-фазы	DOG, положительный предел, отрицательный предел и вход Z-фазы	Использование настроек HWCONFIG или инструкции PUCONF
11	После завершения режима Mode = 1 и отработки положительного и отрицательного пределов, вывод заданного числа выходных импульсов	DOG, положительный предел, отрицательный предел	Использование настроек HWCONFIG

12	После завершения режима Mode = 2 и обработки положительного и отрицательного пределов, вывод заданного числа выходных импульсов	DOG, положительный предел, отрицательный предел	Использование настроек HWCONFIG
255	Изменение текущей позиции для оси	Нет	Использование заданного значения <b>TarSpeed</b>
Другое	Зарезервированы		

Примечание. Указанные сценарии возврата в нулевую точку не могут не быть реализованы, если точки входа для выбранного режима не используются вместе с настройкой в HWCONFIG.

- TarSpeed** задает максимальную выходную частоту для возврата в нулевую точку. Значение настройки представляет собой 32-битное значение со знаком. Когда значение параметра находится в диапазоне от 0 до 10, диапазон заданного значения составляет 100 ~ 100 000 (Гц). Если значение параметра равно 255, значение **TarSpeed** станет текущим значением выходного положения модуля PU.
- JogSpeed** – это частота Jog режима для достижения нулевой точки. Значение настройки представляет собой 16-битное значение со знаком в диапазоне 1 ~ 10000 (Гц).
- Когда во время выполнения инструкции будет достигнута указанная нулевая точка, включится флаг **Done**. Флаг **Done** сбрасывается вручную. Инструкция устанавливает этот флаг только в том случае, если вывод выходных импульсов завершен.
- Если при работе инструкции в процессе вывода выходных импульсов возникает ошибка, включается флаг ошибки. Коды ошибок отображаются в **ErrCode**.
- Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено

API	Код инструкции			Операнд								Функция				
1408		PUJOG		Module ~ ErrCode								Выход JOG модуля PU				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Module								●					○	○		
Axis								●					○	○		
JogSpeed								●					○	○		
Busy		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
JogSpeed			●				●						
Busy	●												
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:

PUJOG	
En	
Module	Busy
Axis	Error
JogSpeed	ErrCode

- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- JogSpeed** : Выходная частота Jog режима
- Busy** : Выходной сигнал на исполнение
- Error** : Флаг ошибки
- ErrCode** : Код ошибки

Описание:

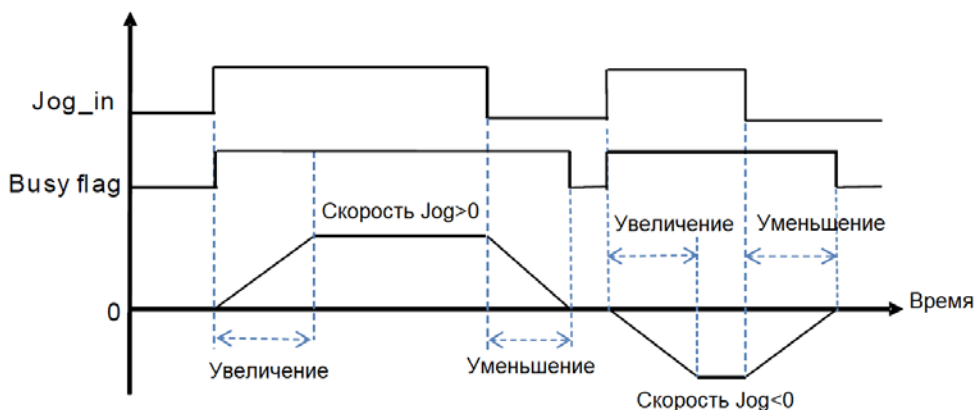
- Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
- Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.
- JogSpeed** задает частоту выходного сигнала. Значение настройки представляет собой 32-битное

значение со знаком в диапазоне от -100 000 до 100 000 (Гц). Когда значение больше 0, сигнал на выходе будет определять положительное направление (и выход, определяющий направление, отключен). Когда значение меньше 0, сигнал на выходе будет определять отрицательное направление (и выход, определяющий направление, включен). Когда значение равно 0, выход отключается.

4. Если при работе инструкции в процессе вывода выходных импульсов возникает ошибка, включается флаг ошибки. Коды ошибок отображаются в **ErrCode**.
5. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено

6. Ниже показана временная диаграмма выходного сигнала. (Jog\_in – это переключатель для запуска инструкции, а флаг Busy – **Busy**):



7. После того, как инструкция PUJOG отключена, а флаг **Busy** выключен, можно выполнить другое управление выходом.

API	Код инструкции			Операнд								Функция				
1409	D	PUMPG		Module ~ ErrCode								Выход MPG модуля PU				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Module								●					○	○		
Axis								●					○	○		
InMode								●					○	○		
InPulse								●								
InSpeed								●								
Rate								●								○
OPulse								●								
OSpeed								●								
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
Axis		●			●	●							
InMode		●			●	●							
InPulse			●				●						
InSpeed			●				●						
Rate									●				
OPulse			●				●						
OSpeed			●				●						
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символьное обозначение:

DPUMPG	
En	
Module	OPulse
Axis	OSpeed
InMode	Error
InPulse	ErrCode
InSpeed	
Rate	

- Module** : Номер модуля
- Axis** : Номер оси на выходе модуля
- InMode** : Входной режим энкодера и умножение частоты для подсчета
- InPulse** : Количество входных импульсов
- InSpeed** : Измеренная входная частота
- Rate** : Коэффициент для ввода/вывода (число с плавающей запятой)
- OPulse** : Количество выходных импульсов
- OSpeed** : Измеренная выходная частота

**Error** : Флаг ошибки

**ErrCode** : Код ошибки

**Описание:**

1. Инструкция PUMPG применима только к модулю AS02PU, прошивка для модуля должна быть версии V1.02.00 или выше.
2. **Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
3. **Axis** задает номер выходной оси для указанного модуля PU. Значения настройки 1 ~ 4 представляют выход на оси 1 ~ 4 указанного модуля PU соответственно. Если модуль PU не имеет соответствующего номера оси, включается флаг ошибки **Error**.
4. **InMode** устанавливает входной режим энкодера и умножение частоты для подсчета.

Значения **InMode** показаны в таблице ниже:

Старшие 8 бит		Младшие 8 бит	
Значение	Функция	Значение	Функция
16#00	Входной сигнал, фазы A/B	16#00	Четырехкратная частота
16#01	Входной сигнал, фаза A	16#01	Исходная частота
16#02	Входной сигнал, CW (A) / CCW (B)	16#02	Двойная частота

Например: Если используется входной сигнал фазы A/B и двойная частота, вводится значение 16#0002.

Не используйте значения, которые не были указаны в приведенной выше таблице, поскольку они представляют зарезервированные функции.

Для метода подсчета фаз A/B и CW/CCW см. описание HC (высокоскоростной счетчик ПЛК). Если используется однофазный вход или вход CW/CCW, можно выбрать только исходную или двойную частоту. Если вводится неправильное значение, инструкция будет использовать по умолчанию исходную частоту.

5. **InPulse** отображает количество уже введенных импульсов, которое представляет собой 32-битное значение со знаком. Каждый раз, когда инструкция начинает работать, модуль PU автоматически сбрасывает значение до 0 и затем начинает подсчет.
6. **InSpeed** отображает уже обнаруженную входную частоту, которая представляет собой 32-битное значение. Основное время для обнаружения частоты составляет 20 мс. Поэтому, если в течение 20 мс нет значения подсчета, обнаруженная входная частота равна 0. Если в течение 20 мс появляется подсчитанное значение, выходной сигнал начинается с минимальной частоты 50 Гц. Даже если значение **OSpeed** ниже 50 Гц с помощью преобразования на основе значения **Rate**, выходной сигнал все равно будет 50 Гц.
7. **Rate** – это коэффициент для ввода/вывода, а значение – число с плавающей запятой. Количество фактических выходных импульсов и частоты соответственно равно числу входных импульсов и частоте, умноженной на значение коэффициента **Rate**.

Например: входная частота составляет 100 Гц, а коэффициент **Rate** – 0,5. Таким образом, выходная частота  $100 \times 0,5 = 50$  Гц. Если максимальная выходная частота после преобразования превышает 100 кГц, выходная частота ограничивается 100 кГц.

Примечание. Длительный максимальный выходной сигнал может привести к тому, что по мере того, как MPG перестает работать, количество выходных сигналов все равно увеличивается, и выходной сигнал необходимо продолжать выдавать, пока он не будет отключен.

8. **OPulse** показывает количество выходных импульсов. **OSpeed** отображает выходную частоту. Они являются 32-битными значениями со знаком.
9. Когда инструкция DPUMPG отключена, проверьте частоту выходного сигнала, достигла ли частота 0. Если инструкция отключена до того, как частота достигнет 0, модуль PU немедленно отключит выход и выдача рассчитанных с помощью коэффициента **Rate** импульсов продолжаться не будет.
10. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено

11. Когда инструкция DPUMPG включается или отключается, ПЛК уведомляет модуль о включении или отключении функции высокоскоростного счетчика. Таким образом, инструкция не может использоваться вместе с инструкцией API1410 DPUCNT. В противном случае может случиться так, что две инструкции одновременно активируют или отключают подсчет для одного модуля.

API	Код инструкции			Операнд								Функция					
1410	D	PUCNT		Module ~ ErrCode								Функция высокоскоростного счетчика модуля PU					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Module								●					○	○		
InMode								●					○	○		
Period								●					○	○		
ZeroS	●	●	●	●												
InPulse								●								
InSpeed								●								
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Module		●			●	●							
InMode		●			●	●							
Period		●			●	●							
ZeroS	●												
InPulse			●				●						
InSpeed			●				●						
Error	●												
ErrCode			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	-	AS

Символьное обозначение:

DPUCNT	
En	
Module	InPulse
InMode	InSpeed
Period	Error
ZeroS	ErrCode

- Module** : Номер модуля
- InMode** : Входной режим энкодера и умножение частоты для подсчета
- Period** : Период времени для регистрации частоты
- ZeroS** : Сброс счетчика на 0
- InPulse** : Число входных импульсов
- InSpeed** : Число импульсов в цикле
- Error** : Флаг ошибки
- ErrCode** : Код ошибки



**Описание:**

1. Инструкция PUCNT применима только к модулю AS02PU.
2. **Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Эта инструкция относится исключительно к модулям PU справа от ЦПУ и не применима к модулям PU справа от удаленного модуля. Если пронумерованный модуль не является модулем PU, включается флаг ошибки **Error**.
3. **InMode** устанавливает входной режим энкодера и умножение частоты для подсчета.

Значения **InMode** показаны в таблице ниже:

Старшие 8 бит		Младшие 8 бит	
Значение	Функция	Значение	Функция
16#00	Входной сигнал, фазы A/B	16#00	Четырехкратная частота
16#01	Входной сигнал, фаза A	16#01	Исходная частота
16#02	Входной сигнал, CW (A) / CCW (B)	16#02	Двойная частота

Например: Если используется входной сигнал фазы A/B и двойная частота, вводится значение 16#0002.

Не используйте значения, которые не были указаны в приведенной выше таблице, поскольку они представляют зарезервированные функции.

Для метода подсчета фаз A/B и CW/CCW см. описание HC (высокоскоростной счетчик ПЛК). Если используется однофазный вход или вход CW/CCW, можно выбрать только исходную или двойную частоту. Если вводится неправильное значение, инструкция будет использовать по умолчанию исходную частоту.

4. **Period** – это заданное значение времени цикла для регистрации частоты в диапазоне от 10 мс до 1000 мс. Если значение настройки превышает данный диапазон, в качестве настройки ПЛК будет автоматически принято максимальное или минимальное значение.
5. **InPulse** отображает количество уже введенных импульсов, которое представляет собой 32-битное значение. Значение подсчета – фиксированное в энергонезависимой памяти значение.  
Если значение нужно сбросить до 0, включите параметр **ZeroS** во время выполнения инструкции.
6. **InSpeed** отображает значение подсчета для каждого периода времени, которое является 32-битным значением. Если вам нужно преобразовать его в значение с единицей измерения Гц, используйте формулу преобразования самостоятельно.
7. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1400	Модуль не поддерживает данную функцию
16#1401	Данные, сохраненные в модуле, некорректны или выходят за допустимые пределы
16#1402	Отсутствует отклик модуля, превышение времени ожидания коммуникации
16#1403	В модуле PU отсутствует заданный номер оси на выходе
16#1404	Неправильная выходная частота на модуле PU
16#1405	Ось, заданная модулем PU, выводит данные. Указание несколько раз одного вывода запрещено

8. Когда инструкция PUCNT включается или отключается, ПЛК уведомляет модуль о включении или отключении функции высокоскоростного счетчика. Таким образом, инструкция не может использоваться вместе с инструкцией API1409 DPUMPG. В противном случае может случиться так, что две инструкции одновременно активируют или отключают подсчет для одного модуля.

API	Код инструкции			Операнд								Функция				
1415		LCCAL		Group, Module ~ ErrCode								Калибровка канала модуля LC (модуль тензодатчиков)				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Group								●					○	○		
Module								●					○	○		
ChNo								●					○	○		
Trigger	●	●	●	●												
TPoint								●					○	○		
TWeight								●								
CPoint								●								○
Done		●	●	●												
ADone		●	●	●												
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Group		●			●	●							
Module		●			●	●							
ChNo		●			●	●							
Trigger	●												
TPoint		●	●		●	●							
TWeight									●				
CPoint		●			●	●							
Done	●												
ADone	●		●										
Error	●												
ErrCode		●	●		●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:

LCCAL	
En	
Group	CPoint
Module	Done
ChNo	ADone
Trigger	Error
TPoint	ErrCode
TWeight	

- Group** : Номер группы
- Module** : Номер модуля
- ChNo** : Номер канала
- Trigger** : Запуск калибровки
- TPoint** : Общее число точек калибровки
- TWeight** : Калиброванное значение массы
- CPoint** : Номер точки, для которой завершена калибровка

- Done** : Флаг выполнения одной калибровки  
**ADone** : Флаг выполнения всех калибровок  
**Error** : Флаг ошибки  
**ErrCode** : Код ошибки

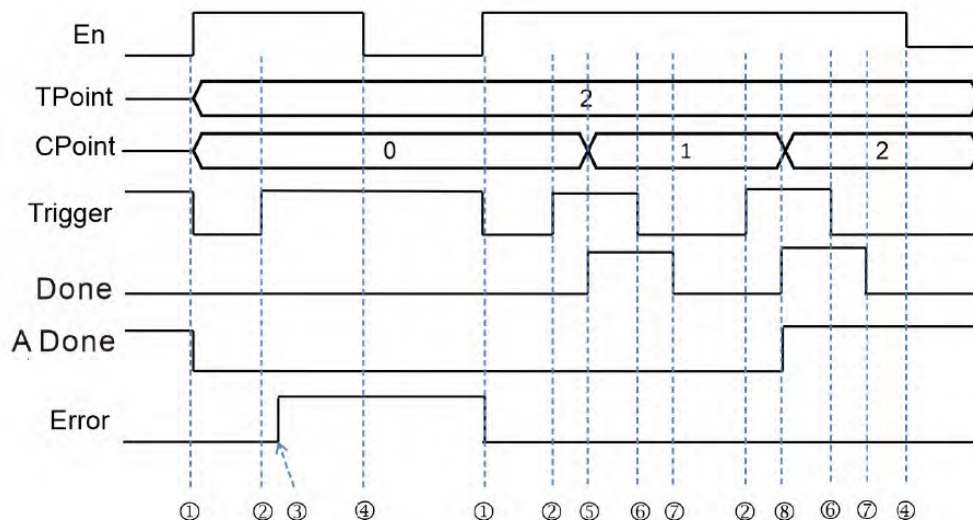
**Описание:**

1. Инструкция LCCAL поддерживается только модулем AS02LC. Прежде чем использовать инструкцию, необходимо определить конфигурационное положение текущего модуля в HWCONFIG.
2. **Group** – это номер группы указанного модуля LC, подключенного справа от ПЛК или удаленного модуля. Номер ПЛК равен 0, номер первого удаленного модуля равен 1 и так далее. Максимальный номер группы - 15. Если указанный модуль не является модулем LC, включается флаг ошибки **Error**.
3. **Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Если указанный модуль не является модулем LC, включается флаг ошибки **Error**.
4. **ChNo** – номер канала указанного модуля LC. Если входное значение не является номером канала модуля LC, включается флаг ошибки **Error**.
5. **Trigger** – это команда запуска одноточечной калибровки. Когда **Trigger** включается, модуль LC будет уведомлен о калибровке. После завершения калибровки включается флаг **Done**. Если выполняется калибровка по всем точкам, также включается флаг **ADone**. Перед калибровкой следующей калибровочной точки необходимо отметить, что **Done** включен, а затем выключить **Trigger**. После этого флаг **Done** будет сброшен.
6. **TPoint** – это количество общих точек для калибровки. После запуска инструкции значение не может быть изменено, т.к. значение **TPoint** было передано в модуль LC для калибровки перед стартом инструкции.
7. Когда **ADone** включен, полная калибровка будет остановлена. Калибровку можно выполнить снова, если снова запустить инструкцию.
8. **CPoint** – это число точек, которые были откалиброваны и не могут быть изменены пользователем. Когда значение **CPoint** >= **TPoint**, инструкция считает калибровку завершенной, включается флаг **ADone**.
9. Когда команда LCCAL запущена первоначально, значение **CPoint** автоматически сбрасывается на 0, **Trigger**, **Done** и **ADone** выключены, калибровка выполняется. После того, как **Trigger** включается и модуль LC завершит калибровку, значение **CPoint** будет автоматически увеличено на 1, а флаг **Done** будет включен. Можно наблюдать текущие точки, для которых калибровка завершена с помощью значения **CPoint**. Например, если значение **CPoint** равно 2, это означает, что триггер запускался дважды, и модуль завершил калибровку двух точек.
10. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1410	Ошибка в номере группы LC или номере модуля
16#1411	Модуль LC не имеет такого номера канала
16#1412	Превышение времени ожидания калибровки модуля

	LC (заданное время: 500 мс)ожидания коммуникации
16#1413	Модуль LC завершил калибровку. Отключите инструкцию и перезапустите его.

11. Пример последовательности действий при калибровке показан на диаграмме ниже:



Обозначения на диаграмме:

- ① Инструкция LCCAL запускается, значение **CPoint** и флаги **Trigger**, **Done**, **ADone** и **Error** автоматически очищаются.
- ② Флаг калибровки включен.
- ③ Инструкция находит ошибку номера модуля и отображает код ошибки номера модуля LC.
- ④ Остановка выполнения инструкции LCCAL.
- ⑤ Модуль LC завершает одноточечную калибровку, значение **CPoint** увеличивается на 1, включается флаг **Done**.
- ⑥ Сброс флага **Trigger**.
- ⑦ Инструкция сбрасывает флаг **Done**.
- ⑧ Модуль LC завершает полную калибровку, значение **CPoint** увеличивается на 1, включаются флаги **Done** и **ADone**.

API	Код инструкции				Операнд								Функция				
1416		LCWEI			Group, Module ~ ErrCode								Чтение значения массы через модуль LC (модуль тензодатчиков)				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Group								●					○	○		
Module								●					○	○		
ChNo								●					○	○		
Stable								●					○	○		
ZeroS	●	●	●	●												
TareS	●	●	●	●												
TareW								●								
Weight								●								
Status								●								
Error		●	●	●												
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Group		●			●	●							
Module		●			●	●							
ChNo		●			●	●							
Stable		●			●	●							
ZeroS	●												
TareS	●												
TareW									●				
Weight									●				
Status		●			●	●							
Error	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:

LCWEI	
En	
Group	Weight
Module	Status
ChNo	Error
Stable	ErrCode
ZeroS	
TareS	
TareW	

- Group** : Номер группы
- Module** : Номер модуля
- ChNo** : Номер канала
- Stable** : Задание диапазона стабильной веса (~ 10000 LSB) (1)
- ZeroS** : Сброс массы на 0
- TareS** : Флаг задания веса тары
- TareW** : Значение веса тары

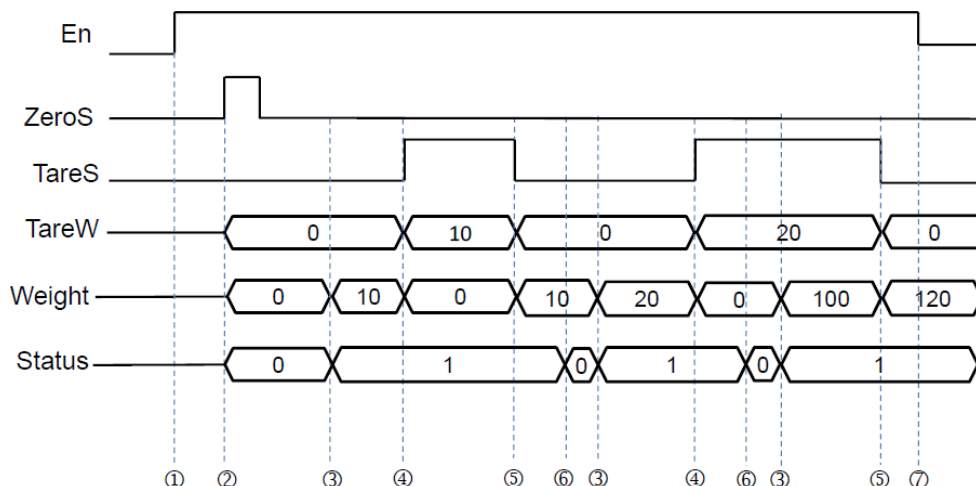
**Weight** : Текущее значение веса  
**Status** : Код состояния модуля LC  
**Error** : Флаг ошибки  
**ErrCode** : Код ошибки

**Описание:**

1. Инструкция LCWEI поддерживается только модулями LC. Прежде чем использовать инструкцию, необходимо определить конфигурационное положение текущего модуля в HWCONFIG.
2. **Group** – это номер группы указанного модуля LC, подключенного справа от ПЛК или удаленного модуля. Номер ПЛК равен 0, номер первого удаленного модуля равен 1 и так далее. Максимальный номер группы - 15. Если указанный модуль не является модулем LC, включается флаг ошибки **Error**.
3. **Module** задает серийный номер модулей справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Максимальное число модулей – 32. Если указанный модуль не является модулем LC, включается флаг ошибки **Error**.
4. **ChNo** – номер канала указанного модуля LC. Если входное значение не является номером канала модуля LC, включается флаг ошибки **Error**.
5. **Stable** задает значение в пределах диапазона стабильного веса. Это исходные данные с единицей измерения LSB. Значение может быть установлено в диапазоне от 1 до 10000, и, если значение настройки выходит за допустимые пределы, максимальное или минимальное значение автоматически принимаются как настройка. Заданное значение параметра действительно при первоначальном запуске инструкции. Если значение в модуле LC необходимо изменить, сначала отключите инструкцию, установите новое значение и затем запустите инструкцию для принятия новой настройки.
6. После того, как инструкция LCWEI включена, указанный канал будет автоматически установлен в режим отображения «вес нетто». Если вам нужно знать вес брутто (общий вес), добавьте значение **TareW** и значение **Weight** самостоятельно.
7. **ZeroS** – это флаг сброса текущего веса на 0. Когда флаг **ZeroS** включается, значения **TareW** и **Weight** сбрасываются на 0.
8. **TareS** – это флаг для установки веса тары. Когда **TareS** включен, значение **TareW** будет равно текущему значению веса, а значение **Weight** будет сброшено на 0. Когда **TareS** выключен, значение **TareW** вернется к текущему значению веса, и значение **TareW** будет сброшено на 0.
9. **Weight** – это значение веса, рассчитанное путем вычета веса тары. Состояние флага **TareW** определяет, включена ли функция веса тары. Когда значение **TareW** равно 0, вес тары не задан.
10. **Status** – это обычно используемый код состояния для инструкции по интеграции модуля LC. См. таблицу ниже.

Значение	0	1	2	3	4	5
Описание	Измерение веса или отсутствие нагрузки	Вес стабильный	Аппаратная ошибка / ошибка калибровки	Процесс калибровки	Вес превышает диапазон	Ошибка номера модуля / номера канала

11. Во время считывания веса, при возникновении ошибки в модуле LC, **Status** отображает соответствующий код ошибки и включается флаг ошибки **Error**. Когда статус вернется в нормальное состояние, флаг ошибки будет сброшен автоматически. Подробнее об ошибках см. Руководство по эксплуатации модуля LC.
12. Пример последовательности действий при измерении веса показан на диаграмме ниже:



Обозначения на диаграмме:

- ① Запуск инструкции LCWEI.
- ② Инструкция сбрасывает данные на 0, далее инструкция сбрасывает **TareW**, **Weight** и **Status**.
- ③ Объект на платформе для взвешивания. Когда измеряется стабильное значение веса, значение **Status** становится равным 1, а **Weight** отображает значение веса.
- ④ Включение **TareS** означает, что значение **Weight** перемещается в **TareW**, а затем значение **Weight** сбрасывается.
- ⑤ Выключение **TareS** означает, что значение **TareW** возвращается к **Weight**, а затем значение **TareW** сбрасывается.
- ⑥ Объект на платформе для взвешивания. В настоящий момент **Status** показывает состояние измерения веса.
- ⑦ Отключение инструкции LCWEI. **TareW**, **Weight** и **Status** сохраняют последние значения и состояние.

13. Коды ошибок, отображаемые **ErrCode**, перечислены в таблице ниже.

Код ошибки	Описание
16#1410	Ошибка в номере группы LC или номере модуля
16#1411	Модуль LC не имеет такого номера канала



## 6.16 Инструкции значений с плавающей запятой

### 6.16.1 Описание инструкций значения с плавающей запятой

API	Код инструкции			Операнд							Функция
1500		FSIN	P	S, D							Синус значения с плавающей запятой

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

Символьное обозначение:

FSIN		FSINP	
En		En	
S	D	S	D

S : Исходное значение

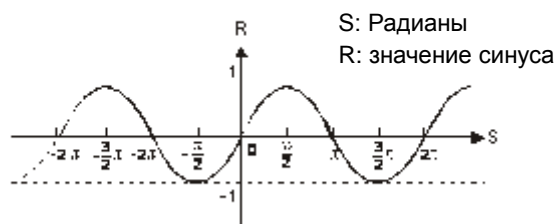
D : Значение синуса

Описание:

1. Реле SM695 определяет исходное значение, задаваемое операндом **S**: Радианы или градусы.
2. Если SM695 = OFF, исходное значение задается операндом **S** в радианах. Радиан = градус × π/180.
3. Если SM695 = ON, исходное значение задается операндом **S** в градусах.

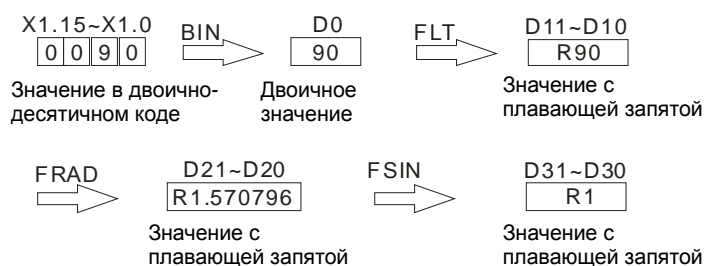
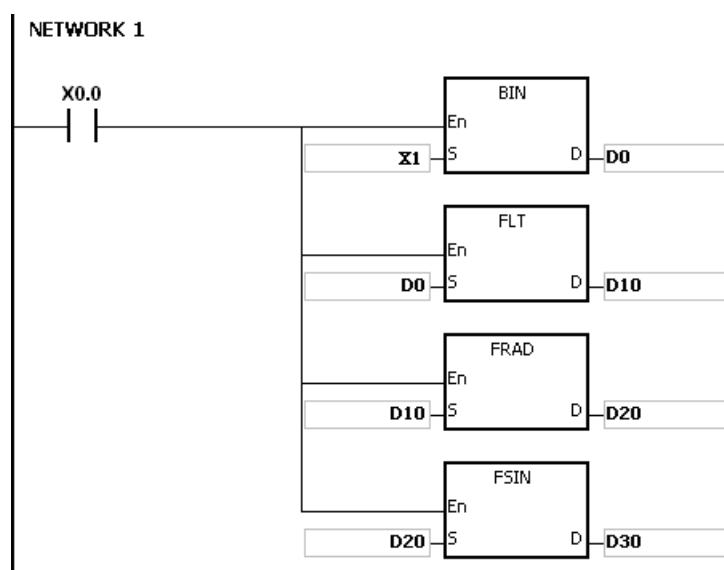
Градус = радиан × 180/π (0° ≤ градус ≤ 360°)

4. Если результат преобразования равен 0, реле SM600 = ON.
5. Синус значения, заданного операндом **S**, записывается в регистр, заданный операндом **D**.
6. Зависимость между значениями синуса и радианами показана ниже.



**Пример:**

Если вход X0.0 = ON, число в двоично-десятичном коде в X1.15~X1.0 преобразуется в двоичное число и результат преобразования записывается в регистр D0. Двоичное значение, записанное в регистре D0, преобразуется в значение с плавающей запятой и записывается в регистр (D11, D10). Значение с плавающей запятой, записанное в регистре (D11, D10), преобразуется в радианы и записывается в регистр (D21, D20). Синус значения в радианах, записанного в регистре (D21, D20), записывается в регистр (D31, D30) и синус представляет собой значение с плавающей точкой.



**Дополнительные замечания:**

1. Если значение операнда S выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.
2. Если реле SM695 = ON и значение операнда S меньше 0 или больше 360, инструкция не выполняется, SM0 = ON и записывается код ошибки 16#2003.

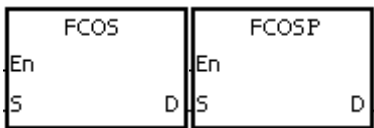
API	Код инструкции			Операнд							Функция						
1501		FCOS	P	<b>S, D</b>							Косинус значения с плавающей запятой						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S</b>	●	●			●	●	●	●	●		○					○
<b>D</b>		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>									●				
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

Символьное обозначение:



**S** : Исходное значение

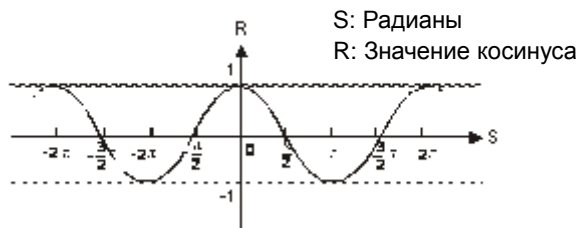
**D** : Значение косинуса

Описание:

1. Реле SM695 определяет исходное значение, задаваемое операндом **S**: Радианы или градусы.
2. Если SM695 = OFF, исходное значение задается операндом **S** в радианах. Радиан = градус  $\times \pi / 180$ .
3. Если SM695 = ON, исходное значение задается операндом **S** в градусах.

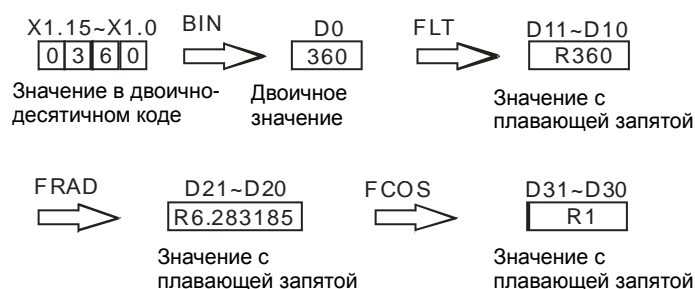
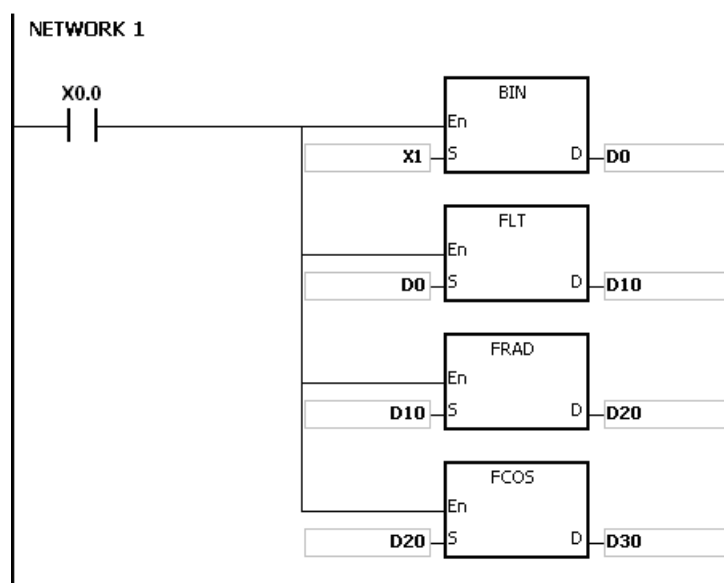
$$\text{Градус} = \text{радиан} \times 180 / \pi \quad (0^\circ \leq \text{градус} \leq 360^\circ)$$

4. Если результат преобразования равен 0, реле SM600 = ON.
5. Косинус значения, заданного операндом **S**, записывается в регистр, заданный операндом **D**.
6. Зависимость между значениями косинуса и радианами показана ниже.



**Пример:**

Если вход X0.0 = ON, значение в двоично-десятичном коде в X1.15~X1.0 преобразуется в двоичное значение и результат преобразования записывается в регистр D0. Двоичное значение, записанное в регистре D0, преобразуется в значение с плавающей запятой и записывается в регистр (D11, D10). Значение с плавающей запятой, записанное в регистре (D11, D10), преобразуется в радианы и записывается в регистр (D21, D20). Косинус значения в радианах, записанного в регистре (D21, D20), записывается в регистр (D31, D30), и косинус представляет собой значение с плавающей точкой.



**Дополнительные замечания:**

1. Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.
2. Если реле SM695 = ON и значение операнда S меньше 0 или больше 360, инструкция не выполняется, SM0 = ON и записывается код ошибки 16#2003.

API	Код инструкции			Операнд							Функция					
1502		FTAN	P	S, D							Тангенс значения с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

Символьное обозначение:

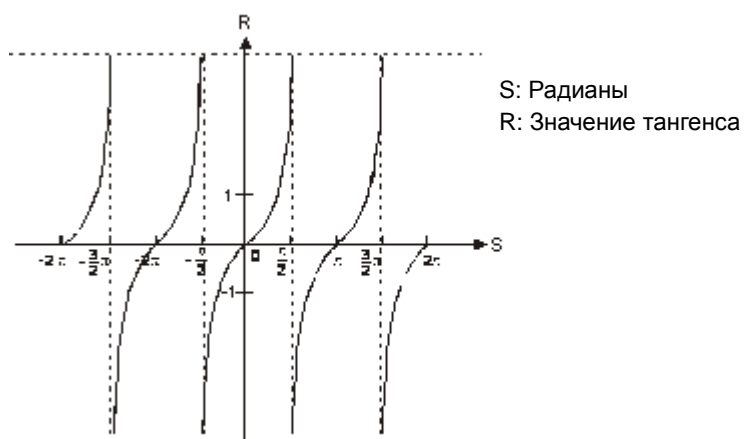
FTAN		FTANP	
En		En	
S	D	S	D

**S** : Исходное значение

**D** : Значение тангенса

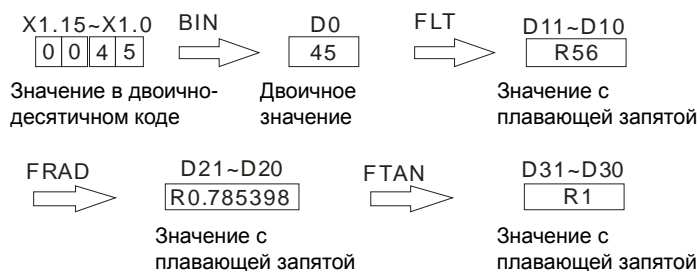
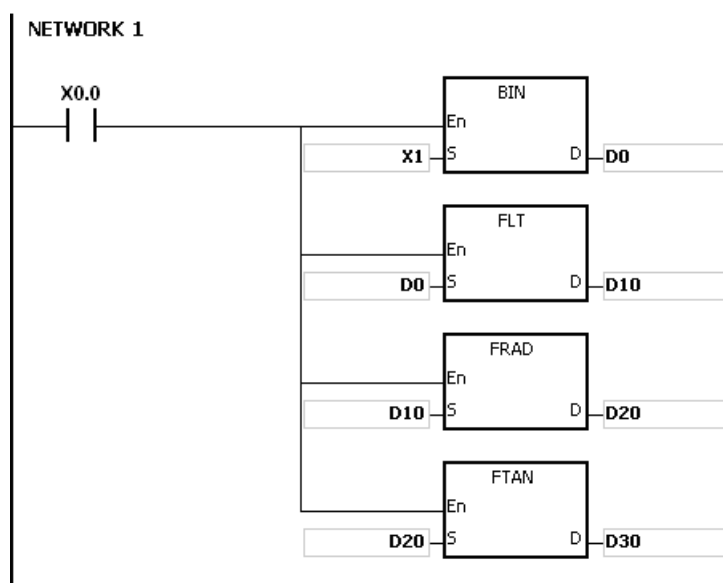
Описание:

1. Реле SM695 определяет исходное значение, задаваемое операндом **S**: Радианы или градусы.
2. Если SM695 = OFF, исходное значение задается операндом **S** в радианах. Радиан = градус × π/180.
3. Если SM695 = ON, исходное значение задается операндом **S** в градусах.  
Градус = радиан × 180/π (0° ≤ градус ≤ 360°)
4. Если результат преобразования равен 0, реле SM600 = ON.
5. Тангенс значения, заданного операндом **S**, записывается в регистр, заданный операндом **D**.
6. Зависимость между значениями тангенса и радианами показана ниже.



**Пример:**

Если вход X0.0 = ON, значение в двоично-десятичном коде в X1.15~X1.0 преобразуется в двоичное значение и результат преобразования записывается в регистр D0. Двоичное значение, записанное в регистре D0, преобразуется в значение с плавающей запятой и записывается в регистр (D11, D10). Значение с плавающей запятой, записанное в регистре (D11, D10), преобразуется в радианы и записывается в регистр (D21, D20). Тангенс значения в радианах, записанного в регистре (D21, D20), записывается в регистр (D31, D30), и тангенс представляет собой значение с плавающей точкой.



**Дополнительные замечания:**

1. Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.
2. Если реле SM695 = ON и значение операнда S меньше 0 или больше 360, инструкция не выполняется, SM0 = ON и записывается код ошибки 16#2003.

API	Код инструкции			Операнд							Функция					
1503		FASIN	P	S, D							Арксинус значения с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

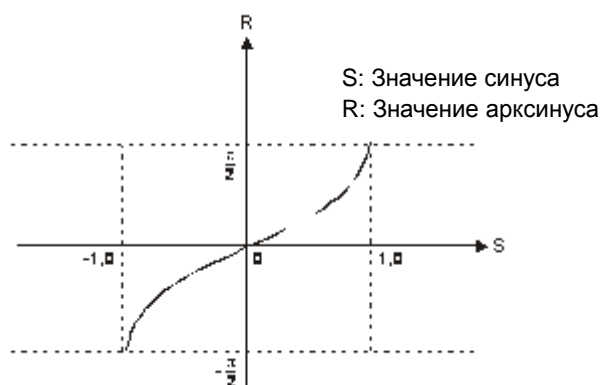
FASIN		FASINP	
En		En	
S	D	S	D

S : Исходное значение  
D : Значение арксинуса

**Описание:**

1. Значение арксинуса  $=\sin^{-1}$

Зависимость между значениями синуса и арксинуса показана ниже.

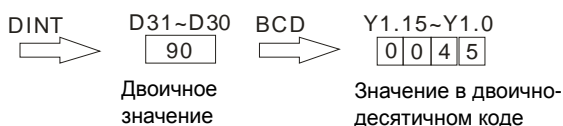
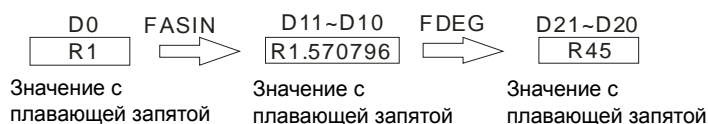
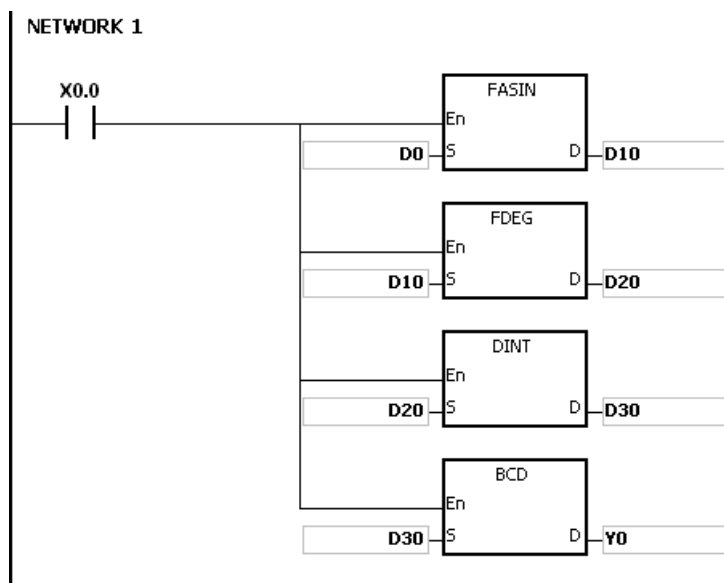


2. Если результат преобразования равен 0, реле SM600 = ON.



**Пример:**

Если вход X0.0 = ON, арксинус значения с плавающей точкой, хранящегося в регистре (D1, D0), записывается в регистр (D11, D10). Значение арксинуса, записанное в регистре (D11, D10), преобразуется в градусы и записывается в регистр (D21, D20). Градусы, записанные в регистре (D11, D10), преобразуется в целое значение и записывается в регистр (D31, D30). Целое, записанное в регистре (D31, D30), преобразуется в значение в двоично-десятичном коде и записывается в Y0.15~Y0.0.



**Дополнительные замечания:**

1. Значение с плавающей запятой, задаваемое операндом **S**, может быть в диапазоне от -1.0 до +1.0. Если значение с плавающей запятой вне данного диапазона, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.
2. Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

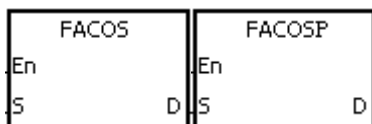
API	Код инструкции			Операнд							Функция					
1504		FACOS	P	S, D							Арккосинус значения с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

Символьное обозначение:

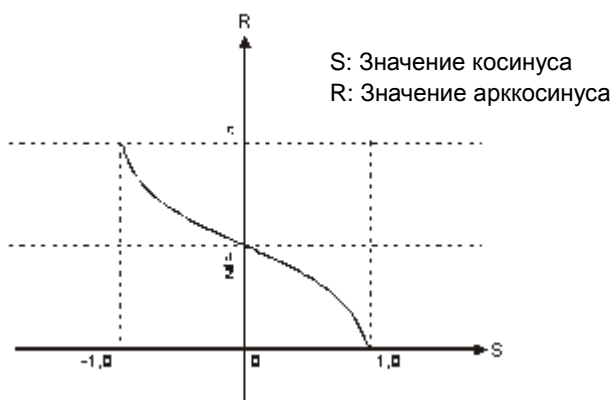


S : Исходное значение  
D : Значение арккосинуса

Описание:

1. Значение арккосинуса  $=\cos^{-1}$

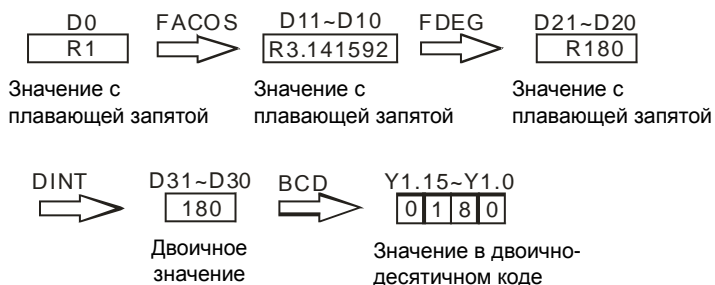
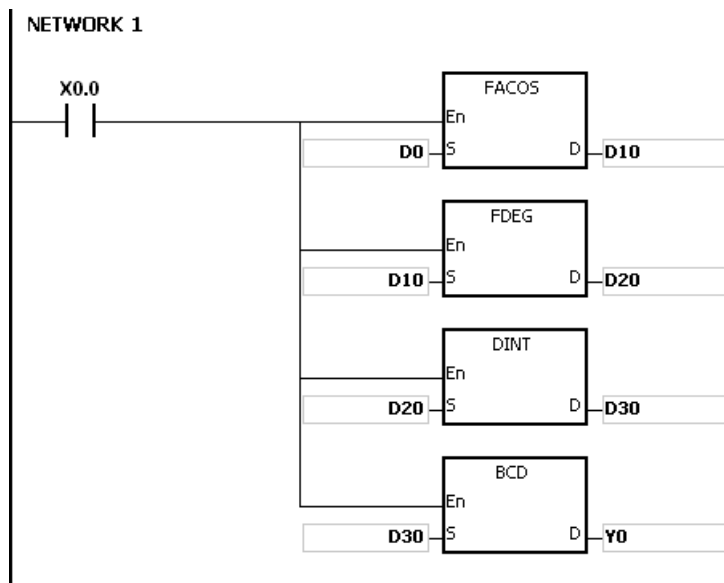
Зависимость между значениями косинуса и арккосинуса показана ниже.



2. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, реле SM602 = ON.
3. Если абсолютное значение результата преобразования меньше значения, которое может быть представлено минимальным значением с плавающей запятой, реле SM601 = ON.
4. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

Если вход X0.0 = ON, арккосинус значения с плавающей точкой, хранящегося в регистре (D1, D0), записывается в регистр (D11, D10). Значение арккосинуса, записанное в регистре (D11, D10), преобразуется в градусы и записывается в регистр (D21, D20). Градусы, записанные в регистре (D11, D10), преобразуется в целое значение и записывается в регистр (D31, D30). Целое, записанное в регистре (D31, D30), преобразуется в значение в двоично-десятичном коде и записывается в Y0.15~Y0.0.



**Дополнительные замечания:**

1. Значение с плавающей запятой, задаваемое операндом **S**, может быть в диапазоне от -1.0 до +1.0. Если значение с плавающей запятой вне данного диапазона, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.
2. Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

API	Код инструкции			Операнд								Функция				
1505		FATAN	P	S, D								Арктангенс значения с плавающей запятой				
Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING			
S									●							
D									●							
								Импульсная инструкция	16-битная инструкция	32-битная инструкция						
								AS	-	AS						

**Символьное обозначение:**

FATAN		FATANP	
En		En	
S	D	S	D

**S** : Исходное значение  
**D** : Значение арктангенса

**Описание:**

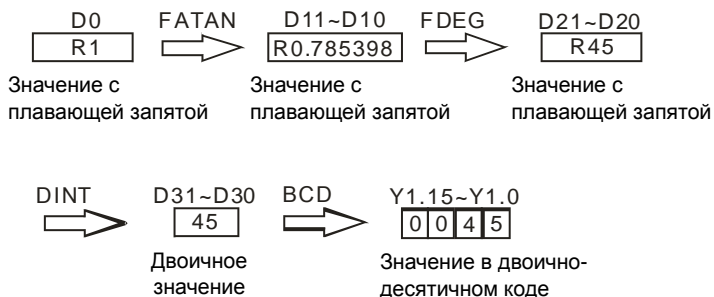
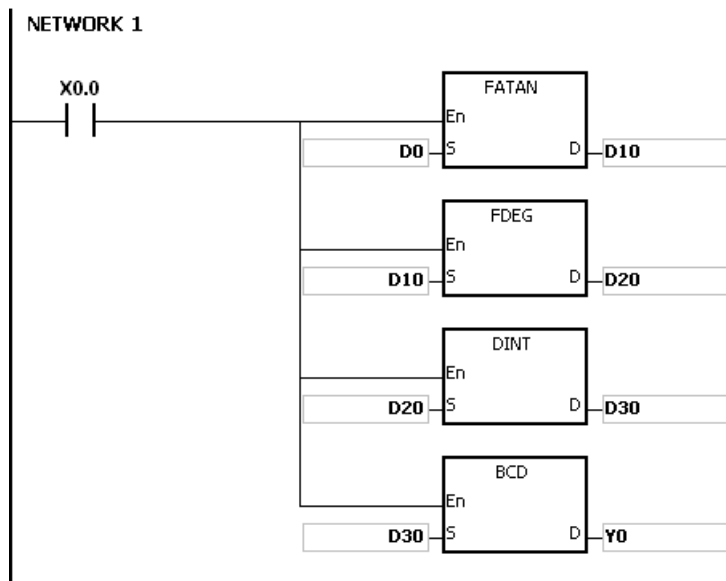
1. Значение арктангенса =  $\tan^{-1}$
2. Зависимость между значениями тангенса и арктангенса показана ниже.



3. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

Если вход X0.0 = ON, арктангенс значения с плавающей точкой, хранящегося в регистре (D1, D0), записывается в регистр (D11, D10). Значение арктангенса, записанное в регистре (D11, D10), преобразуется в градусы и записывается в регистр (D21, D20). Градусы, записанные в регистре (D11, D10), преобразуется в целое значение и записывается в регистр (D31, D30). Целое, записанное в регистре (D31, D30), преобразуется в значение в двоично-десятичном коде и записывается в Y0.15~Y0.0.



**Дополнительные замечания:**

Если значение операнда S выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

API	Код инструкции			Операнд							Функция					
1506		FSINH	P	S, D							Гиперболический синус значения с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

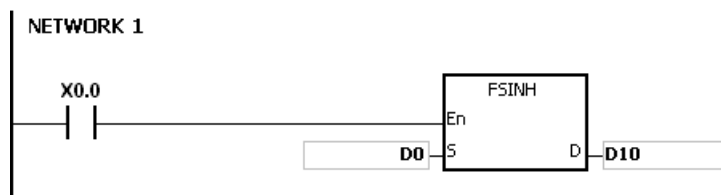


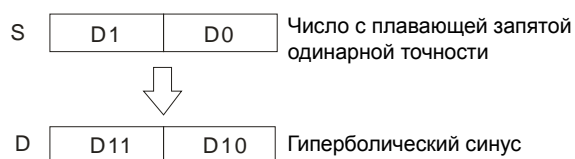
**Описание:**

1. Значение гиперболического синуса =  $(e^s - e^{-s})/2$ .
2. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, значение **D** = 16#7F800000 и SM602 = ON.
3. Если абсолютное значение результата преобразования меньше значения, которое может быть представлено минимальным значением с плавающей запятой, значение **D** = 16#7F800000 и SM601 = ON.
4. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

1. Если вход X0.0 = ON, гиперболический синус значения с плавающей точкой, хранящегося в регистре (D1, D0), записывается в регистр (D11, D10). Гиперболический синус, записанный в регистре (D11, D10), представляет собой значение с плавающей запятой.





2. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, реле SM602 = ON.
3. Если абсолютное значение результата преобразования меньше значения, которое может быть представлено минимальным значением с плавающей запятой, реле SM601 = ON.
4. Если результат преобразования равен 0, реле SM600 = ON.

**Дополнительный результат:**

Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

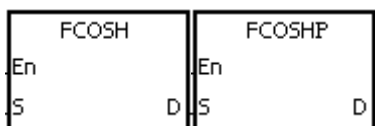
API	Код инструкции			Операнд							Функция					
1507		FCOSH	P	S, D							Гиперболический косинус значения с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



**S** : Исходное значение

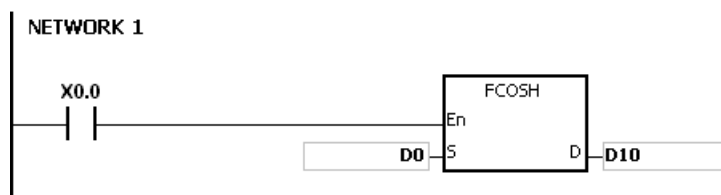
**D** : Значение гиперболического косинуса

**Описание:**

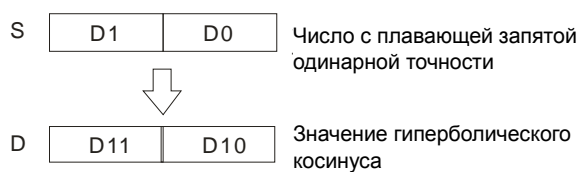
1. Значение гиперболического косинуса =  $(e^S + e^{-S})/2$ .
2. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, значение **D** = 16#7F800000 и SM602 = ON.
3. Если абсолютное значение результата преобразования меньше значения, которое может быть представлено минимальным значением с плавающей запятой, значение **D** = 16#FF800000 и SM601 = ON.
4. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

1. Если вход X0.0 = ON, гиперболический косинус значения с плавающей точкой, хранящегося в регистре (D1, D0), записывается в регистр (D11, D10). Гиперболический косинус, записанный в регистре (D11, D10), представляет собой значение с плавающей запятой.







2. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, реле SM602 = ON.
3. Если абсолютное значение результата преобразования меньше значения, которое может быть представлено минимальным значением с плавающей запятой, реле SM601 = ON.
4. Если результат преобразования равен 0, реле SM600 = ON.

**Дополнительные замечания:**

Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

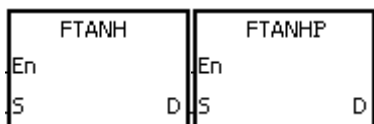
API	Код инструкции			Операнд								Функция				
1508		FTANH	P	S, D								Гиперболический тангенс значения с плавающей запятой				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



**S** : Исходное значение

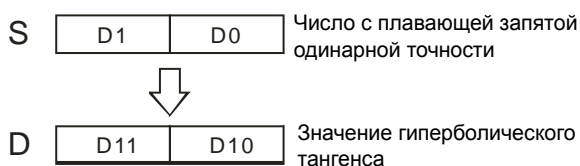
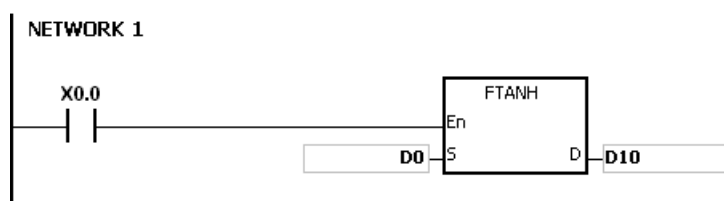
**D** : Значение гиперболического тангенса

**Описание:**

1. Значение гиперболического тангенса =  $(e^s - e^{-s}) / (e^s + e^{-s})$ .
2. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

1. Если вход X0.0 = ON, гиперболический тангенс значения с плавающей точкой, хранящегося в регистре (D1, D0), записывается в регистр (D11, D10). Гиперболический тангенс, записанный в регистре (D11, D10), представляет собой значение с плавающей запятой.



2. Если результат преобразования равен 0, реле SM600 = ON.

**Дополнительные замечания:**

Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

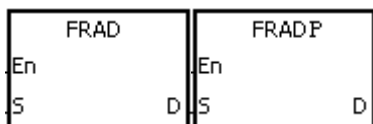
API	Код инструкции			Операнд							Функция					
1509		FRAD	P	S, D							Преобразование градусов в радианы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



**S** : Исходные данные (в градусах)

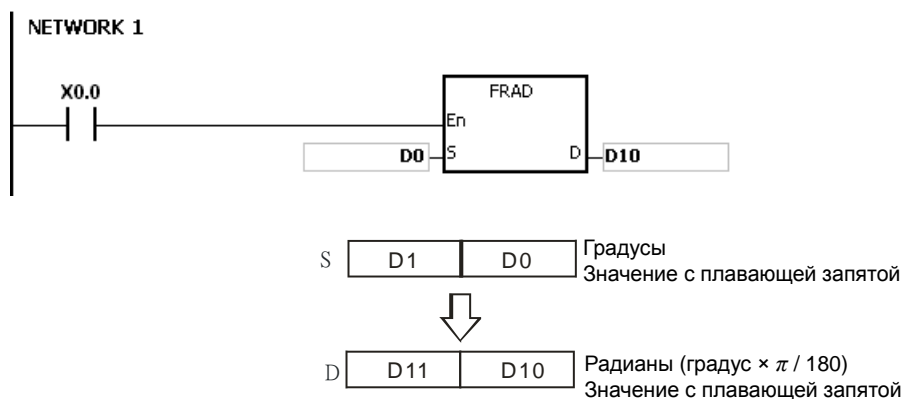
**D** : Результат преобразования (в радианах)

**Описание:**

1. Уравнение, приведенное ниже, предназначено для перевода градусов в радианы.
2. Радиан = градус × (π/180).
3. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

Если вход X0.0 = ON, градусы, записанные в регистре (D1, D0), переводятся в радианы, и результат записывается в регистр (D11, D10). Радианы, записанные в регистре (D11, D10), представляют собой значение с плавающей запятой.



**Дополнительные замечания:**

Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

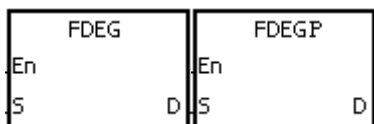
API	Код инструкции			Операнд							Функция					
1510		FDEG	P	S, D							Преобразование радиан в градусы					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



**S** : Исходные данные (в радианах)

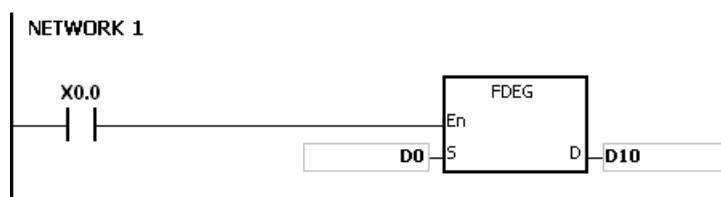
**D** : Результат преобразования (в градусах)

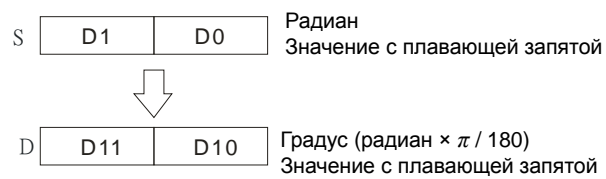
**Описание:**

1. Уравнение, приведенное ниже, предназначено для перевода радиан в градусы.
2.  $\text{Градус} = \text{радиан} \times (180/\pi)$ .
3. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, значение **D** = 16#7F7FFFFF.
4. Если абсолютное значение результата преобразования меньше значения, которое может быть представлено минимальным значением с плавающей запятой, значение **D** = 16#7F7FFFFF.
5. Если результат преобразования равен 0, реле SM600 = ON.

**Пример:**

Если вход X0.0 = ON, радианы, записанные в регистре (D1, D0), переводятся в градусы и результат записывается в регистр (D11, D10). Градусы, записанные в регистре (D11, D10), представляют собой значение с плавающей запятой.





**Дополнительные замечания:**

Если значение операнда **S** выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2013.

API	Код инструкции			Операнд							Функция					
1511	D	SQR	P	S, D							Квадратный корень двоичного числа					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●	●		●	●	●				●	●	
		●	●		●	●	●				●	●	

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**



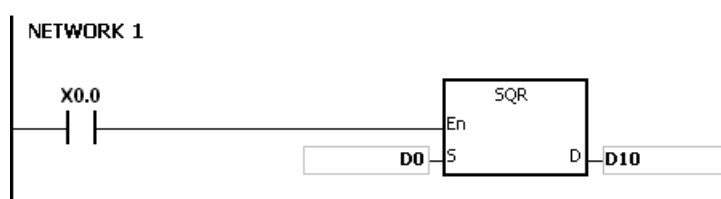
- S** : Исходный регистр
- D** : Регистр, в который записываются данные

**Описание:**

1. Вычисляется квадратный корень значения, заданного операндом **S**, и результат записывается в регистр, заданный операндом **D**.
2. Результат операции записывается в операнд **D** как целое число. Если число с плавающей точкой округляется в меньшую сторону, SM601 = ON.
3. Если результат, записываемый в операнд **D**, равен 0, SM600 = ON.
4. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре **E** – не может.

**Пример:**

Если состояние входа X0.0 = ON, берется квадратный корень значения в регистре D0 и результат записывается в регистр D10.





**Дополнительные замечания:**

1. Значение, задаваемое операндом **S**, может быть только положительным. Если значение, задаваемое операндом **S**, отрицательное, происходит ошибка, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.

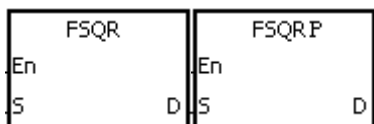
API	Код инструкции			Операнд							Функция					
1512		FSQR	P	S, D							Квадратный корень значения с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



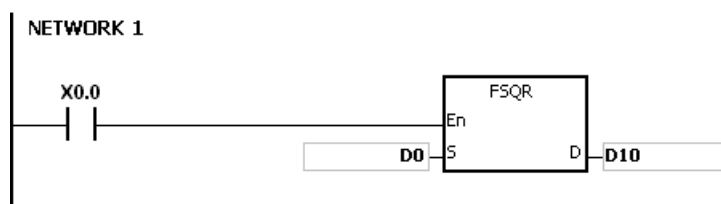
- S** : Исходный регистр
- D** : Регистр, в который записываются данные

**Описание:**

1. Вычисляется квадратный корень значения с плавающей запятой, заданного операндом **S**, и результат записывается в регистр, заданный операндом **D**.
2. Если результат, записываемый в операнд **D**, равен 0, SM600 = ON.

**Пример 1:**

Если вход X0.0 = ON, берется квадратный корень числа с плавающей запятой из регистра (D1, D0) и результат операции записывается в регистр (D11, D10).



**Дополнительные замечания:**

1. Значение, задаваемое операндом **S**, может быть только положительным. Если значение, задаваемое операндом **S**, отрицательное, происходит ошибка, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.

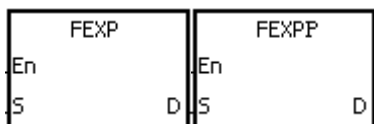
API	Код инструкции			Операнд							Функция					
1513		FEXP	P	S, D							Экспонента от числа с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



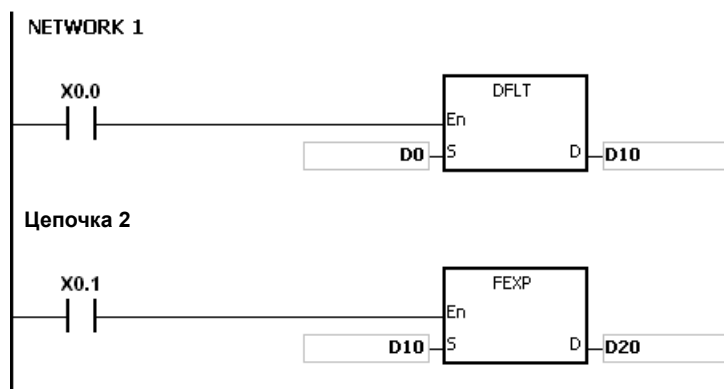
- S** : Исходный регистр
- D** : Регистр, в который записывается результат операции

**Описание:**

1. Возведение в степень представляет собой операцию, в которой есть два числа: основание  $e = 2.71828$  и натуральный показатель, задаваемый операндом **S**.
2.  $EXP[D+1, D]=[S+1, S]$ .
3. Число регистра, задаваемого операндом **S**, может быть положительным или отрицательным. Регистр, задаваемый операндом **D**, может быть 32-битным регистром, а число регистра, задаваемого операндом **S**, может быть с плавающей точкой.
4. Значение регистра, задаваемого операндом **D**, представляет собой  $e^S$  ( $e = 2.71828$  и **S** представляет исходное значение)
5. Если абсолютное значение результата преобразования больше значения, которое может быть представлено максимальным значением с плавающей запятой, значение регистра, задаваемого операндами **D**, =  $16\#7F800000$  и  $SM602 = ON$ .
6. Если результат, записываемый в операнд **D**, равен 0,  $SM600 = ON$ .

**Пример:**

1. Если вход  $X0.0 = ON$ , значение регистра (D1, D0) преобразуется в значение с плавающей запятой и результат записывается в регистр (D11, D10).
2. Если вход  $X0.1 = ON$ , производится возведение в степень, а натуральный показатель берется из регистра (D11, D10). В результате получается число с плавающей запятой, которое записывается в регистр (D21, D20).



API	Код инструкции			Операнд							Функция						
1514		FLOG	P	<b>S<sub>1</sub>, S<sub>2</sub>, D</b>							Логарифм значения с плавающей запятой						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S<sub>1</sub></b>	●	●			●	●	●	●	●		○					○
<b>S<sub>2</sub></b>	●	●			●	●	●	●	●		○					○
<b>D</b>		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>									●				
<b>S<sub>2</sub></b>									●				
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

FLOG		FLOGP	
En		En	
S1	D	S1	D
S2		S2	

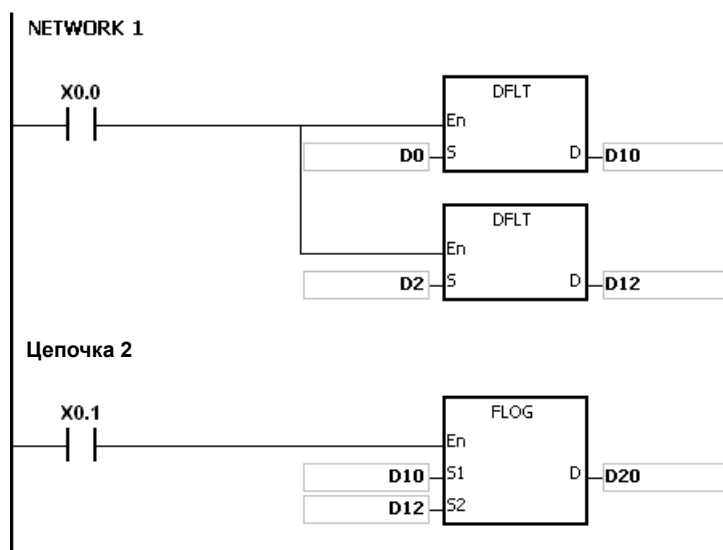
- S<sub>1</sub>** : Регистр, в который записано основание
- S<sub>2</sub>** : Исходное значение
- D** : Регистр, в который записывается результат операции

**Описание:**

1. Берется логарифм значения, задаваемого операндом **S<sub>2</sub>**, по значению, задаваемого операндом **S<sub>1</sub>**, и результат в виде числа с плавающей запятой одинарной точности записывается в операнд **D**.
2. Значения операндов **S<sub>1</sub>** и **S<sub>2</sub>** могут быть только положительными.
3.  $S_1^D = S_2 \rightarrow D = \log_{S_1} S_2$
4. Пример: представим, что операндами **S<sub>1</sub>** и **S<sub>2</sub>** заданы числа 5 и 125 соответственно. Найдем логарифм  $\log_5 125$ .
5.  $S_1^D = S_2 \rightarrow 5^D = 125 \rightarrow D = \log_5 125 = 3$ .
6. Если результат, записываемый в операнд **D**, равен 0, SM600 = ON.

**Пример:**

1. Если вход X0.0 = ON, значения регистров (D1, D0) и (D3, D2) преобразуются в числа с плавающей запятой и результаты записываются в регистры (D11, D10) и (D13, D12) соответственно.
2. Если вход X0.1 = ON, берется логарифм числа с плавающей запятой из регистра (D13, D12) по числу с плавающей запятой из регистра (D11, D10) и результат операции записывается в регистр (D21, D20).



**Дополнительные замечания:**

1. Если значение операнда **S<sub>1</sub>** меньше или равно 1 или если значение операнда **S<sub>2</sub>** меньше или равно 0, инструкция не выполняется, **SM0 = ON** и в регистр **SR0** записывается код ошибки 16#2003.

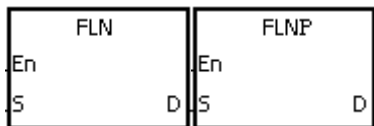
API	Код инструкции			Операнд							Функция					
1515		FLN	P	S, D							Натуральный логарифм двоичного числа с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●	●	●	●		○					○
D		●			●	●	●	●			○					

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S									●				
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**



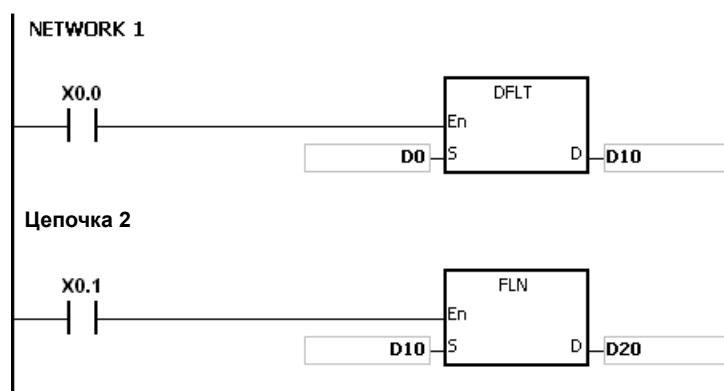
- S** : Исходный регистр
- D** : Регистр, в который записывается результат операции

**Описание:**

1. Берется натуральный логарифм значения, задаваемого операндом **S**, операцией с числом с плавающей точкой одинарной точности.
2. Значение операнда **S** может быть только положительным.
3.  $e^D = S \rightarrow$  значение **D** =  $\ln S$ .
4. Если результат, записываемый в операнд **D**, равен 0, SM600 = ON.

**Пример:**

1. Если вход X0.0 = ON, значение регистра (D1, D0) преобразуется в значение с плавающей запятой и результат записывается в регистр (D11, D10).
2. Если вход X0.1 = ON, берется натуральный логарифм числа с плавающей запятой из регистра (D11, D10) и результат операции записывается в регистр (D21, D20).

**Дополнительные замечания:**

1. Если значение операнда S меньше или равно 0, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.



API	Код инструкции			Операнд							Функция					
1516		FPOW	P	<b>S<sub>1</sub>, S<sub>2</sub>, D</b>							Возведение в степень с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S<sub>1</sub></b>	●	●			●	●	●	●	●							○
<b>S<sub>2</sub></b>	●	●			●	●	●	●	●							○
<b>D</b>		●			●	●	●	●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>									●				
<b>S<sub>2</sub></b>									●				
<b>D</b>									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

FPOW		FPOWP	
En		En	
S1	D	S1	D
S2		S2	

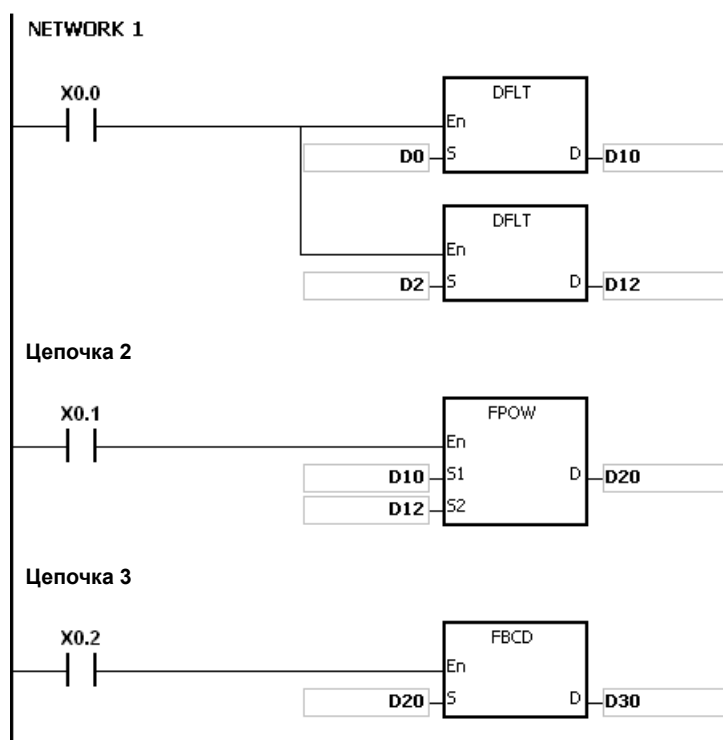
- S<sub>1</sub>** : Регистр, в который записано основание
- S<sub>2</sub>** : Регистр, в который записана степень
- D** : Регистр, в который записывается результат операции

**Описание:**

1. Цифра с плавающей запятой однократной точности, записанная в операнде **S<sub>1</sub>**, возводится в степень, записанную в операнде **S<sub>2</sub>**, и полученная цифра с плавающей запятой однократной точности записывается в операнде **D**.
2.  $D = \text{POW}[S_1+1, S_1]^{[S_2+1, S_2]}$
3. Значение, записываемое в операнде **S<sub>1</sub>**, может быть только положительным, а значение операнда **S<sub>2</sub>** может быть как положительным, так и отрицательным.
4. Представим, что операндами **S<sub>1</sub>** и **S<sub>2</sub>** заданы числа 5 и 3 соответственно.  $D = 5^3 = 125$ .
5. Если абсолютное значение результата операции больше значения, которое может быть представлено максимальным значением с плавающей запятой, значение **D** = 16#7F7FFFFF и SM602 = ON.
6. Если абсолютное значение результата операции меньше значения, которое может быть представлено минимальным значением с плавающей запятой, значение **D** = 16#FF800000 и SM601 = ON.
7. Если результат, записываемый в операнд **D**, равен 0, SM600 = ON.

**Пример:**

1. Если вход X0.0 = ON, значения регистров (D1, D0) и (D3, D2) преобразуются в числа с плавающей запятой и результаты записываются в регистры (D11, D10) и (D13, D12) соответственно.
2. Если вход X0.1 = ON, цифра с плавающей запятой из регистра (D11, D10) возводится в степень с плавающей запятой из регистра (D13, D12) и результат операции записывается в регистр (D21, D20).
3. Если вход X0.2 = ON, двоичное число с плавающей запятой из регистра (D21, D20) преобразуется в число с плавающей запятой в двоично-десятичном коде и результат преобразования записывается в регистр (D31, D30).



**Дополнительные замечания:**

1. Если значение S<sub>1</sub> меньше нуля, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.

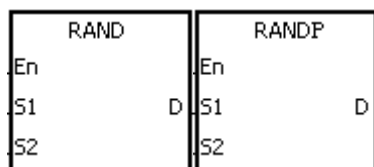
API	Код инструкции			Операнд							Функция					
1517		RAND	P	<b>S<sub>1</sub>, S<sub>2</sub>, D</b>							Генерирование произвольного числа					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	●	●			●	●		●	●		○	○	○	○		
S <sub>2</sub>	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



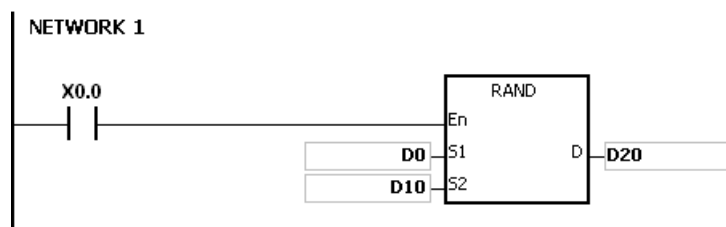
- S<sub>1</sub>** : Минимальное значение
- S<sub>2</sub>** : Максимальное значение
- D** : Регистр, в который записываются данные

**Описание:**

1. Данная инструкция предназначена для генерирования произвольного числа в диапазоне чисел, ограниченном операндами **S<sub>1</sub>** и **S<sub>2</sub>** (минимум и максимум соответственно), и результат записывается в регистре, заданном операндом **D**.
2. Если при выполнении инструкции значение операнда **S<sub>1</sub>** больше значения операнда **S<sub>2</sub>**, значения операндов **S<sub>1</sub>** и **S<sub>2</sub>** берутся как минимум и максимум соответственно.

**Пример:**

Если вход X0.0 = ON, генерируется произвольное число в диапазоне от минимума, заданного в регистре D0, до максимума, заданного регистром D10, и результат записывается в регистр D20.



**Дополнительные замечания:**

Значения операндов **S<sub>1</sub>** и **S<sub>2</sub>** могут быть в диапазоне от 0 до 0~32767. Если значение операнда **S<sub>1</sub>** или **S<sub>2</sub>** вне пределов диапазона, инструкция не выполняется, SM0 = ON и в регистр SR0 записывается код ошибки 16#2003.

## 6.17 Инструкции для часов реального времени

### 6.17.1 Описание инструкций для часов реального времени

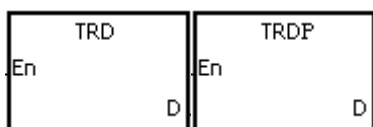
API	Код инструкции			Операнды							Функция			
1600		TRD	P	D							Считывание времени			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D					●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:



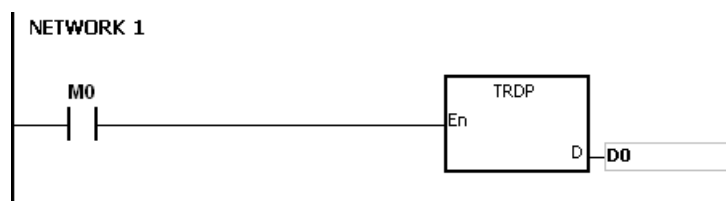
**D** : Операнд для сохранения результата

Описание:

1. **D**: операнд для сохранения считанного текущего времени
2. Операнд **D** занимает семь последовательных регистров.
3. Встроенные часы реального времени в модуле ЦПУ представляют данные, относящиеся к текущим году, дню недели, месяцу, дню, минуте и секунде. Данные сохраняются в SR391 ~ SR397. инструкция TRD используется для чтения текущего времени в семи вышеуказанных регистрах.
4. В регистре SR391 сохраняются две последние цифры значения года.

Пример:

Когда M0 включен, текущее время считывается из часов реального времени в D0~D6. Значение 1 в SR397 представляет Понедельник, значение 2 представляет Вторник и т.д. до значения 7 (Воскресенье).



Специальные регистры данных	Пункт	Значение		Основные регистры данных	Пункт
SR391	Год	00~99	→	D0	Год
SR392	Месяц	1~12	→	D1	Месяц
SR393	День	1~31	→	D2	День
SR394	Час	0~23	→	D3	Час
SR395	Минута	0~59	→	D4	Минута
SR396	Секунда	0~59	→	D5	Секунда
SR397	День недели	1~7	→	D6	День недели

**Примечание:**

1. Если D+6 выходит за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.
2. Когда SM220 включен, часы реального времени округляют показания до ±30 секунд. Если значение находится в диапазоне от 0 до 29, значение минут не изменяется. Если значение находится в диапазоне от 30 до 59, значение минут прибавляется на 1, значения секунд в обоих случаях сбрасывается на 0.

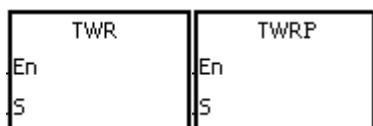
API	Код инструкции			Операнды							Функция						
1601		TWR	P	S							Запись времени						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S					●	●		●	●							

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



**S** : Источник данных

**Описание:**

- S**: Операнд, в который записано значение настройки
- Операнд **S** занимает семь последовательных регистров.
- При необходимости настройки встроенных в ЦПУ часов реального времени, эту инструкцию можно использовать для записи правильного текущего времени во встроенные часы реального времени.
- Когда инструкция выполняется, новое время настройки мгновенно записывается в часы реального времени в ПЛК. Поэтому, когда инструкция выполняется, пользователи должны убедиться, что настройка нового времени соответствует реальному времени.
- Предлагается использовать импульсные инструкции. Если контакт Н/О, инструкция постоянно записывает время. Но ПЛК записывает время только при первом сканировании. Если встроенные часы реального времени необходимо обновить, вы можете закрыть контакт до сканирования, а затем выполнить эту инструкцию еще раз, чтобы обновить часы.

**Пример:**

Когда M0 включен, скорректированное время записывается в часы реального времени, встроенные в ПЛК.



Новые настройки времени	Специальные регистры данных	Пункт	Значение	→	Основные регистры данных	Пункт	Часы реального времени
	SR391	Год	00~99	→	D0	Год	
	SR392	Месяц	1~12	→	D1	Месяц	
	SR393	День	1~31	→	D2	День	
	SR394	Час	0~23	→	D3	Час	
	SR395	Минута	0~59	→	D4	Минута	
	SR396	Секунда	0~59	→	D5	Секунда	
	SR397	День недели	1~7	→	D6	День недели	

**Примечание:**

1. Если значение в **S** выходит за допустимый диапазон, инструкция не выполняется, включается флаг SM, код ошибки 16#2003 записывается в регистр SR.
2. Если **D+6** выходит за допустимый диапазон, инструкция не выполняется, включается флаг SM, код ошибки 16#2003 записывается в регистр SR.
3. Если операнд **S** задается в ПО ISPSOft, тип данных устанавливается как ARRAY [7] для WORD/INT.

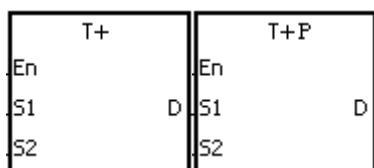
API	Код инструкции			Операнды							Функция						
1602		T+	P	$S_1 \cdot S_2 \cdot D$							Сложение времени						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$					●	●		●	●		○					
$S_2$					●	●		●	●		○					
D					●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:



$S_1$  : Источник 1

$S_2$  : Источник 2

D : Операнд сохранения результата

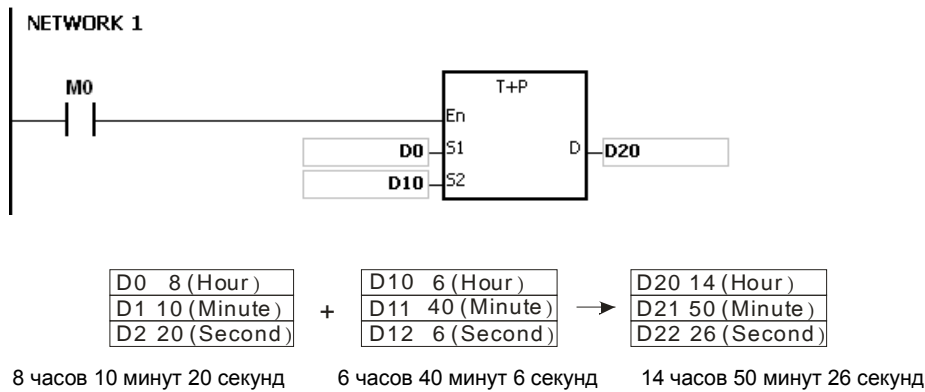
Описание:

1. Значения часов, минут и секунд в часах реального времени, заданных в  $S_2$ , добавляются к значениям часов, минут и секунд в часах реального времени, заданных в  $S_1$ , сумма сохраняется в регистре, указанном в D.
2. Операнды  $S_1$ ,  $S_2$  и D занимают три последовательных регистра.
3. Если сумма больше или равна 24 часам, включается флаг SM602 и в операнде D сохраняется значение часа, превышающее 24 часа.
4. Если сумма равна 0 (0 часов 0 минут 0 секунд), включается флаг SM600.

Пример:

Когда M0 включен, инструкция T+ выполняется. Значения часов, минут и секунд, заданных в D10~D12 складываются с значениями часов, минут и секунд, заданных в D0~D2, результат сохраняется в D20~D22.





**Примечание:**

1. Если значения в **S<sub>1</sub>** или **S<sub>2</sub>** выходят за допустимый диапазон, инструкция не выполняется, включается флаг **SM0**, код ошибки 16#2003 записывается в регистр **SR0**.
2. Если **S<sub>1</sub>+2**, **S<sub>2</sub>+2** или **D+2** выходят за допустимый диапазон, инструкция не выполняется, включается флаг **SM0**, код ошибки 16#2003 записывается в регистр **SR0**.
3. Если операнд **S<sub>1</sub>** задается в ПО ISPSOft, тип данных должен быть ARRAY [3] для WORD/INT.
4. Если операнд **S<sub>2</sub>** задается в ПО ISPSOft, тип данных должен быть ARRAY [3] для WORD/IN.
5. Если операнд **D** задается в ПО ISPSOft, тип данных должен быть ARRAY [3] для WORD/INT.

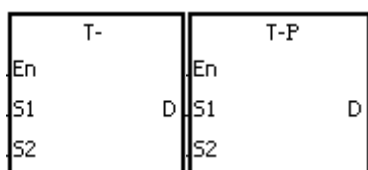
API	Код инструкции				Операнды							Функция					
1603		T-		P	$S_1 \cdot S_2 \cdot D$							Вычитание времени					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$					●	●		●	●		○					
$S_2$					●	●		●	●		○					
<b>D</b>					●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



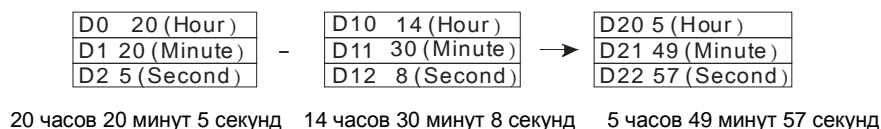
- $S_1$  : Источник 1
- $S_2$  : Источник 2
- D** : Операнд сохранения результата

**Описание:**

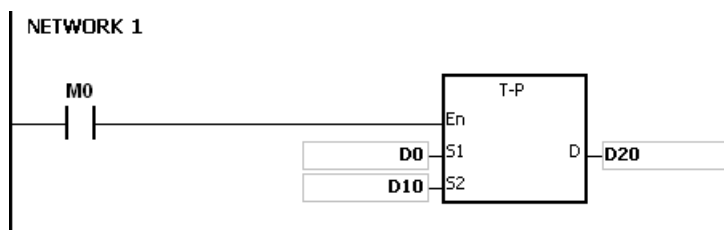
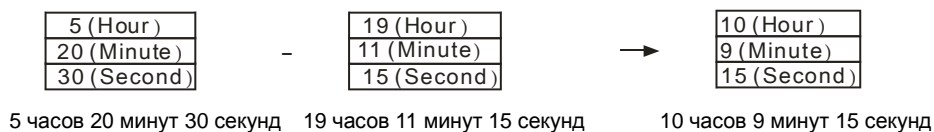
- Из значений часов, минут и секунд в часах реального времени, заданных в  $S_2$ , вычитаются значения часов, минут и секунд в часах реального времени, заданных в  $S_1$ , разность сохраняется в регистре, указанном в **D**.
- Операнды  $S_1$ ,  $S_2$  и **D** занимают три последовательных регистра.
- Если разность отрицательна, включается флаг SM601 и в операнде **D** сохраняется значение часа, превышающее 24 часа.
- Если разность равна 0 (0 часов 0 минут 0 секунд), включается флаг SM600.

**Пример:**

- Когда M0 включен, инструкция T- выполняется. Из значений часов, минут и секунд, заданных в D10~D12 вычитаются значения часов, минут и секунд, заданных в D0~D2, результат сохраняется в D20~D22.



2. Если разность отрицательна, включается флаг SM601.



**Примечание:**

1. Если значения в **S<sub>1</sub>** или **S<sub>2</sub>** выходят за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.
2. Если **S<sub>1</sub>+2**, **S<sub>2</sub>+2** или **D+2** выходят за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.
3. Если операнд **S<sub>1</sub>** задается в ПО ISPSOft, тип данных должен быть ARRAY [3] для WORD/INT.
4. Если операнд **S<sub>2</sub>** задается в ПО ISPSOft, тип данных должен быть ARRAY [3] для WORD/INT.
5. Если операнд **D** задается в ПО ISPSOft, тип данных должен быть ARRAY [3] для WORD/INT.

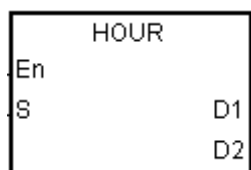
API	Код инструкции			Операнды							Функция												
1604		HOUR															Счетчик времени работы						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S					●	●		●	●		○	○	○	○		
D <sub>1</sub>								●								
D <sub>2</sub>		●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D <sub>1</sub>		●			●	●							
D <sub>2</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	

**Символ:**



- S** : Заданное значение времени до включения выхода
- D<sub>1</sub>** : Текущее время
- D<sub>2</sub>** : Выход

**Описание:**

1. **S**: Заданное значение времени до включения выхода (Ед. изм.: час)
  - D<sub>1</sub>**: Текущее время (Ед. изм.: час)
  - D<sub>2</sub>**: Выход
2. Операнд **S** используется 16-битной инструкцией и его значение находится в диапазоне 1...32 767.
3. Инструкция HOUR:
  - D<sub>1</sub>**: Текущее время (Ед. изм.: час)
    - Значение в **D<sub>1</sub>** находится в диапазоне 0...32 767.
  - D<sub>1+1</sub>**: Текущее время, меньше одного часа (Ед. изм.: секунда)
    - Значение в **D<sub>1+1</sub>** находится в диапазоне 0...3 599.
  - D<sub>1+2</sub>** – системный операнд. Значение в нем не может быть изменено при выполнении инструкции. В противном случае отобразится ошибка.

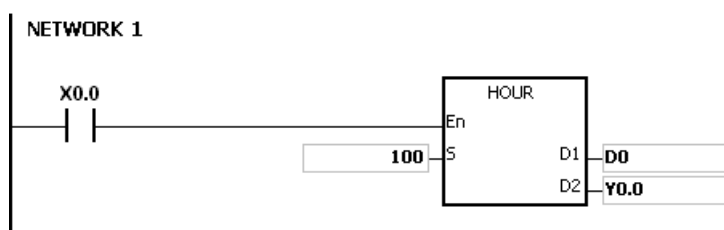
Когда значение текущего времени достигает 32 767 часа 3 599 секунд, подсчет времени останавливается.

После сброса на 0 значений в  $D_1$  и  $D_1+1$  подсчет времени возобновится.

4. Когда время, в течение которого входной контакт был включен, достигает заданного времени, выход включается. Пока время, в течение которого входной контакт был включен, не достигает заданного времени, выход не включен. Эта функция позволяет управлять временем работы и обслуживанием оборудования.
5. После включения выхода таймер продолжает подсчет.
6. Когда используется онлайн-редактирование, пожалуйста, сбросьте заданный контакт, чтобы инициализировать инструкцию.

### Пример 1:

16-битная инструкция HOUR: Когда X0.0 включен, начинается подсчет. Когда время, в течение которого X0.0 был включен, достигает 100 часов, включается Y0.0. Текущее время сохраняется в D0, текущее время, которое составляет менее одного часа, записывается в D1. D2 – системный операнд, значение в нем не может быть изменено, иначе отображается ошибка.



### Примечание:

1. Когда значение  $S$  меньше или равно 0, инструкция не выполняется, состояние выхода не изменяется.
2. Если значение в  $D_1$ , используемого в инструкции HOUR, меньше 0, состояние выхода не изменяется.
3. Если  $D_1+2$ , используемого в инструкции HOUR выходят за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.
4. Если операнд  $D_1$  используется в 16-битной инструкции и задается в ПО ISPSOft, тип данных устанавливается как ARRAY [3] для WORD/INT.
5. Если операнд  $D_1$  используется в 32-битной инструкции и задается в ПО ISPSOft, тип данных устанавливается как ARRAY [2] для DWORD/DINT.

API	Код инструкции			Операнды								Функция					
1605		TCMP	P	$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot D$								Сравнение времени					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●		○	○	○	○		
$S_2$	●	●			●	●		●	●		○	○	○	○		
$S_3$	●	●			●	●		●	●		○	○	○	○		
$S$					●	●		●								
$D$		●	●	●				●		○						

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							
$S$		●			●	●							
$D$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**

TCMP		TCMPP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	
S		S	

- $S_1$  : Часы заданного времени
- $S_2$  : Минуты заданного времени
- $S_3$  : Секунды заданного времени
- $S$  : Текущее время
- $D$  : Результат сравнения

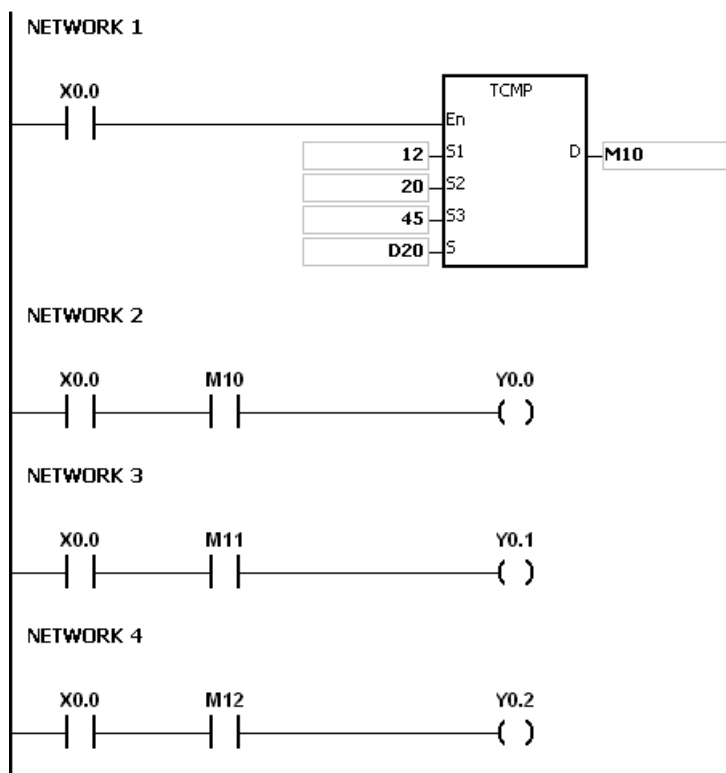
**Описание:**

1. Значения часов, минут и секунд в  $S_1 \sim S_3$  сравниваются со значениями часов, минут и секунд в  $S$ , результат сохраняется в  $D$ .
2. Часы текущего времени находится в  $S$ , их значение должно находиться в диапазоне от 0 до 23. Минуты текущего времени находится в  $S+1$ , их значение должно находиться в диапазоне от 0 до 59. Секунды текущего времени находится в  $S+2$ , их значение должно находиться в диапазоне от 0 до 59.
3. Операнд  $D$  занимает три последовательных регистра. Результат сравнения сохраняется в  $D$ ,  $D+1$  и  $D+2$ .
4. Обычно используется инструкция TRD для считывания текущего времени с часов реального времени, а затем инструкция TCMP для сравнения времени.

5. Если задание времени в **S<sub>1</sub>~S<sub>3</sub>** выше значения текущего времени в **S**, **D** включен, **D+1** отключен, **D+2** отключен.
6. Если задание времени в **S<sub>1</sub>~S<sub>3</sub>** равно значению текущего времени в **S**, **D** отключен, **D+1** включен, **D+2** отключен.
7. Если задание времени в **S<sub>1</sub>~S<sub>3</sub>** ниже значения текущего времени в **S**, **D** отключен, **D+1** отключен, **D+2** включен.

**Пример:**

1. Когда X0.0 включен, инструкция начинает выполнение. Заданное время 12 часов 20 минут 45 секунд сравнивается с текущим временем в D20 ~ D22, а результат сравнения сохраняется в M10 ~ M12. Когда X0.0 отключается, инструкция не выполняется. Кроме того, состояния M10, M11 и M12 остаются такими же, как и до включения X0.0.
2. Если необходимо получить результат сравнения  $\geq$ ,  $\leq$  или  $\neq$ , M10~M12 могут подключаться последовательно или параллельно.



**Примечание:**

1. Если значение в **S+2** выходит за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.

2. Если значение в **D+2** выходит за допустимый диапазон, инструкция не выполняется, включается флаг **SM0**, код ошибки 16#2003 записывается в регистр **SR0**.
3. Если значение в **S** выходит за допустимый диапазон, инструкция не выполняется, включается флаг **SM0**, код ошибки 16#2003 записывается в регистр **SR0**.
4. Если значения в **S<sub>1</sub>~S<sub>3</sub>** выходят за допустимый диапазон, инструкция не выполняется, включается флаг **SM0**, код ошибки 16#2003 записывается в регистр **SR0**.
5. Если операнд **S** задается в ПО ISPSOft, тип данных определяется как **ARRAY [3]** для **WORD**.
6. Если операнд **D** задается в ПО ISPSOft, тип данных определяется как **ARRAY [3]** для **BOOL**.



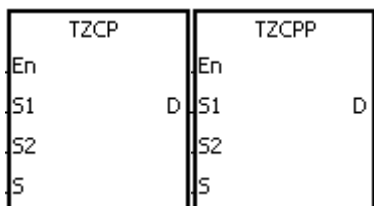
API	Код инструкции			Операнды							Функция					
1606		TZCP	P	$S_1 \cdot S_2 \cdot S \cdot D$							Сравнение временных зон					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$					●	●		●	●							
$S_2$					●	●		●	●							
$S$					●	●		●								
$D$		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S$		●			●	●							
$D$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



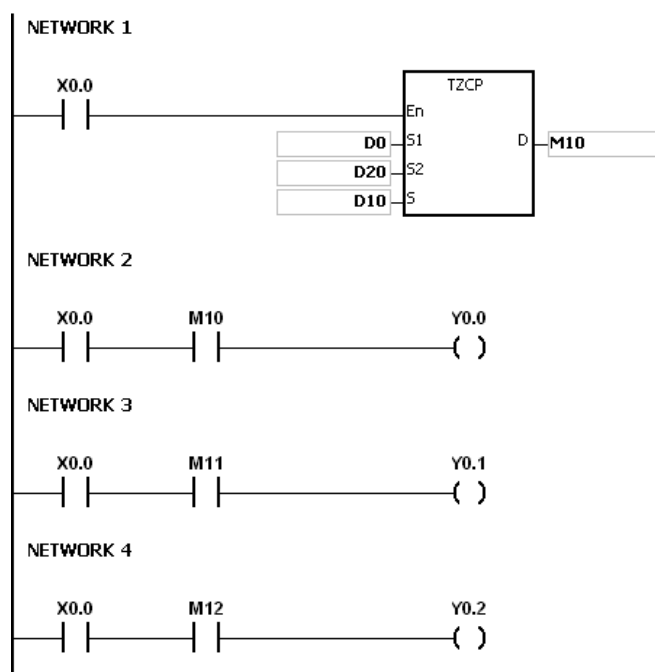
- $S_1$  : Нижний предел времени
- $S_2$  : Верхний предел времени
- $S$  : Текущее время
- $D$  : Результат сравнения

**Описание:**

1. Инструкция сравнивает текущее время в  $S$  с нижним пределом времени, заданном в  $S_1$  и с верхним пределом времени, заданном в  $S_2$ , результат сравнения сохраняется в  $D$ .
2. Часы, минуты и секунды для нижнего предела задаются соответственно в  $S_1$ ,  $S_1+1$  и  $S_1+2$ .
3. Часы, минуты и секунды для верхнего предела задаются соответственно в  $S_2$ ,  $S_2+1$  и  $S_2+2$ .
4. Часы, минуты и секунды текущего времени задаются соответственно в  $S$ ,  $S+1$  и  $S+2$ .
5. Время в  $S_1$  должно быть меньше времени в  $S_2$ . Если значение в  $S_1$  больше, чем значение в  $S_2$ , значение в  $S_1$  при выполнении инструкции TZCP будет считаться верхним пределом, а в  $S_2$  – нижним.
6. Обычно используется инструкция TRD для считывания текущего времени с часов реального времени, а затем инструкция TZCP для сравнения времени.
7. Если текущее значение времени в  $S$  меньше нижнего предела в  $S_1$ , включается  $D$ . Если текущее значение времени в  $S$  больше верхнего предела в  $S_2$ , включается  $D+2$ . В других случаях, включается  $D+1$ .

**Пример:**

Когда X0.0 включен, инструкция TZCP начинает выполнение. Включаются M10, M11 или M12. При отключении X0.0 инструкция TZCP не выполняется, состояние M10, M11 и M12 остаются такими же, как до включения X0.0.

**Примечание:**

1. Если значения в  $S_1+2$ ,  $S_2+2$ ,  $S+2$  или  $D+2$  выходят за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.
2. Если значения в  $S_1$ ,  $S_2$ , and  $S$  выходят за допустимый диапазон, инструкция не выполняется, включается флаг SM0, код ошибки 16#2003 записывается в регистр SR0.
3. Если операнд  $S_1$  задается в ПО ISPSOft, тип данных определяется как ARRAY [3] для WORD/INT.
4. Если операнд  $S_2$  задается в ПО ISPSOft, тип данных определяется как ARRAY [3] для WORD/INT.
5. Если операнд  $S$  задается в ПО ISPSOft, тип данных определяется как ARRAY [3] для WORD/INT.
6. Если операнд  $D$  задается в ПО ISPSOft, тип данных определяется как ARRAY [3] для BOOL.

API	Код инструкции			Операнды							Функция					
1607		DST	P	<b>S · S<sub>1</sub> · S<sub>2</sub> · S<sub>3</sub> · S<sub>4</sub> · S<sub>5</sub> · D</b>							Летнее время					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S</b>					●	●		●					○	○		
<b>S<sub>1</sub></b>					●	●		●					○	○		
<b>S<sub>2</sub></b>					●	●		●					○	○		
<b>S<sub>3</sub></b>					●	●		●					○	○		
<b>S<sub>4</sub></b>					●	●		●					○	○		
<b>S<sub>5</sub></b>					●	●		●					○	○		
<b>D</b>		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●			●	●							
<b>S<sub>1</sub></b>		●			●	●							
<b>S<sub>2</sub></b>		●			●	●							
<b>S<sub>3</sub></b>		●			●	●							
<b>S<sub>4</sub></b>		●			●	●							
<b>S<sub>5</sub></b>		●			●	●							
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

6

**Символ:**

DST		DSTP	
En		En	
S	D	S	D
S1		S1	
S2		S2	
S3		S3	
S4		S4	
S5		S5	

- S** : Функциональные коды для летнего времени
- S<sub>1</sub>** : Начальный месяц летнего времени
- S<sub>2</sub>** : Начальная дата летнего времени
- S<sub>3</sub>** : Месяц окончания летнего времени
- S<sub>4</sub>** : Дата окончания летнего времени
- S<sub>5</sub>** : Длительность прибавления летнего времени (минуты)
- D** : Функция состояния летнего времени

**Описание:**1. **S**: Функциональные коды летнего времени

Версии прошивки до версии V1.04 (исключая V1.04) поддерживают следующие функциональные коды:

Функциональный код	Описание
0	Отключение летнего времени
1	Режим летнего времени 1
2	Считывание летнего времени
3~	Зарезервированы или также чтение летнего времени

Версии прошивки выше версии V1.04 (включая V1.04) поддерживают следующие функциональные коды:

Функциональный код	Описание
0	Отключение летнего времени
1	Режим летнего времени 1
2	Считывание летнего времени
3	Режим летнего времени 2
4	Системное отключение летнего времени
5	Системная установка режима летнего времени 1
6	Системная установка режима летнего времени 2
7~	Зарезервированы или также чтение летнего времени

Примечание 1: Если **S** равен 4, 5 или 7, выполнение данной инструкции невозможно.

Примечание 2: Подробнее о различных режимах см. в следующих разделах.

**S**<sub>1</sub>: установка месяца перехода на летнее время

**S**<sub>2</sub>: установка даты перехода на летнее время

**S** = 1 (включен режим летнего времени 1), **S**<sub>2</sub>: настройки даты перехода на летнее время

**S** = 3 (включен режим летнего времени 2), **S**<sub>2</sub>: настройки недели перехода на летнее время, **S**<sub>2</sub> + 1: день недели  
**S**<sub>2</sub>

**S**<sub>3</sub>: установка месяца окончания летнего времени

**S**<sub>4</sub>: настройки даты окончания летнего времени

**S** = 1 (включен режим летнего времени 1), **S**<sub>4</sub>: настройки даты окончания действия летнего времени

**S** = 3 (включен режим летнего времени 2), **S**<sub>4</sub>: настройки недели окончания летнего времени, **S**<sub>2</sub> + 1: день недели  
для **S**<sub>4</sub>

**S**<sub>5</sub>: настройка изменения значения для летнего времени; единица: минута

**D**: сохраняет состояние летнего времени; когда значение в **D** выключено, переход на летнее время отключен.  
Когда значение в **D** включено, летнее время включено.

2. Описания значений функционального кода **S** для функции летнего времени:

Летнее время	Функциональный код	Описание
Отключено	0	Отключение летнего времени
Включено	1,3	Включение летнего времени
Считывается	2	Считывание летнего времени

- **Функция отключенного летнего времени (см. Пример 1 ниже)**

Когда операнд **S** равен 0, функция летнего времени отключается. Когда **S** устанавливается с целью отключения летнего времени, значения **S<sub>1</sub>-S<sub>5</sub>** не имеют смысла, а операнд **D** показывает отключенное летнее время.

- **Включенная функция летнего времени (см. Примеры 2 и 3)**

Когда значение **S** равно 1 или 3, функция летнего времени включена: **S<sub>1</sub>** и **S<sub>2</sub>**: установка месяца и даты перехода на летнее время; **S<sub>3</sub>** и **S<sub>4</sub>**: установка месяца и даты отключения летнего времени; **S<sub>5</sub>**: настройка изменения значения для летнего времени; единица: минута; операнд **D** сохраняет состояние летнего времени. Когда функция летнего времени включена и система запускается после этого в первый раз, системное время однократно добавит значение, заданное в **S<sub>5</sub>**. Когда функция летнего времени отключена, и система запускается после этого в первый раз, системное время однократно добавит значение, заданное в **S<sub>5</sub>**.

**Режимы летнего времени:**

Режим	Функциональный код	Описание	Применяемость
Режим 1	0	Месяц и дата	Прошивка версии V1.00 и ниже
Режим 2	3	Неделя и день недели	Прошивка версии V1.04 и выше

**Режим 1 (S=1): задание месяцем и датой (см. Пример 2)**

Операнд	Описание
<b>S<sub>1</sub></b>	Месяц включения летнего времени Диапазон: 1-12
<b>S<sub>2</sub></b>	Дата включения летнего времени Диапазон: 1-31
<b>S<sub>3</sub></b>	Месяц отключения летнего времени Диапазон: 1-12
<b>S<sub>4</sub></b>	Дата отключения летнего времени Диапазон: 1-31
<b>S<sub>5</sub></b>	Настройка изменения значения для летнего времени, ед.изм.: минуты Диапазон: 1-1439 (в течение 1 дня)

Примечание 1: Если эта функция включена, значение в **D** также включено.

Примечание 2: Если дата установлена некорректно, функция летнего времени не может быть включена. При этом включается флаг SM0, а в SR0 записывается код ошибки 16#200B. Например, если установлена не существующая дата, например 31 апреля, или дата начала меньше, чем конечная дата в текущем календарном году, например, начальная дата – 01 октября, а дата окончания – 01 апреля.

Примечание 3: Если значение **S<sub>5</sub>** установлено вне допустимого диапазона, функция летнего времени не может быть включена. При этом включается флаг SM0, а в SR0 записывается код ошибки 16#200B.

**Режим 2 (S=3): задание недель и днем недели (см. Пример 3)**

Операнд	Описание
<b>S<sub>1</sub></b>	Месяц включения летнего времени Диапазон: 1-12
<b>S<sub>2</sub></b> <b>S<sub>2</sub> + 1</b>	Неделя включения летнего времени, диапазон: 1-4 День недели для <b>S<sub>2</sub></b> , диапазон: 1-7 (Пн:1...Вс:7)
<b>S<sub>3</sub></b>	Месяц отключения летнего времени Диапазон: 1-12
<b>S<sub>4</sub></b> <b>S<sub>4</sub> + 1</b>	Неделя отключения летнего времени, диапазон: 1-4 День недели для <b>S<sub>2</sub></b> , диапазон: 1-7 (Пн:1...Вс:7)
<b>S<sub>5</sub></b>	Настройка изменения значения для летнего времени, ед.изм.: минуты Диапазон: 1-1439 (в течение 1 дня)

Примечание 1: Если эта функция включена, значение в **D** также включено.

Примечание 2: Диапазон настройки для **S<sub>2</sub>** и **S<sub>4</sub>** составляет 1...4 или -1... -4. Значение -1 указывает на последнюю неделю месяца, а -2 указывает на предпоследнюю неделю месяца. Если значение в **S<sub>2</sub>** равно -2 и **S<sub>2</sub> + 1** равно 7, это указывает на последнее 2 воскресенье месяца. Если дата установлена неверно, функция летнего времени не может быть включена. При этом включается флаг SM0, а в SR0 записывается код ошибки 16#200B.

Примечание 3: Если значение в **S<sub>2</sub> + 1** / **S<sub>4</sub> + 1** выходит за допустимые пределы, значение по умолчанию равно 7, что означает воскресенье.

Примечание 4: Если значение **S<sub>5</sub>** установлено вне допустимого диапазона, функция летнего времени не может быть включена. При этом включается флаг SM0, а в SR0 записывается код ошибки 16#200B.

Примечание 5: Если значение операндов **S<sub>2</sub>** и **S<sub>4</sub>** равно K или 16#, значения не сохраняются. При этом включается флаг SM0, а в SR0 записывается код ошибки 16#2003.

- **Считывание функции летнего времени (см. Пример 1-3)**

Когда значение операнда **S** равно 2, функция летнего времени считывается. **S<sub>1</sub>** и **S<sub>2</sub>**: установка месяца и даты перехода на летнее время; **S<sub>3</sub>** и **S<sub>4</sub>**: установка месяца и даты отключения летнего времени; **S<sub>5</sub>**: настройка изменения значения для летнего времени; единица: минута. Когда **S** настроено на считывание параметров функции летнего времени, и **D** включен, ПЛК сохраняет значения настроек в операндах **S<sub>1</sub>-S<sub>5</sub>**. Если значение операндов **S<sub>2</sub>** и **S<sub>4</sub>** равно K или 16#, значения не сохраняются. При этом включается флаг SM0, а в SR0 записывается код ошибки 16#2003.

Устройства с версией прошивки выше V1.04 (включая V1.04) добавляют 4 к функциональным кодам **S** после того, как будут прочитаны параметры летнего времени. Например, после считывания параметров летнего времени функциональные коды 0, 1, 3 изменятся на 4, 5 и 7.

Когда функция летнего времени отключена, операнды имеют следующие значения:

Операнд	Описание
<b>S</b>	Прошивка до V1.04 (не включая V1.04): фиксированный функциональный код 2 Прошивка выше V1.04 (включая V1.04): функциональный код 4, индикация: функция летнего времени отключена
<b>S<sub>1</sub> - S<sub>5</sub></b>	Операнды не существуют
<b>D</b>	Состояние функции летнего времени: отключена

Когда функция летнего времени включена в режиме 1, операнды имеют следующие значения:

Операнд	Описание
<b>S</b>	Прошивка до V1.04 (не включая V1.04): фиксированный функциональный код 2 Прошивка выше V1.04 (включая V1.04): функциональный код 5, индикация: функция летнего времени включена в режиме 1
<b>S<sub>1</sub></b>	Месяц включения летнего времени
<b>S<sub>2</sub></b>	Дата включения летнего времени
<b>S<sub>3</sub></b>	Месяц отключения летнего времени
<b>S<sub>4</sub></b>	Дата отключения летнего времени
<b>S<sub>5</sub></b>	Настройка изменения значения для летнего времени, ед.изм.: минуты
<b>D</b>	Состояние функции летнего времени: включена в режиме 1

Когда функция летнего времени включена в режиме 2, операнды имеют следующие значения:

Операнд	Описание
<b>S</b>	Функциональный код 7, индикация: функция летнего времени включена в режиме 2
<b>S<sub>1</sub></b>	Месяц включения летнего времени
<b>S<sub>2</sub></b>	Неделя включения летнего времени
<b>S<sub>2</sub> + 1</b>	День недели для <b>S<sub>2</sub></b>
<b>S<sub>3</sub></b>	Месяц отключения летнего времени
<b>S<sub>4</sub></b>	Неделя отключения летнего времени
<b>S<sub>4</sub> + 1</b>	День недели для <b>S<sub>2</sub></b>
<b>S<sub>5</sub></b>	Настройка изменения значения для летнего времени, ед.изм.: минуты
<b>D</b>	Состояние функции летнего времени: включена в режиме 2

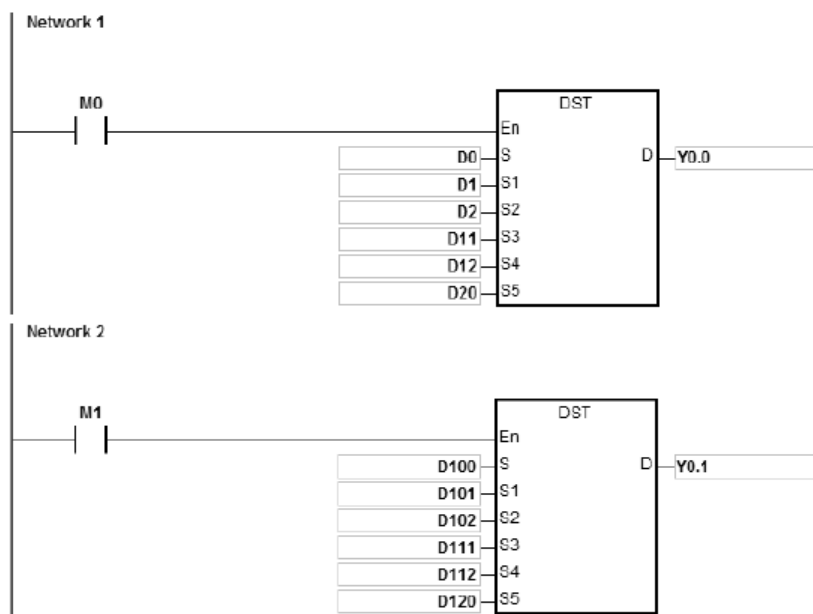
3. Эта инструкция предназначена для включения / отключения функции летнего времени. Если контакт нормально открыт или закрыт, это не повлияет на настройку летнего времени. (см. пример 2 для более подробной информации о том, как включить контакт M0). Вы можете сбросить летнее время, выполнив инструкцию еще раз. Нет необходимости отключать, а затем включать эту функцию для сброса летнего времени.
4. При задании летнего времени с началом 01 апреля и концом 01 сентября, с разницей 60 минут; часы реального времени работают, как показано ниже:

Функция летнего времени отключена	Функция летнего времени включена
1 марта, 3 часа	1 марта, 3 часа

31 марта, 3 часа	31 марта, 3 часа
1 апреля, 3 часа	1 апреля, 4 часа
1 мая, 3 часа	1 мая, 4 часа
1 июня, 3 часа	1 июня, 4 часа
1 июля, 3 часа	1 июля, 4 часа
1 августа, 3 часа	1 августа, 4 часа
31 августа, 3 часа	31 августа, 4 часа
1 сентября, 3 часа	1 сентября, 3 часа

**Пример 1:**

Отключение функции летнего времени и чтение ее состояния:



Заданные параметры:

Объект	Значение	Описание
D0	0	Функция летнего времени отключена
D1	X	Операнд не существует
D2	X	Операнд не существует
D11	X	Операнд не существует
D12	X	Операнд не существует
D20	X	Операнд не существует

Контакт M0 включен

Y0.0 выключен, функция летнего времени отключена.

D100=K2, считывание состояния функции летнего времени.

Контакт M1 включен

Заданные параметры:

Объект	Значение	Описание
D100	2	Прошивка до V1.04 (не включая V1.04): фиксированный функциональный код 2, индикация: считывание состояния функции летнего времени

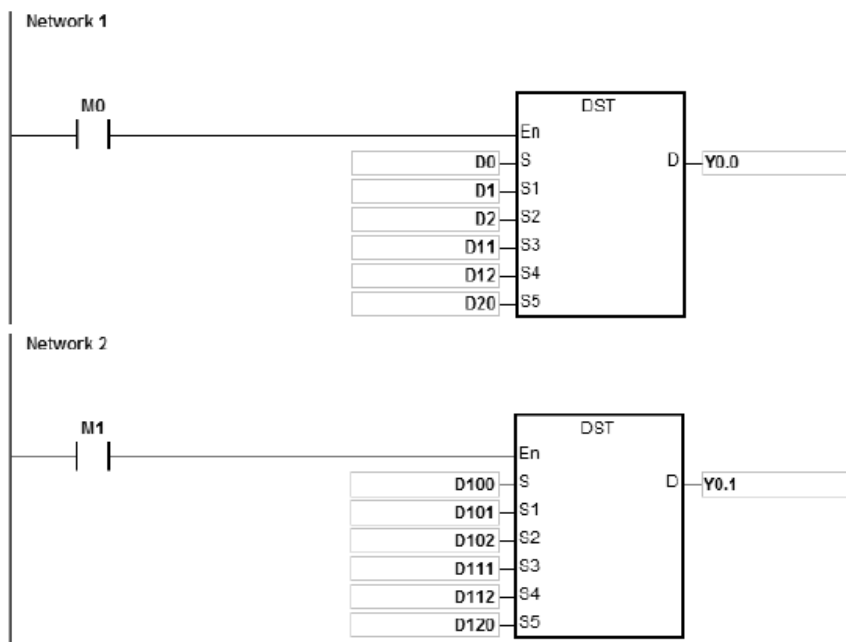


	4	Прошивка выше V1.04 (включая V1.04): функциональный код 4, индикация: функция летнего времени отключена
D101	X	Операнд не существует
D102	X	Операнд не существует
D111	X	Операнд не существует
D112	X	Операнд не существует
D120	X	Операнд не существует
Y0.1	ВЫКЛ	Отключен

**Пример 2:**

Включение функции летнего времени и чтение ее состояния.

Начало летнего времени 01 апреля и окончание 03 сентября, разность – 60 минут:



Заданные параметры:

Объект	Значение	Описание
D0	1	Функция летнего времени включена в режиме 1
D1	4	Месяц начала: Апрель
D2	1	Дата начала: 01
D11	9	Месяц окончания: Сентябрь
D12	3	Дата окончания: 03
D20	60	Разница: 60 минут

Контакт M0 включен

Y0.0 включен, функция летнего времени включена.

Системное время ПЛК 01 апреля добавляет 60 минут и 03 сентября вычитает 60 минут.

D100=K2, считывание состояния функции летнего времени.

Контакт M1 включен

Заданные параметры:

Объект	Значение	Описание
D100	2	Прошивка до V1.04 (не включая V1.04):

		фиксированный функциональный код 2, индикация: считывание состояния функции летнего времени
	5	Прошивка выше V1.04 (включая V1.04): функциональный код 4, индикация: функция летнего времени включена в режиме1
D101	4	Месяц начала: Апрель
D102	1	Дата начала: 01
D111	9	Месяц окончания: Сентябрь
D112	3	Дата окончания: 03
D120	60	Разница: 60 минут
Y0.1	ВКЛ	Включен

Используйте инструкцию DST или HWCONFIG в ISPSOft, чтобы прочитать состояние перехода на летнее время. HWCONFIG автоматически преобразует результат с номера недели в соответствующие даты и месяцы.

Включить снова контакт M0; это на функцию летнего времени, оно не сбрасывается.

Заданные параметры:

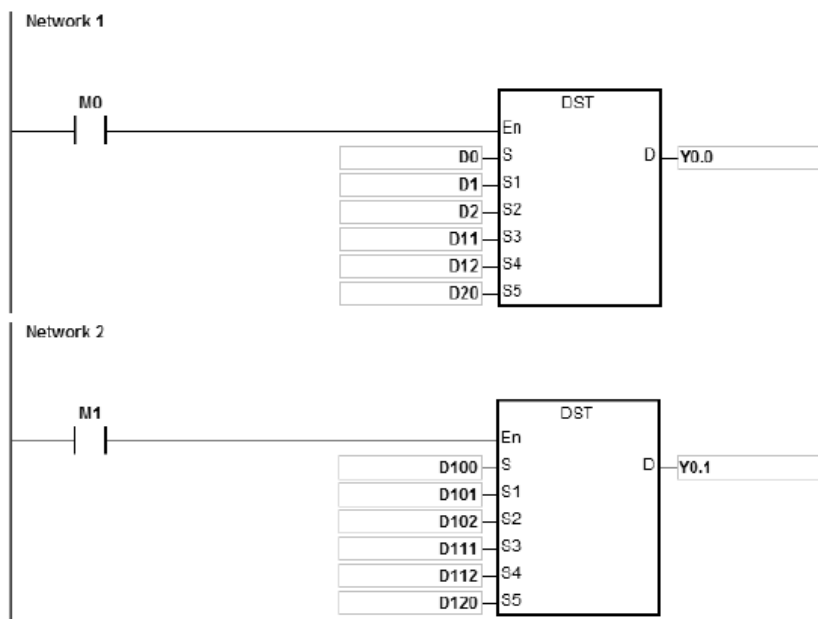
Объект	Значение	Описание
D100	2	Прошивка до V1.04 (не включая V1.04): фиксированный функциональный код 2, индикация: считывание состояния функции летнего времени
	5	Прошивка выше V1.04 (включая V1.04): функциональный код 4, индикация: функция летнего времени включена в режиме1
D101	4	Месяц начала: Апрель
D102	1	Дата начала: 01
D111	9	Месяц окончания: Сентябрь
D112	3	Дата окончания: 03
D120	60	Разница: 60 минут
Y0.1	ВКЛ	Включен

Если летнее время начинается с 3 часов 01 апреля, добавляется 60 минут; часы реального времени показывают 4 часа 01 апреля. Независимо от того, сколько раз контакт M0 отключен или включен, часы реального времени сохраняют одинаковое время для функции летнего времени.

### Пример 3:

Включение функции летнего времени в режиме 2.

Зададим летнее время с 2й среды мая и до 3й пятницы сентября, разница 60 минут:



Заданные параметры:

Объект	Значение	Описание
D0	3	Функция летнего времени включена в режиме 2
D1	5	Месяц начала: Май
D2	2	Стартовая неделя: 2
D3	3	Стартовый день: Среда
D11	9	Месяц окончания: Сентябрь
D12	3	Неделя окончания: 3
D13	5	День окончания: Пятница
D20	60	Разница: 60 минут

Контакт M0 включен

Y0.0 включен, функция летнего времени включена.

В 2017 году вторая среда мая – 10 мая, а 3 пятница сентября – 15 сентября. Системное время ПЛК 10 мая добавляет 60 минут и 15 сентября вычитает 60 минут.

D100=K2, считывание состояния функции летнего времени.

Контакт M1 включен

Заданные параметры:

Объект	Значение	Описание
D100	2	Прошивка до V1.04 (не включая V1.04): фиксированный функциональный код 2, индикация: считывание состояния функции летнего времени
	7	Прошивка выше V1.04 (включая V1.04): функциональный код 4, индикация: функция летнего времени включена в режиме 2
D101	5	Месяц начала: Май
D102	2	Стартовая неделя: 2
D103	3	Стартовый день: Среда
D111	9	Месяц окончания: Сентябрь

D112	3	Неделя окончания: 3
D113	5	День окончания: Пятница
D120	60	Разница: 60 минут
Y0.1	ВКЛ	Включен

Используйте инструкцию DST или HWCONFIG в ISPSoft, чтобы прочитать состояние перехода на летнее время.

HWCONFIG автоматически преобразует результат с номера недели в соответствующие даты и месяцы.

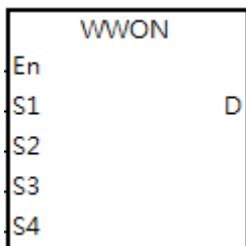
API	Код инструкции			Операнды							Функция						
1608		WWON		<b>S<sub>1</sub> · S<sub>2</sub> · D</b>							Еженедельная настройка рабочего времени						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								○								
S <sub>2</sub>								○								
S <sub>3</sub>								○								
S <sub>4</sub>								○								
D		○	○	○												

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
S <sub>4</sub>		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**



- S<sub>1</sub> : Час начала работы (занимает последовательные 7 регистров)
- S<sub>2</sub> : Минута начала работы (занимает последовательные 7 регистров)
- S<sub>3</sub> : Час окончания работы (занимает последовательные 7 регистров)
- S<sub>4</sub> : Минута окончания работы (занимает последовательные 7 регистров)
- D : Управляемый выход

**Описание:**

1. S<sub>1</sub>~ S<sub>1+6</sub> позволяют задать часы начала работы в воскресенье / понедельник / вторник / среду / четверг / пятницу / субботу соответственно. Этот операнд занимает 7 последовательных регистров, пользователи могут использовать переменные в ARRAY для задания операндов.
2. S<sub>2</sub>~ S<sub>2+6</sub> S<sub>1</sub>~ S<sub>1+6</sub> позволяют задать минуты начала работы в воскресенье / понедельник / вторник / среду / четверг / пятницу / субботу соответственно. Этот операнд занимает 7 последовательных регистров, пользователи могут использовать переменные в ARRAY для задания операндов.
3. S<sub>3</sub>~ S<sub>3+6</sub> позволяют задать часы окончания работы в воскресенье / понедельник / вторник / среду / четверг / пятницу / субботу соответственно. Этот операнд занимает 7 последовательных регистров, пользователи могут использовать переменные в ARRAY для задания операндов.
4. S<sub>4</sub>~ S<sub>4+6</sub> позволяют задать минуты окончания работы в воскресенье / понедельник / вторник / среду / четверг /

пятницу / субботу соответственно. Этот операнд занимает 7 последовательных регистров, пользователи могут использовать переменные в ARRAY для задания операндов.

5. Когда значение часа в  $S_1$  больше значения, установленного в  $S_3$ , это означает, что время прекращения работы на следующий день. Например, когда пользователи установили время для начала работы в 18:00 в понедельник и время прекращения работы в 6:00, это означает, что время прекращения работы в 6:00 вторника.

День	Время начала работы				Время окончания работы			
	Начало	Час	Минута	Секунда	Окончание	Час	Минута	Секунда
Воскресенье	$S_1$	24	00	00	$S_3$	24	00	00
Понедельник	$S_{1+1}$	18	00	00	$S_{3+1}$	06	00	00

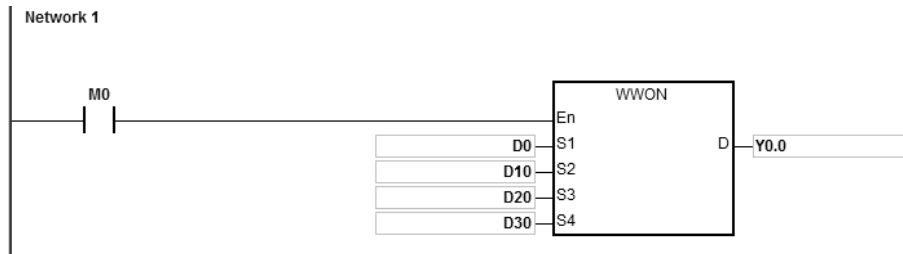
6. Диапазон задания часа 0~23. При выходе за пределы диапазона инструкция не работает. Диапазон задания минут 0~59. При выходе за пределы диапазона инструкция работает, но в качестве настройки принимается 0.
7. Когда требуется задать время работы более 1 дня, можно задать значение часа как 24, и это означает, что система не будет проверять промежуточные время начала и время окончания работы. Например, чтобы установить начальное рабочее время на 8.00 в понедельник и время окончания на 20.00 во вторник, настройки будут следующими:  $S_{1+1} = 8$ ,  $S_{3+1} = 24$ ,  $S_{1+2} = 24$  и  $S_{3+2} = 20$ , см. таблицу ниже:

День	Время начала работы				Время окончания работы			
	Начало	Час	Минута	Секунда	Окончание	Час	Минута	Секунда
Воскресенье	$S_1$	24	00	00	$S_3$	24	00	00
Понедельник	$S_{1+1}$	08	00	00	$S_{3+1}$	24	00	00
Вторник	$S_{1+2}$	24	00	00	$S_{3+2}$	20	00	00

8. Эта инструкция должна работать с часами реального времени. Перед эксплуатацией убедитесь, что аккумулятор надежно установлен и работает нормально.
9. Нет ограничения на количество выполняемых инструкций, но объект управления выходом **D** не может использоваться повторно. При повторно использовании **D**, будет обрабатываться только последний выход для инструкции WWON.
8. Если требуется более 1 рабочего процесса, используйте инструкцию WWON несколько раз по мере необходимости. Обратите внимание, что объект **D** управления выходом не может использоваться повторно.

**Пример 1:**

Задание рабочего времени с 8:00 до 18:00 с понедельника по пятницу, суббота и воскресенье - выходные.



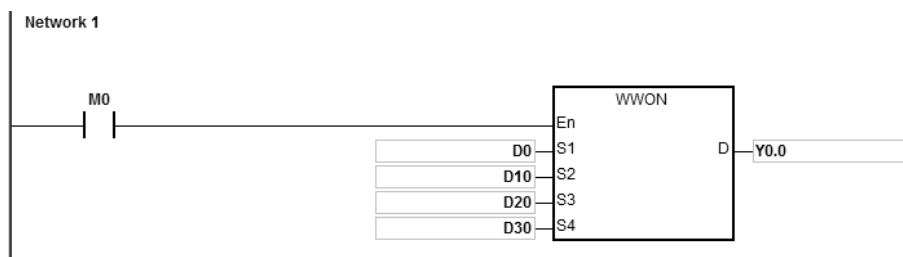
Настройки объекта D:

День	Время начала работы				Время окончания работы			
	Начало	Час	Начало	Час	Окончание	Час	Окончание	Час
Воскресенье	D0	24	D10	00	D20	24	D30	00
Понедельник	D1	08	D11	00	D21	18	D31	00
Вторник	D2	08	D12	00	D22	18	D32	00
Среда	D3	08	D13	00	D23	18	D33	00
Четверг	D4	08	D14	00	D24	18	D34	00
Пятница	D5	08	D15	00	D25	18	D35	00
Суббота	D6	24	D16	00	D26	24	D36	00

Когда M0 включен, выход Y0.0 включается с 8:00 до 18:00 с понедельника по пятницу.

**Пример 2:**

Задание рабочего времени с 18:00 понедельника по 08:00 вторника и с 18:00 вторника до 08:00 среды. Повторяйте эту схему до 08:00 субботы, воскресенье – выходной.



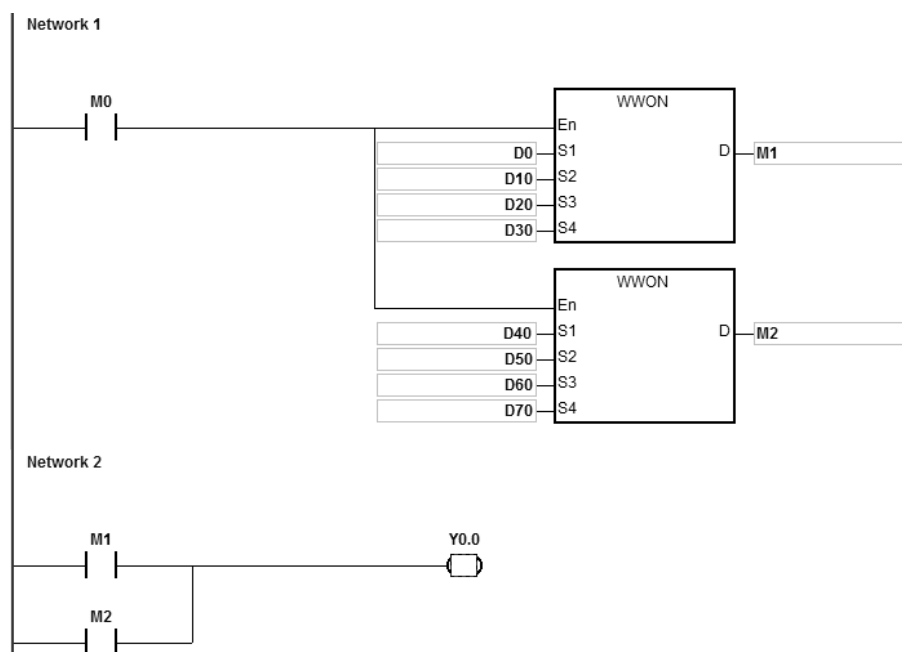
Настройки объекта D:

День	Время начала работы				Время окончания работы			
	Начало	Час	Начало	Час	Окончание	Час	Окончание	Час
Воскресенье	D0	24	D10	00	D20	24	D30	00
Понедельник	D1	18	D11	00	D21	08	D31	00
Вторник	D2	18	D12	00	D22	08	D32	00
Среда	D3	18	D13	00	D23	08	D33	00
Четверг	D4	18	D14	00	D24	08	D34	00
Пятница	D5	18	D15	00	D25	08	D35	00
Суббота	D6	24	D16	00	D26	24	D36	00

Когда M0 включен, выход Y0.0 включается с 18:00 до 8:00 следующего дня с понедельника по пятницу.

**Пример 3:**

Задание рабочего времени с 08:00 до 12:00 и с 14:00 до 17:30 с понедельника по пятницу. Суббота и воскресенье – выходные.





Настройки D для утренних часов работы:

День	Время начала работы				Время окончания работы			
	Начало	Час	Начало	Час	Окончание	Час	Окончание	Час
Воскресенье	D0	24	D10	00	D20	24	D30	00
Понедельник	D1	08	D11	00	D21	12	D31	00
Вторник	D2	08	D12	00	D22	12	D32	00
Среда	D3	08	D13	00	D23	12	D33	00
Четверг	D4	08	D14	00	D24	12	D34	00
Пятница	D5	08	D15	00	D25	12	D35	00
Суббота	D6	24	D16	00	D26	24	D36	00

Настройки D для послеобеденных часов работы:

День	Время начала работы				Время окончания работы			
	Начало	Час	Начало	Час	Окончание	Час	Окончание	Час
Воскресенье	D40	24	D50	00	D60	24	D70	00
Понедельник	D41	14	D51	00	D61	17	D71	30
Вторник	D42	14	D52	00	D62	17	D72	30
Среда	D43	14	D53	00	D63	17	D73	30
Четверг	D44	14	D54	00	D64	17	D74	30
Пятница	D45	14	D55	00	D65	17	D75	30
Суббота	D46	24	D56	00	D66	24	D76	00

Когда M0 включен, выход Y0.0 включается с 08:00 до 12:00 и с 14:00 до 17:30 с понедельника по пятницу.

## 6.18 Инструкции периферийных устройств

### 6.18.1 Описание инструкций периферийных устройств

API	Код инструкции			Операнд								Функция				
1700	D	TKY		S, D <sub>1</sub> , D <sub>2</sub>								Клавиатура с десятью кнопками				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	○															
D <sub>1</sub>					●	●	●	●								
D <sub>2</sub>		○	○	○				○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S	●												
D <sub>1</sub>		●	●		●	●	●						
D <sub>2</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

Символьное обозначение:

TKY		DTKY	
En		En	
S	D1	S	D1
	D2		D2

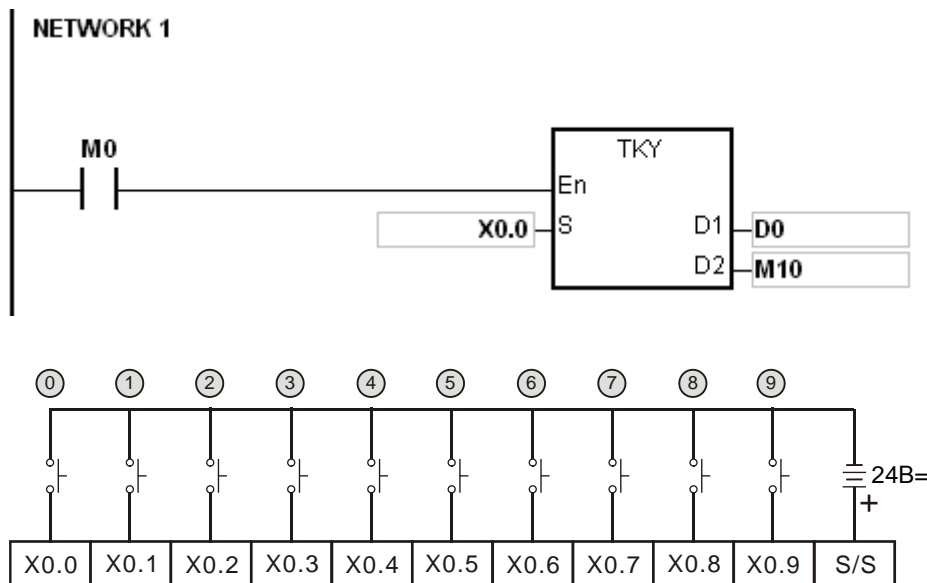
- S** : Исходный регистр
- D<sub>1</sub>** : Регистр, в который записывается значение
- D<sub>2</sub>** : Выходной сигнал

Описание:

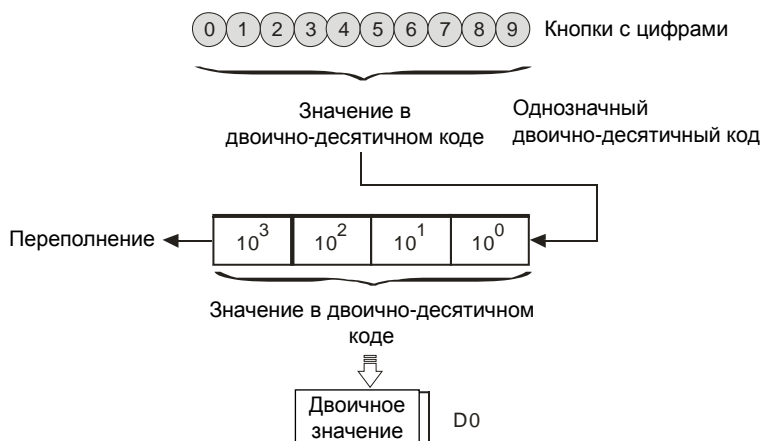
1. Десять входов внешних сигналов, начиная с входа, задаваемого операндом **S**, представляют цифры 0–9 в десятичной системе. К входам подсоединены десять кнопок. Можно вводить четырехзначное десятичное число от 0 до 9,999 (16-битная инструкция) или восьмизначное десятичное число от 0 до 99,999,999 (32-битная инструкция), нажимая кнопки в соответствующем порядке. Десятичное число записывается в регистре **D<sub>1</sub>**, а выходные сигналы в **D<sub>2</sub>**.
2. Операнд **S** занимает десять битов.
3. Операнд **D<sub>2</sub>** занимает одиннадцать битов. Запрещается менять состояние битов во время исполнения инструкции.
4. Если условный контакт выключен, одиннадцать битов, начиная с заданного операндом **D<sub>2</sub>**, имеют состояние OFF.
5. При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.
6. 32-битный счетчик может использоваться, только если **D<sub>1</sub>** использует 32-битные инструкции.

**Пример:**

- Десять входов внешних сигналов, начиная с входа X0.0, подсоединены к десяти кнопкам, представляющим числа от 0 до 9 в десятичной системе. Когда M0 = ON, инструкция начинает исполняться. Введенное значение сохраняется в двоичном виде в регистре D0, а выходные сигналы записываются в M10~M19.

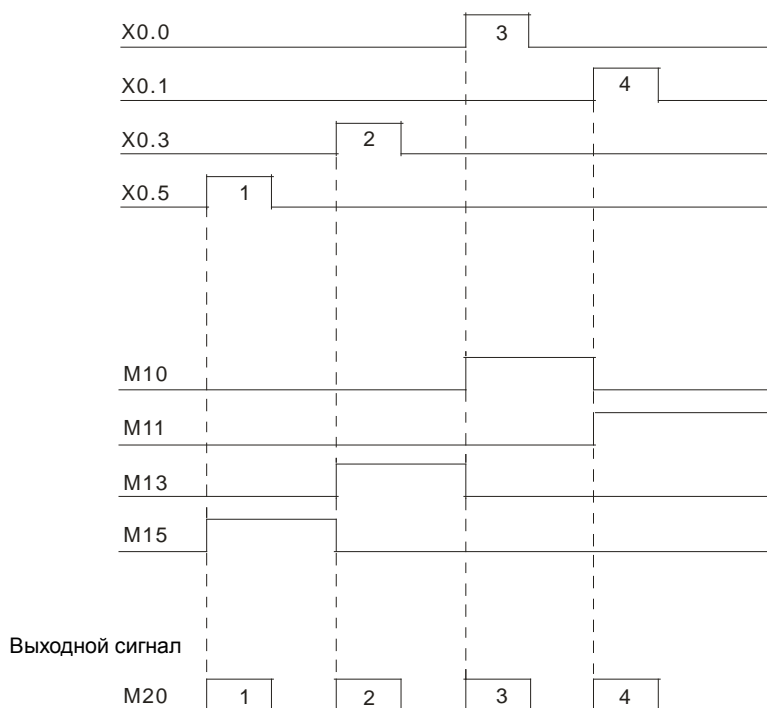


Примечание: На данном примере используется модуль цифровых входов AN16AM10N-5A.



- Если кнопки подсоединены к входам X0.5, X0.3, X0.0 и X0.1 и кнопки нажимаются в порядке, показанном на схеме, в регистр D0 записывается значение 5,301. Максимальным значением, которое можно записать в регистр D0, является 9,999. Если число более чем четырехзначное, первая цифра слева будет считаться лишней.

3. В период времени после нажатия кнопки, подсоединенной к входу X0.2, и до нажатия других кнопок M12 = ON. Данное правило справедливо и для других кнопок.
4. При нажатии кнопки, подсоединенной к входу в диапазоне от X0.0 до X0.9, соответствующий выход в диапазоне от M10 до M19 переходит в состояние ON.
5. При нажатии одной из кнопок выход M20 = ON.
6. Если условный контакт M0 выключен (OFF), записанное в регистре D0 значение остается без изменений. Но M10~M20 выключаются (OFF).



**Дополнительные замечания:**

1. Если операнд **S** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [10] типа BOOL.
2. Если операнд **D<sub>2</sub>** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [11] типа BOOL.

API	Код инструкции			Операнд							Функция					
1701	D	HKY		S <sub>1</sub> , S <sub>2</sub> , D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub>							Клавиатура с шестнадцатью кнопками					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	○															
S <sub>2</sub>								●								
D <sub>1</sub>		○														
D <sub>2</sub>					●	●	●	●								
D <sub>3</sub>		○	○	○				○								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>	●												
S <sub>2</sub>		●			●	●							
D <sub>1</sub>	●												
D <sub>2</sub>		●	●		●	●	●						
D <sub>3</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	AS

**Символьное обозначение:**

HKY		DNKY	
En		En	
S1	D1	S1	D1
S2	D2	S2	D2
	D3		D3

- S<sub>1</sub> : Исходный регистр ввода
- S<sub>2</sub> : Только для контроллера
- D<sub>1</sub> : Исходный регистр вывода
- D<sub>2</sub> : Регистр, в который записывается значение
- D<sub>3</sub> : Выходной сигнал

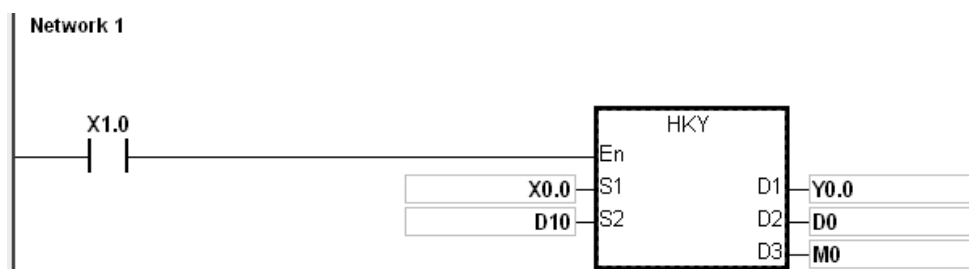
**Описание:**

1. Четыре входа внешних сигналов, начиная с входа заданного операндом S, подсоединены к четырем внешним выходам, начиная с выхода заданного операндом D<sub>1</sub>, получается клавиатура с 16 кнопками. Значение, вводимое нажатиями кнопок, записывается в операнд D<sub>2</sub>, а выходные сигналы в операнд D<sub>3</sub>. Если одновременно нажимается несколько кнопок, записывается наименьшее значение.
2. Значение, вводимое кнопками, временно записывается в операнд D<sub>2</sub>. Если выполняется 16-битная инструкция HKY, максимальным значением, записываемым в операнде D<sub>2</sub>, будет 9,999. Если число более чем четырехзначное, первая цифра слева будет считаться лишней. Если выполняется 32-битная инструкция DNKY, максимальным значением, записываемым в операнде D<sub>2</sub>, будет 9,999. Если число более чем восьмизначное, первая цифра слева будет считаться лишней.
3. По окончании выполнения инструкции SM692 = ON. Иначе говоря, SM692 = ON для цикла сканирования после завершения сканирования матрицы.

4. 32-битный счетчик может использоваться, только если D<sub>2</sub> использует 32-битные инструкции.

**Пример:**

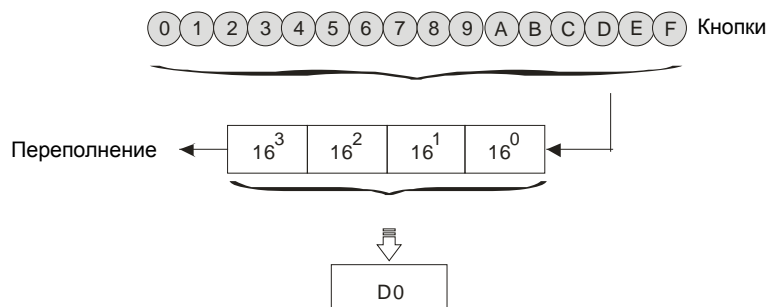
1. Три входа X0.0~X0.3 подсоединены к четырем выходам Y0.0~Y0.3, и получается клавиатура с 16 кнопками. Когда вход X1.0 = ON, инструкция начинает исполняться. Введенное значение сохраняется в двоичном виде в регистре D0, а выходные сигналы записываются в M0~M7.



Функция SM691:

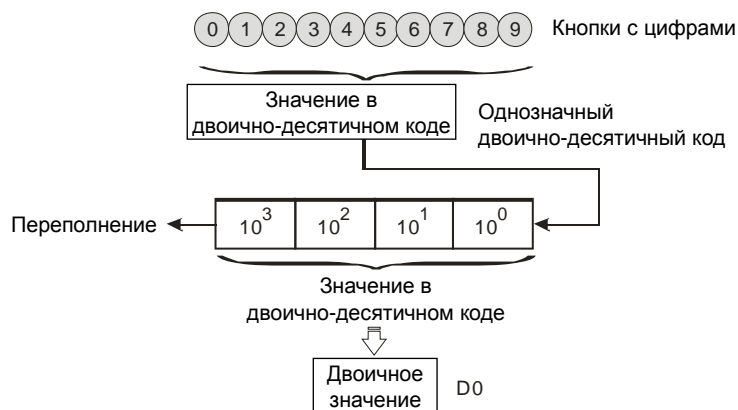
- Если SM691 = ON, при исполнении инструкции НКУ в качестве десятичных значений берутся 0~F.

■ Кнопки с цифрами:



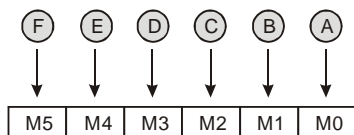
- Если SM691 = OFF, при исполнении инструкции НКУ в качестве функциональных кнопок используются A~F.

■ Кнопки с цифрами:



■ Функциональные кнопки:

- ◆ При нажатии A M0 остается в состоянии ON. При нажатии D M0 переходит в состояние OFF, а M3 остается ON.
- ◆ При нажатии нескольких функциональных кнопок наибольший приоритет имеет нажатая первой.

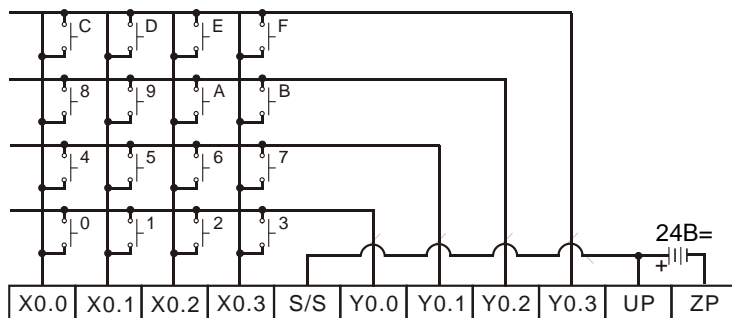


2. Выходные сигналы:

- При нажатии кнопки в диапазоне от A до F M6 будет в состоянии ON.
- При нажатии кнопки в диапазоне от 0 до 9 M7 будет в состоянии ON.

3. Если условный контакт X1.0 выключен (OFF), записанное в регистре D0 значение остается без изменений. Но M0–M7 выключаются (OFF).

4. Внешние цепи:



Примечание: На данном примере используется модуль транзисторных выходов АН16АР16Т-5А.

**Дополнительные замечания:**

1. Если во время исполнения инструкции время цикла опроса будет слишком длинным или, наоборот, коротким, состояние выключателей считать правильно не удастся. Поэтому см. следующие рекомендации по решению данного вопроса.
  - Если цикл опроса слишком короткий, вход/выход может не успеть вовремя изменить состояние, поэтому результат считывания состояния входов может получиться недостоверным. Поэтому, чтобы избежать этого, можно ввести фиксированное время цикла опроса.
  - Если цикл опроса слишком длинный, выключатель может изменять состояние слишком долго. Поэтому можно записать данную инструкцию в прерывание по таймеру, чтобы определить фиксированное время исполнения данной инструкции.
2. Если операнд S объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа BOOL.
3. Если операнд D1 объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа BOOL.
4. Если операнд D3 объявлен в редакторе ISPSOft, типом данных будет МАССИВ [8] типа BOOL.

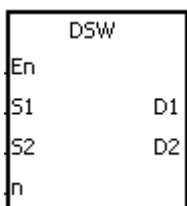
API	Код инструкции			Операнд							Функция					
1702		DSW		$S_1, S_2, D_1, D_2, n$							Микропереключатель					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	○															
$S_2$								●								
$D_1$		○														
$D_2$					●	●		●								
$n$					●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$	●												
$S_2$		●			●	●							
$D_1$	●												
$D_2$		●			●	●							
$n$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- $S_1$  : Исходный регистр ввода
- $S_2$  : Только для контроллера.
- $D_1$  : Исходный регистр вывода
- $D_2$  : Регистр, в который записывается значение
- $n$  : Количество микропереключателей

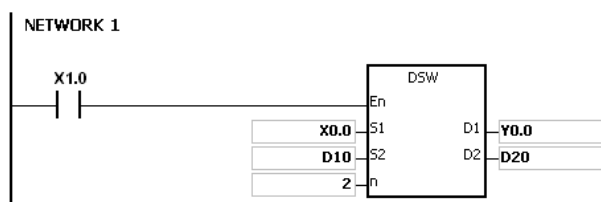
**Описание:**

1. Четыре или восемь входов, начиная с входа задаваемого операндом  $S_1$ , подсоединены к четырем выходам, начиная с выхода задаваемого операндом  $D_1$ , и получается четырехзначный микропереключатель или два четырехзначных микропереключателя. Значение, вводимое микропереключателем, записывается в операнд  $D_2$ . Количество микропереключателей (один четырехзначный или два четырехзначных) задается операндом  $n$ .
2. Если операнд  $n = 1$ , операнд  $D_2$  занимает один регистр. Если операнд  $n = 2$ , операнд  $D_2$  занимает два регистра.
3. Операнды  $S_2$  и  $S_2+1$ , которые используются только контроллером, занимают два регистра. Запрещается менять значения этих регистров.
4. По окончании выполнения инструкция SM694 = ON для цикла сканирования.
5. Если условный контакт выключен, четыре выхода, начиная с заданного операндом  $D_1$ , остаются в состоянии OFF.
6. При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.

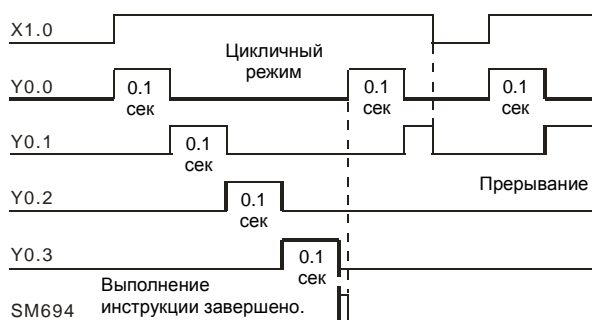


**Пример:**

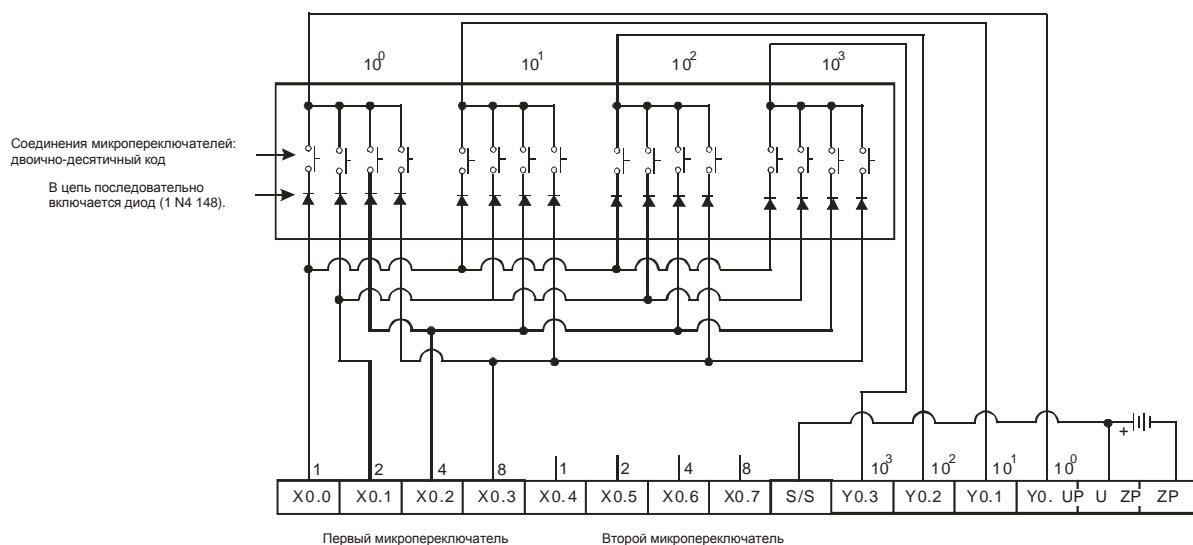
1. Входы X0.0~X0.3 подсоединены к выходам Y0.0~Y0.3 и образуют первый микропереключатель, а входы X0.4~X0.7 подсоединены к выходам Y0.0~Y0.3 и образуют второй микропереключатель. Когда вход X1.0 = ON, инструкция начинает исполняться. Значение, вводимое первым микропереключателем, преобразуется в двоичное значение, и результат данного преобразования записывается в регистр D20. Значение, вводимое вторым микропереключателем, преобразуется в двоичное значение, и результат данного преобразования записывается в регистр D21.



2. Когда X1.0 = ON, выходы Y0.0~Y0.3 переходят в состояние ON в циклическом режиме. По окончании выполнения инструкция SM694 = ON для цикла сканирования.
3. Выходы Y0.0~Y0.3 должны быть транзисторными.



4. Микропереключатели:



Примечание: на данном примере используется модуль транзисторных выходов AH16AP11T-5A.

**Дополнительные замечания:**

1. Если значение **n** вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
2. Если операнд D1 объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа BOOL.

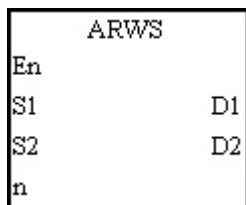
API	Код инструкции			Операнд								Функция				
1703		ARWS		<b>S<sub>1</sub>, S<sub>2</sub>, D<sub>1</sub>, D<sub>2</sub>, n</b>								Кнопки со стрелками				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>	○															
S <sub>2</sub>								●								
D <sub>1</sub>					●	●		●								
D <sub>2</sub>		○														
n					●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>	●												
S <sub>2</sub>		●			●	●							
D <sub>1</sub>		●			●	●							
D <sub>2</sub>	●												
n		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- S<sub>1</sub> : Исходный регистр ввода
- S<sub>2</sub> : Только для контроллера.
- D<sub>1</sub> : Регистр, в который записывается заданное значение
- D<sub>2</sub> : Исходный регистр вывода
- n : Положительная/отрицательная логика

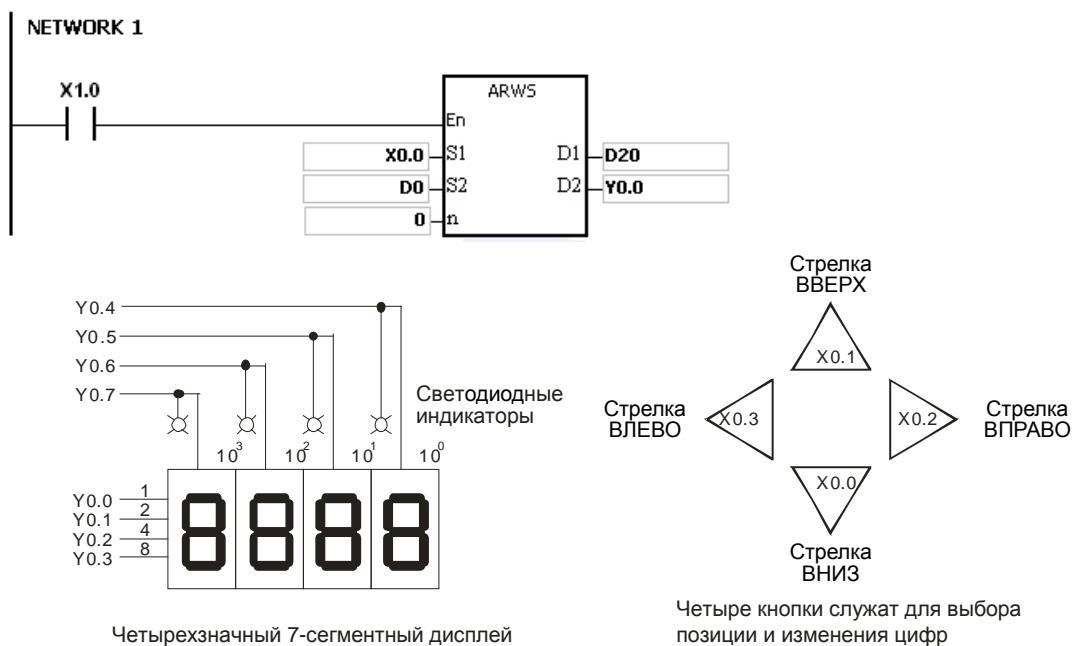
**Описание:**

1. При выполнении инструкции операнд S<sub>1</sub> определяется как кнопка со стрелкой вниз, S<sub>1</sub>+1 как кнопка со стрелкой вверх, S<sub>1</sub>+2 как кнопка со стрелкой вправо и S<sub>1</sub>+3 как кнопка со стрелкой влево. Заданное значение записывается в операнд D<sub>1</sub> и может быть в диапазоне от 0 до 9,999.
2. Операнд S<sub>1</sub> занимает четыре битовых регистра подряд.
3. S<sub>2</sub> используется только контроллером. Запрещается менять его значение.
4. Операнд D<sub>2</sub> занимает восемь битовых регистров подряд.
5. Если условный контакт выключен, восемь битовых регистров, начиная с заданного операндом D<sub>2</sub>, имеют состояние OFF.
6. Операнд n может иметь значение в диапазоне от 0 до 3. Подробнее см. примечания по инструкции API1704 SEGL.

- При редактировании в режиме онлайн необходимо сбрасывать состояние условного контакта, чтобы инициализировать инструкцию.

**Пример:**

- При выполнении инструкции вход X0.0 определяется как кнопка со стрелкой вниз, вход X0.1 как кнопка со стрелкой вверх, вход X0.2 как кнопка со стрелкой вправо и вход X0.3 как кнопка со стрелкой влево. Заданное значение записывается в регистр D20 и может быть в диапазоне от 0 до 9,999.
- Если вход X1.0 = ON, выбирается цифра на позиции  $10^3$ . Если нажата кнопка со стрелкой влево, позиции меняются следующим образом ( $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$ ).
- Если нажата кнопка со стрелкой вправо, меняются следующим образом ( $10^3 \rightarrow 102 \rightarrow 10^1 \rightarrow 100 \rightarrow 10^3 \rightarrow 102$ ). Светодиодные индикаторы соответствующих позиций соединены с выходами Y0.4~Y0.7. Когда цифры позиций меняются, светодиодные индикаторы, соответственно, загораются.
- Если нажата кнопка со стрелкой влево, цифра на позиции меняется ( $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots 8 \rightarrow 9 \rightarrow 0 \rightarrow 1$ ). Если нажата кнопка со стрелкой вниз, цифра на позиции меняется ( $0 \rightarrow 9 \rightarrow 8 \rightarrow \dots 1 \rightarrow 0 \rightarrow 9$ ). Новая цифра появляется на 7-сегментном дисплее.



**Дополнительные замечания:**

- Если значение **n** вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200B.
- Если операнд **S<sub>1</sub>** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа BOOL.
- Если операнд **D<sub>2</sub>** объявлен в редакторе ISPSOft, типом данных будет МАССИВ [8] типа BOOL.

API	Код инструкции			Операнд							Функция					
1704		SEGL		<b>S<sub>1</sub>, S<sub>2</sub>, D, n</b>							7-сегментный дисплей с защелкой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S<sub>1</sub></b>	●	●			●	●		●	●		○	○				
<b>S<sub>2</sub></b>								●								
<b>D</b>		○														
<b>n</b>					●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>		●			●	●							
<b>S<sub>2</sub></b>		●			●	●							
<b>D</b>	●												
<b>n</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**



- S<sub>1</sub>** : Исходный регистр
- S<sub>2</sub>** : Только для контроллера.
- D** : Исходный регистр вывода
- n** : Положительная/отрицательная логика

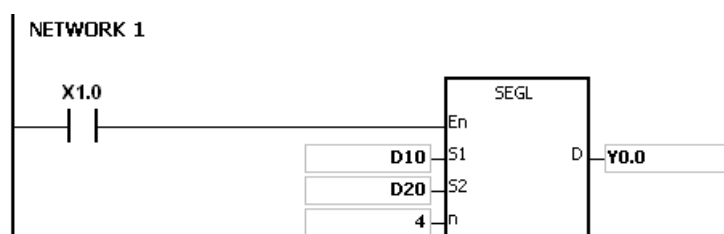
**Описание:**

1. Восемь выходов, начиная с выхода заданного операндом **D**, подсоединены к четырехзначному 7-сегментному дисплею или двенадцать выходов, начиная с заданного операндом **D**, подсоединены к двум четырехзначным 7-сегментным дисплеям. Каждая позиция имеет драйвер, преобразующий число в двоично-десятичном коде в данные для отображения на 7-сегментном дисплее, и каждый драйвер имеет по защелке, используемой для хранения данных состояния.
2. Значение операнда **S<sub>1</sub>** показывается на первом 7-сегментном дисплее, а значение **S<sub>1</sub>+1** показывается на втором 7-сегментном дисплее.
3. **S<sub>2</sub>** используется только контроллером. Запрещается менять его значение.
4. Операнд **n** может иметь значение в диапазоне от 0 до 7. Подробнее см. дополнительные примечания.
5. Операнд **n** определяет логическую схему выхода (положительная или отрицательная) и количество четырехзначных 7-сегментных дисплеев (один или два).
6. Если четырехзначный 7-сегментный дисплей один, занято восемь выходов. Если четырехзначных 7-сегментных дисплеев два, занято двенадцать выходов.

7. При выполнении инструкции выходы циклично включаются (ON). Если во время выполнения инструкции условный контакт меняет состояние с OFF на ON, выходы снова циклично включаются (ON).
8. По окончания выполнения инструкция SM693 = ON для цикла сканирования.

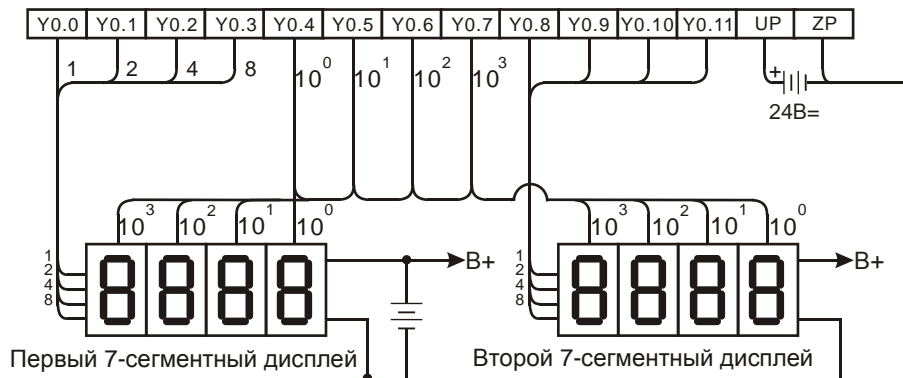
**Пример:**

1. Когда вход X1.0 = ON, инструкция начинает исполняться. Выходы Y0.0~Y0.4 образуют цепь. Значение регистра D10 преобразуется в число в двоично-десятичном коде, и результат выводится на первом 7-сегментном дисплее. Значение регистра D11 преобразуется в число в двоично-десятичном коде, и результат выводится на втором 7-сегментном дисплее. Если значение D10 или D11 больше 9,999, выдается ошибка операции.



2. Когда X1.0 = ON, выходы Y0.4~Y0.7 переходят в состояние ON в циклическом режиме. На включение выходов Y0.4~Y0.7 уходит двенадцать циклов сканирования. По окончании выполнения инструкция SM693 = ON для цикла сканирования.
3. Если четырехзначный 7-сегментный дисплей один, операнд **n** может иметь значение от 0 до 3.
  - Когда контакты 1, 2, 4 и 8 подсоединены параллельно, они подсоединяются к выходам Y0.0~Y0.3 ПЛК, а защелки подсоединяются к выходам Y0.4~Y0.7 ПЛК.
  - Когда вход X1.0 = ON, инструкция начинает исполняться. Выходы Y0.4~Y0.7 циклично включаются, а значение регистра D10 выводится на 7-сегментном дисплее.
4. Если четырехзначных 7-сегментных дисплея два, операнд **n** может иметь значение от 4 до 7.
  - Когда контакты 1, 2, 4 и 8 подсоединены параллельно, они подсоединяются к выходам Y0.8~Y0.11 ПЛК, а защелки подсоединяются к выходам Y0.4~Y0.7 ПЛК.
  - Значение регистра D10 выводится на первом 7-сегментном дисплее, а регистра D11 – на втором. Если значения регистров D10 и D11 равны 1234 и 4321 соответственно, на втором дисплее выводится 1234.

5. Цепи:



Примечание: На данном примере используется модуль транзисторных выходов AH16AN01T-5A.

Дополнительные замечания:

- Операнд **n** определяет количество четырехзначных 7-сегментных дисплеев (один или два).
- Выходы ПЛК должны быть транзисторными типа NPN с открытым коллектором. Кроме этого, у выхода должен быть резистор, подтягивающий к контакту постоянного тока (меньше 30В=). Поэтому, когда выход включен (ON), выдается сигнал низкого потенциала.
- Отрицательная логика:

Число в двоично-десятичном коде				Результат (двоично-десятичный код)				Сигнал			
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	8	4	2	1	A	B	C	D
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	1	0	0	0	1	1	1	1	0
0	0	1	0	0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	1	1	1	0	0
0	1	0	0	0	1	0	0	1	0	1	1
0	1	0	1	0	1	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0	0	1
0	1	1	1	0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1	0

- Положительная логика:

Число в двоично-десятичном коде				Результат (двоично-десятичный код)				Сигнал			
b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>	8	4	2	1	A	B	C	D
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	0	0	1	0
0	0	1	1	1	1	0	0	0	0	1	1
0	1	0	0	1	0	1	1	0	1	0	0
0	1	0	1	1	0	1	0	0	1	0	1
0	1	1	0	1	0	0	1	0	1	1	0
0	1	1	1	1	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	0	0
1	0	0	1	0	1	1	0	1	0	0	1

- Защелка:

Положительная логика		Отрицательная логика	
Зашелка	Сигнал	Зашелка	Сигнал
1	0	0	1

- Настройки параметра n:

Количество 7-сегментных дисплеев	Один				Два			
	+		-		+		-	
Выход (двоично-десятичный код)	+		-		+		-	
Зашелка	+	-	+	-	+	-	+	-
n	0	1	2	3	4	5	6	7

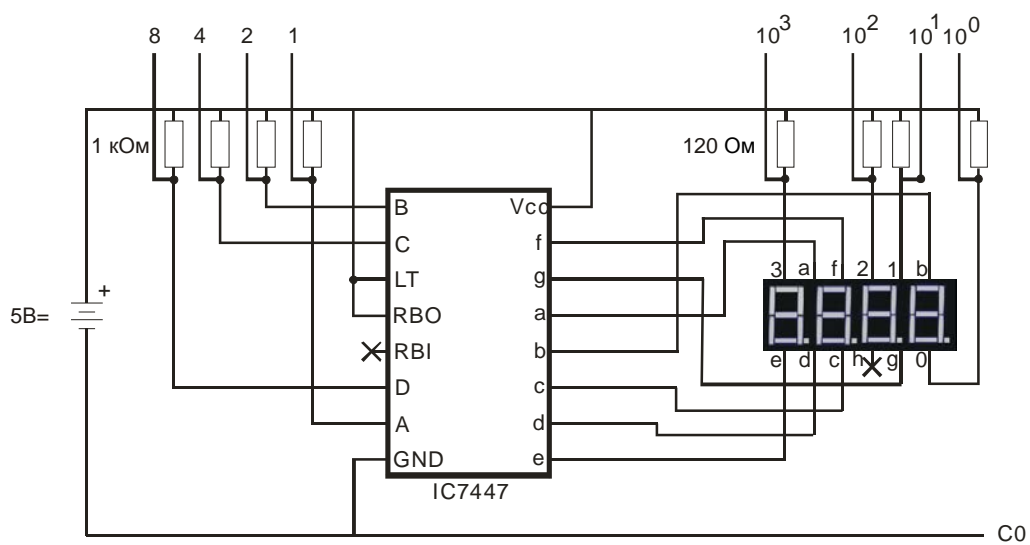
«+»: положительная логика

«-»: отрицательная логика

- Можно редактировать параметры n, чтобы изменить логическую схему транзисторного выхода и входа 7-сегментного дисплея.



- Схема подключения четырехзначного 7-сегментного дисплея с общим анодом и IC 7447 показана ниже.



## 6.19 Инструкции коммуникации

### 6.19.1 Описание инструкций коммуникации

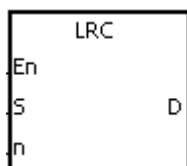
API	Код инструкции			Операнды						Функция							
1806		LRC		S · n · D						Продольная проверка четности							

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S								●	●							
n								●	●				○	○		
D								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



**S** : Исходный объект для инструкции LRC

**n** : Число байтов

**D** : Операнд сохранения результата

Описание:

- См. описание инструкции LRC для дополнительной информации по проверке четности LRC.
- Операнд **n** должен быть четным числом и его значение должно находиться в диапазоне от 1 до 1000. Если значение **n** не находится в пределах диапазона, возникает ошибка, инструкция не выполняется, флаги SM0 и SM1 включены, а в SR0 записывается код ошибки 16#200B.
- Режим 16-битного преобразования: когда SM606 выключен, шестнадцатеричные данные в **S**, делятся на старшие 8 бит данных и младшие 8 бит данных. Инструкция LRC применяется к каждому байту, и результат операции сохраняется в старших 8 битах и младших 8 битах в операнде **D**. Число байтов зависит от значения **n**.
- Режим 8-битного преобразования: когда SM606 включен, шестнадцатеричные данные в **S**, делятся на старшие 8 бит данных и младшие 8 бит данных. Инструкция LRC применяется к каждому байту, и результат операции сохраняется в младших 8 битах в двух регистрах в операнде **D**. Число байтов зависит от значения **n** (значения старших 8 бит в двух регистрах равны 0).

**Пример:**

1. ПЛК подключается к преобразователю частоты (ПЧ) серии VFD (режим ASCII: SM210 выключен, 8-битный режим: SM606 включен). ПЛК отправляет команду и считывает данные в шести объектах по адресам, начиная с 16#2101 в ПЧ серии VFD.

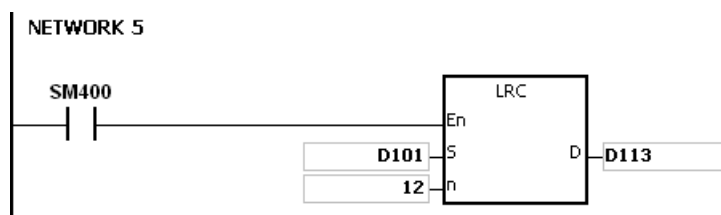
ПЛК ⇔ VFD

ПЛК отправляет данные " : 01 03 2101 0006 D4 CR LF".

Отправляемые ПЛК данные.

Регистр	Данные		Описание	
D100 Мл. 8 бит	' : '	16#3A	STX	
D101 Мл. 8 бит	'0'	16#30	ADR 1	AD (1, 0) – адрес станции ПЧ.
D102 Мл. 8 бит	'1'	16#31	ADR 0	
D103 Мл. 8 бит	'0'	16#30	CMD 1	CMD (10) – командный код.
D104 Мл. 8 бит	'3'	16#33	CMD 0	
D105 Мл. 8 бит	'2'	16#32	Исходный адрес данных	
D106 Мл. 8 бит	'1'	16#31		
D107 Мл. 8 бит	'0'	16#30		
D108 Мл. 8 бит	'1'	16#31		
D109 Мл. 8 бит	'0'	16#30	Количество данных (по словам)	
D110 Мл. 8 бит	'0'	16#30		
D111 Мл. 8 бит	'0'	16#30		
D112 Мл. 8 бит	'6'	16#36		
D113 Мл. 8 бит	'D'	16#44	LRC CHK 0	LRC CHK (01) – код проверки ошибок.
D114 Мл. 8 бит	'4'	16#34	LRC CHK 1	
D115 Мл. 8 бит	CR	16#0D	END	
D116 Мл. 8 бит	LF	16#0A		

LRC CHK (01) – код проверки ошибок. Его можно рассчитать с помощью инструкции LRC. (8-битный режим: флаг SM606 включен)



Код проверки LRC:  $16\#01+16\#03+16\#21+16\#01+16\#00+16\#06=16\#2C$

Второе дополнение  $16\#2C$  равно  $16\#D4$ . 'D' ( $16\#44$ ) сохраняется в 8 младших битах D113, а '4' ( $16\#34$ ) сохраняется в 8 младших битах в D114.

**Примечание:**

1. Формат данных коммуникации в режиме ASCII mode:

<b>STX</b>	' : '	Символ начала сообщения ' : ' ( $16\#3A$ ).
<b>Address Hi</b>	' 0 '	Коммуникационный адрес:
<b>Address Lo</b>	' 1 '	8-битный адрес состоит из двух кодов ASCII.
<b>Function Hi</b>	' 0 '	Функциональный код:
<b>Function Lo</b>	' 3 '	8-битный функциональный код состоит из двух кодов ASCII.
<b>DATA ( n-1 )</b> ..... <b>DATA 0</b>	' 2 '	Данные: $n \times 8$ -битные данные состоят из $2n$ кодов ASCII.
	' 1 '	
	' 0 '	
	' 2 '	
	' 0 '	
	' 0 '	
	' 2 '	
<b>LRC CHK Hi</b>	' D '	Код проверки LRC:
<b>LRC CHK Lo</b>	' 7 '	8-битный код проверки состоит из двух кодов ASCII.
<b>END Hi</b>	CR	Символ окончания сообщения:
<b>END Lo</b>	LF	END Hi=CR ( $16\#0D$ ), END Lo=LF ( $16\#0A$ )

2. Код проверки LRC: Значения, начинающиеся с адреса коммуникации и до данных, складываются. Второе дополнение суммы является кодом проверки LRC.

Пример:  $16\#01+16\#03+16\#21+16\#02+16\#00+16\#02=16\#29$

Второе дополнение  $16\#29$  равно  $16\#D7$ .

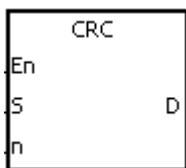
API	Код инструкции			Операнды						Функция					
1807		CRC		S · n · D						Циклическая проверка избыточности					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S								●	●							
n								●	●				○	○		
D								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**



- S** : Исходный операнд, к которому применяется CRC
- n** : Число байт
- D** : Операнд сохранения результата операции

**Описание:**

1. Дополнительную информацию о контрольном коде CRC см. Примечание к описанию инструкции CRC.
2. Значение операнда **n** должно находиться в пределах от 1 до 1000. Если значение **n** не находится в пределах диапазона, возникает ошибка, инструкция не выполняется, флаги SM0 и SM1 включены, а в SR0 записывается код ошибки 16#200B.
5. Режим 16-битного преобразования: когда SM606 выключен, шестнадцатеричные данные в **S**, делятся на старшие 8 бит данных и младшие 8 бит данных. Инструкция CRC применяется к каждому байту, и результат операции сохраняется в старших 8 битах и младших 8 битах в операнде **D**. Число байтов зависит от значения **n**.
3. Режим 8-битного преобразования: когда SM606 включен, шестнадцатеричные данные в **S**, делятся на старшие 8 бит данных (недопустимые данные) и младшие 8 бит данных. CRC применяется к каждому байту, и результат операции сохраняется в младших 8-битных двух регистрах. Число байтов зависит от значения **n**.

**Пример:**

1. ПЛК подключается к преобразователю частоты (ПЧ) серии VFD (режим RTU: флаг SM210 включен; 16-битный режим: флаг SM606 включен). Значение 16#12, которое будет записано в операнд при значении 16#2000 в ПЧ серии VFD-S, сначала записывается в ПЛК.

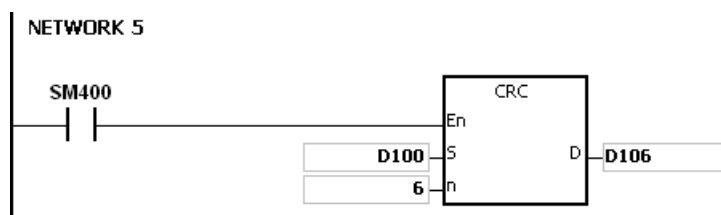
ПЛК⇒VFD

ПЛК отправляет данные 01 06 2000 0012 02 07.

Отправляемые ПЛК данные.

Регистр	Данные	Описание
D100 Мл. 8 бит	16#01	Адрес
D101 Мл. 8 бит	16#06	Функция
D102 Мл. 8 бит	16#20	Адрес данных
D103 Мл. 8 бит	16#00	
D104 Мл. 8 бит	16#00	Данные
D105 Мл. 8 бит	16#12	
D106 Мл. 8 бит	16#02	CRC CHK 0
D107 Мл. 8 бит	16#07	CRC CHK 1

CRC CHK (01) – код проверки ошибок. Его можно рассчитать с помощью инструкции CRC. (8-битный режим: флаг SM606 включен)



Код проверки CRC: 16#02 сохраняется в 8 младших битах D106, а 16#07 сохраняется в 8 младших битах D107.

**Примечание:**

1. Формат коммуникационных данных в режиме RTU:

<b>START</b>	Временной интервал
<b>Address</b>	Коммуникационный адрес: 8-битный двоичный адрес
<b>Function</b>	Функциональный код: 8-битный двоичный код
<b>DATA ( n-1 )</b>	Данные: nх8-битные данные
.....	
<b>DATA 0</b>	
<b>CRC CHK Low</b>	Код проверки CRC: 16-битный код проверки содержит два 8-битных двоичных кода.
<b>CRC CHK High</b>	
<b>END</b>	Временной интервал

2. Код проверки CRC: Код проверки начинается с адреса данных. Операция выглядит следующим образом:

Шаг 1: Предположим, что данные в 16-битном регистре (регистр, в котором хранится код проверки CRC) равны 16#FFFF.

Шаг 2: Логическая инструкция XOR принимает сообщение из старших и младших 8 бит данных в 16-битном регистре и выполняет логическую операцию «Исключающее ИЛИ» по каждой паре соответствующих битов. Результат операции сохраняется в 16-битном регистре.

Шаг 3: Значения битов в 16-битных регистрах сдвигаются на один бит вправо. Значение старшего бита становится равным 0.

Шаг 4: Если значение самого правого бита, которое сдвинуто вправо, равно 0, данные, полученные на шаге 3, сохраняются в 16-битном регистре. В противном случае логическая инструкция XOR принимает значение 16#A001 и данные в 16-битном регистре и выполняет логическую операцию «Исключающее ИЛИ» для каждой пары соответствующих битов. Результат операции сохраняется в 16-битном регистре.

Шаг 5: Повторите шаги 3 и 4, затем выполните операцию с 8-битным сообщением.

Шаг 6: Повторите шаги 2 ~ 5 и получите следующее 8-битное сообщение. Выполните операции над всеми сообщениями. Конечным результатом в 16-битном регистре является код проверки CRC. Обратите внимание, что младшие 8 бит данных в 16-битном регистре меняются со старшими 8-битными данными в 16-разрядном регистре до того, как код проверки CRC помещается в контрольный код сообщения.

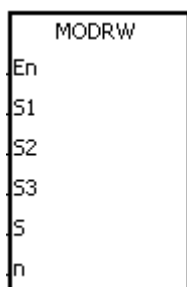
API	Код инструкции			Операнды						Функция					
1808		MODRW		$S_1 \cdot S_2 \cdot S_3 \cdot S \cdot n$						Чтение/запись данных MODBUS					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$								●	●				○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		
$S$								●								
$n$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							
$S$	●	●			●	●							
$n$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



- $S_1$  : Адрес устройства
- $S_2$  : Функциональный код
- $S_3$  : Адрес данных
- $S$  : Регистр чтения/записи данных
- $n$  : Длина данных

Описание:

1. Значение операнда  $S_1$  должно быть в диапазоне от 0 до 254; 0 – широковещательный режим.
2.  $S_2$ : функциональный код

Пример:

Функциональный код	Описание	Длина данных	Объекты, поддерживаемые в ведомых устройствах
01	ПЛК считывает данные с нескольких битовых регистров.	1~1600	X, Y, M, SM, S, T, C, HC
02	ПЛК считывает данные с нескольких битовых регистров.	1~1600	X, Y, M, SM, S, T, C, HC



Функциональный код	Описание	Длина данных	Объекты, поддерживаемые в ведомых устройствах
03	ПЛК считывает данные с нескольких словных регистров.	1~100	X, Y, SR, D, T, C, HC, E
04	ПЛК считывает данные с нескольких словных регистров.	1~100	X
05	ПЛК записывает состояние в битовый регистр.	1	Y, M, SM, S, T, C, HC
06	ПЛК записывает состояние в словный регистр.	1	Y, SR, D, T, C, HC, E
0F	ПЛК записывает данные в несколько битовых регистров.	1~1600	Y, M, SM, S, T, C, HC
10	ПЛК записывает данные в несколько словных регистров.	1~100	Y, SR, D, T, C, HC, E

Поддерживаются только упомянутые выше функциональные коды, другие функциональные коды не поддерживаются. См. приведенные ниже примеры.

3. **S<sub>3</sub>**: Адрес данных. Если адрес назначен неправильно, будет передано сообщение об ошибке.

4. **S**: Регистр, занятый в чтении / записи данных.

Данные, которые будут записаны во внешнее оборудование, заносятся в данный регистр заранее.

Данные, которые считываются с внешнего оборудования, хранятся в данном регистре.

5. **n**: Длина данных.

Для функционального кода словного типа длина данных не может превышать 100 слов.

Для функционального кода битового типа длина данных варьируются от 1 до 1600 бит.

6. Функции **S<sub>3</sub>**, **S** и **n** изменяются с помощью функционального кода.

Функциональный код	<b>S<sub>3</sub></b>	<b>S</b>	<b>n</b>
H01	Адрес, с которого считываются данные.	Регистр сохранения данных.	Длина считываемых данных
H02	Адрес, с которого считываются данные.	Регистр сохранения данных.	Длина считываемых данных
H03	Адрес, с которого считываются данные.	Регистр сохранения данных.	Длина считываемых данных

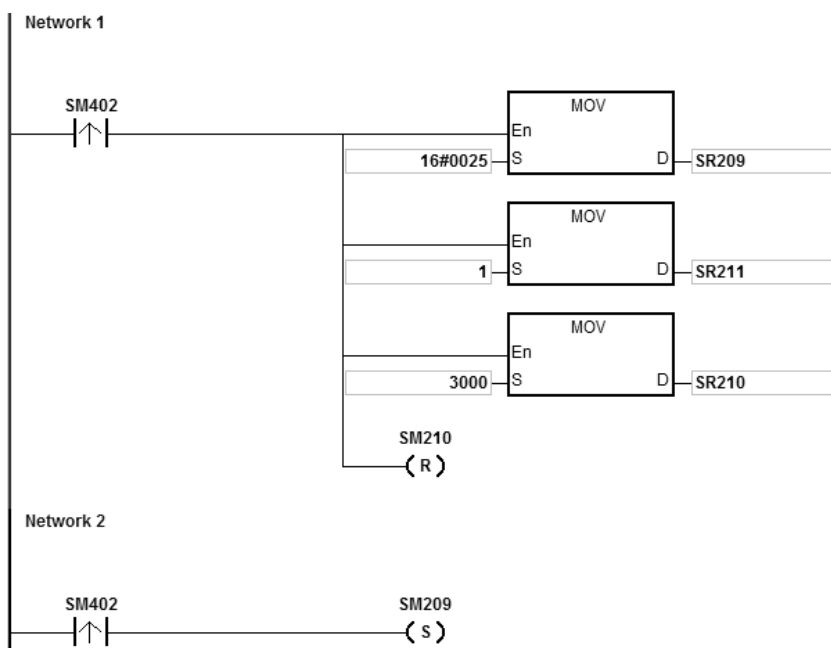
Функциональный код	S <sub>3</sub>	S	n
H04	Адрес, с которого считываются данные.	Регистр сохранения данных.	Длина считываемых данных
H05	Адрес, в который данные записываются.	Значение состояния записи	Не используется
H06	Адрес, в который данные записываются.	Регистр сохранения записанных данных	Не используется
H0F	Адрес, в который данные записываются.	Регистр сохранения записанных данных	Длина записываемых данных
H10	Адрес, в который данные записываются.	Регистр сохранения записанных данных	Длина записываемых данных

7. Ограничений по времени действия инструкции нет, но только одна инструкция может быть выполнена на одном COM-порту одновременно.
8. Если происходит пауза коммуникации, включены флаги времени ожидания. После устранения проблемы пользователь должен сбросить эти флаги. При использовании инструкции MODRW значение паузы не может быть равно 0, оно должно быть задано в диапазоне от 100 до 32767 мс; когда значение задано как 0, она автоматически скорректируется до 200 мс.
9. В режиме MODBUS ASCII пользователю необходимо настроить данные (для не ASCII-режима) для передачи, инструкция преобразует режим не ASCII в ASCII-режим, состоящий из начального кода (:), преобразованного ASCII-кода, контрольной суммы (LRC) и конечного кода (CRLF). Полученные данные сохраняются как символ ASCII во внутреннем регистре. ПЛК автоматически преобразует данные в шестнадцатеричное значение, и если данные коммуникации верны, результат преобразования будет сохранен в **S** и будет включен флаг завершения SM.
10. В режиме MODBUS RTU пользователю необходимо только настроить данные для передачи, инструкция добавит контрольную сумму (CRC), а полученные данные будут сохранены как символ ASCII во внутреннем регистре. ПЛК автоматически преобразует данные в шестнадцатеричное значение, и если данные коммуникации верны, результат преобразования будет сохранен в **S**.
11. Эта инструкция не может использоваться на языке программирования ST, задачах прерывания или назначенных функциональных блоках.

**Пример настройки протокола коммуникации:**

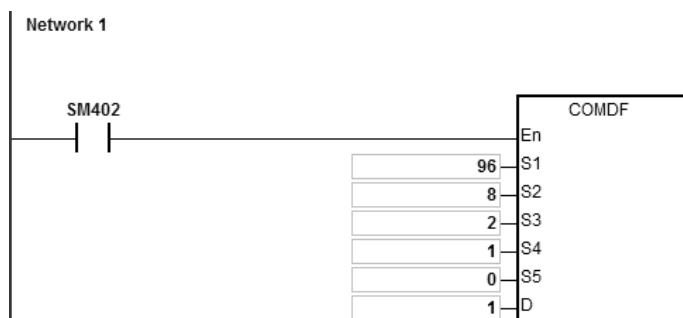
1. Следующие примеры, использующие коммуникационный порт 1 в ПЛК и специальные регистры, демонстрируют настройку протокола коммуникации.

2. Пользователь может настроить порт коммуникации ПЛК через HWCONFIG ISPSOft, через специальные регистры или инструкцию API1813 COMDF. См. Руководство ISPSOft для настроек функции HWCONFIG. Что касается настроек регистра связи (SM, SR), см. Раздел 6.19.3.
3. Настройка коммуникации для этого примера - RS485 ASCII, 9600, 8, E, 1 (SR209 = 16 # 0025).
4. Задайте задержку (паузу) для коммуникации как 3000 мс (SR210 = 3000).
5. Установите режим коммуникации как ASCII-режим (SM210 = OFF).
6. Активируйте протокол коммуникации (SM209 = ON).



6

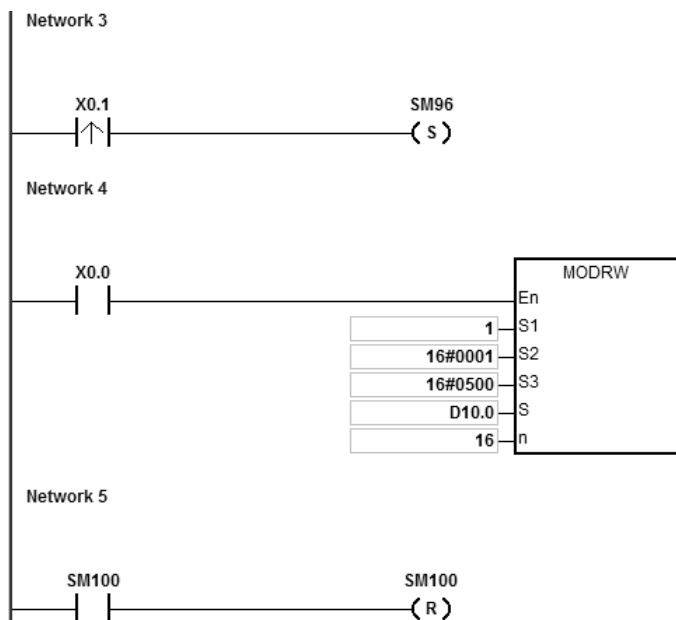
Если порт коммуникации настроен с помощью инструкции API1813 COMDF, этот шаг можно игнорировать.



Если порт коммуникации настроен с помощью процедуры в ПО ISPSOft - > HWCONFIG - > COM Port, этот шаг можно игнорировать.

**Пример 1:**

1. Функциональный код 01 (16#01): ПЛК считывает данные с нескольких битовых регистров, которые не являются дискретными входами (в этом примере считывается 16 фрагментов данных). Для функционального кода 02 операция такая же, как и для функционального кода 01.



ЦПУ ПЛК серии AS подключается к ПЛК серии DVP-ES2.

Когда SM96 и X0.0 включены, ЦПУ AS отправляет и принимает команды с помощью выходов Y0 ~ Y17 ПЛК DVP-ES2.

Когда адрес Y0 равен 16#0500, состояния Y0~17 в DVP-ES2 перечислены ниже:

<b>Устройство</b>	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
<b>Состояние</b>	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ
<b>Значение</b>	D				2			
<b>Устройство</b>	Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10
<b>Состояние</b>	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ
<b>Значение</b>	0				4			

Операнды инструкции MODRW:

Операнд	Описание	Значение
S <sub>1</sub>	Адрес устройства	16#0001
S <sub>2</sub>	Функциональный код	16#0001

<b>S<sub>3</sub></b>	Адрес данных	16#0500
<b>S</b>	Регистр, используемый при чтении/записи данных	D10.0
<b>n</b>	Длина данных	16

**Режим ASCII:**

Коды ASCII необязательно преобразовываются намеренно и имеют значения в формате 16#.

- Команда коммуникации для отправления данных: “ : 01 01 05 00 00 10 E9 CR LF”.
- Команда коммуникации для приема данных: “ : 01 01 02 D2 04 26 CR LF”.

**Режим RTU:**

- Команда коммуникации для отправления данных: “01 01 05 00 00 10 3D 0A”.
- Команда коммуникации для приема данных: “01 01 02 D2 04 E4 9F”.

Если формат корректен, включается флаг SM100.

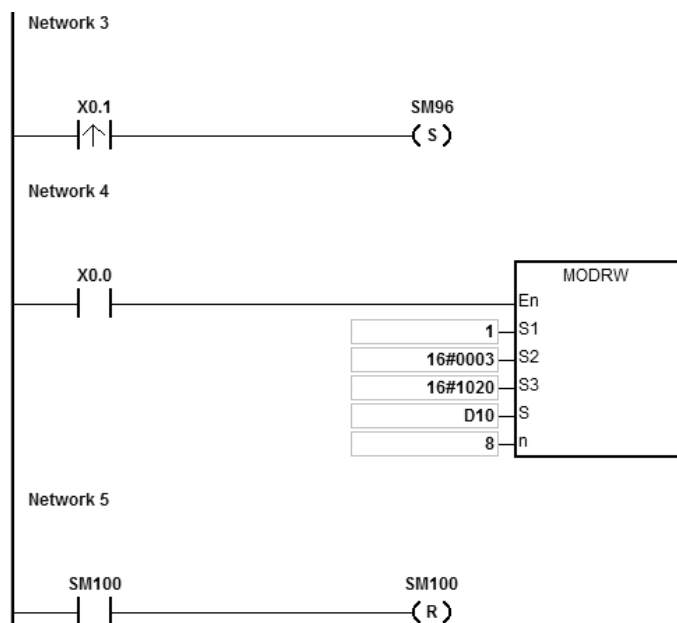
2. Ответное сообщение от DVP-ES2 сохраняется в регистрах от D10.0 до D10.15 (считанные данные из D10.15~D10.0=16#04D2).

<b>Регистр</b>	D10.7	D10.6	D10.5	D10.4	D10.3	D10.2	D10.1	D10.0
<b>Состояние</b>	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ
<b>Значение</b>	D				2			
<b>Регистр</b>	D10.15	D10.14	D10.13	D10.12	D10.11	D10.10	D10.9	D10.8
<b>Состояние</b>	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ
<b>Значение</b>	0				4			

3. По завершении приема данных, отправленных с DVP-ES2, будет подтвержден их формат, и будет определена корректность формата. Если в формате ошибок нет, то будет включен соответствующий флаг SM100, если формат некорректен, то включается флаг SM102.

**Пример 2:**

1. Функциональный код 03 (16#03): ПЛК считывает данные с нескольких битовых регистров, которые не являются дискретными входами (в этом примере считывается 8 фрагментов данных). Для функционального кода 04 операция такая же, как и для функционального кода 03.



2. ЦПУ ПЛК серии AS подключается к ПЛК серии DVP-ES2.  
Когда SM96 и X0.0 включены, ЦПУ AS отправляет и принимает команды от ПЛК DVP-ES2 с помощью регистров D32~D39.
3. Когда адрес D32 равен 16#1020, значения D32~D39 в DVP-ES2 перечислены ниже:

Объект	D32	D33	D34	D35	D36	D37	D38	D39
Значение (16#)	1234	5678	1122	3344	5566	7788	99AA	BBCC

Операнды инструкции MODRW:

Операнд	Описание	Значение
S <sub>1</sub>	Адрес устройства	16#0001
S <sub>2</sub>	Функциональный код	16#0003
S <sub>3</sub>	Адрес данных	16#1020
S	Регистр, используемый при чтении/записи данных	D10
n	Длина данных	8

**Режим ASCII:**

Коды ASCII необязательно преобразовываются намеренно и имеют значения в формате 16#.

- Команда коммуникации для отправления данных: “: 01 03 10 20 00 08 C4 CR LF”.
- Команда коммуникации для приема данных: “: 01 03 10 12 34 56 78 11 22 33 44 55 66 77 88 99 AA BB CC AA CR LF”.

**Режим RTU:**

- Команда коммуникации для отправления данных: “01 03 10 20 00 08 41 06”.
- Команда коммуникации для приема данных: “01 03 10 12 34 56 78 11 22 33 44 55 66 77 88 99 AA BB CC 90 FE”.

Если формат корректен, включается флаг SM100.

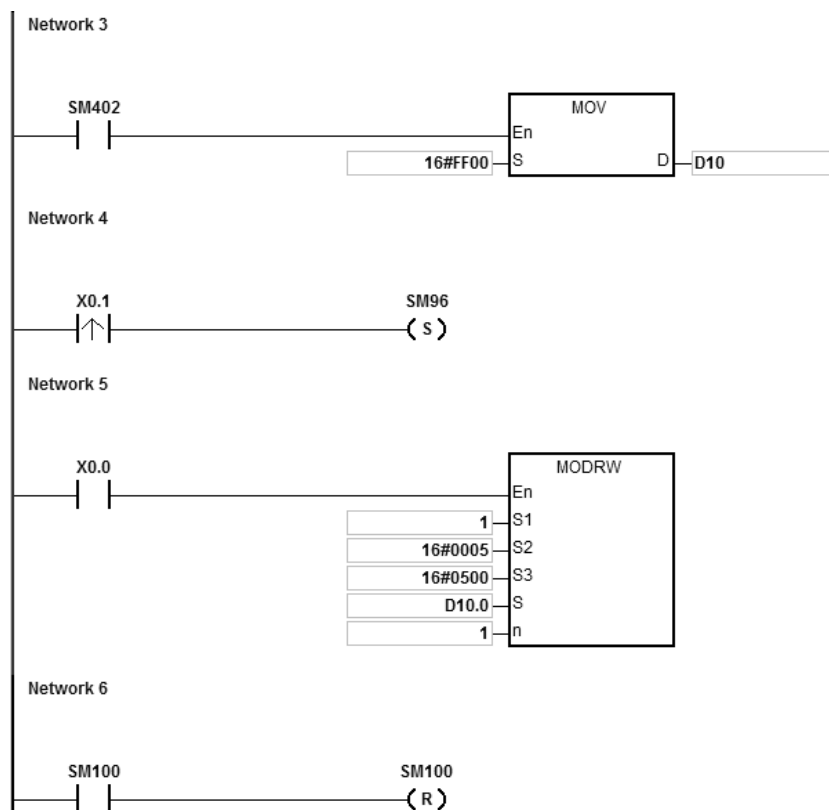
- Ответное сообщение от DVP-ES2 сохраняется в регистрах от D10 до D17.
- Значения в D10–D17:

Объект	D10	D11	D12	D13	D14	D15	D16	D17
Значение (16#)	1234	5678	1122	3344	5566	7788	99AA	BBCC

4. По завершении приема данных, отправленных с DVP-ES2, будет подтвержден их формат, и будет определена корректность формата. Если в формате ошибок нет, то будет включен соответствующий флаг SM100, если формат некорректен, то включается флаг SM102.

**Пример 3:**

1. Функциональный код 05 (16#05): ПЛК записывает состояние в битовый регистр. В этом примере объект включен.



2. ЦПУ ПЛК серии AS подключается к ПЛК серии DVP-ES2. D10.0 включен, выход Y0 в DVP-ES2 также включен. Когда SM96 и X0.0 включены, ПЛК устанавливает состояние Y0.

Операнды инструкции MODRW:

Операнд	Описание	Значение
S <sub>1</sub>	Адрес устройства	1
S <sub>2</sub>	Функциональный код	16#0005
S <sub>3</sub>	Адрес данных	16#0500
S	Регистр, используемый при чтении/записи данных	D10.0
n	Длина данных (здесь не используется)	1

**Режим ASCII:**

Коды ASCII необязательно преобразовываются намеренно и имеют значения в формате 16#.

- Команда коммуникации для отправления данных: “ : 01 05 05 00 FF 00 F6 CR LF”
- Команда коммуникации для приема данных: “ : 01 05 05 00 FF 00 F6 CR LF”



**Режим RTU:**

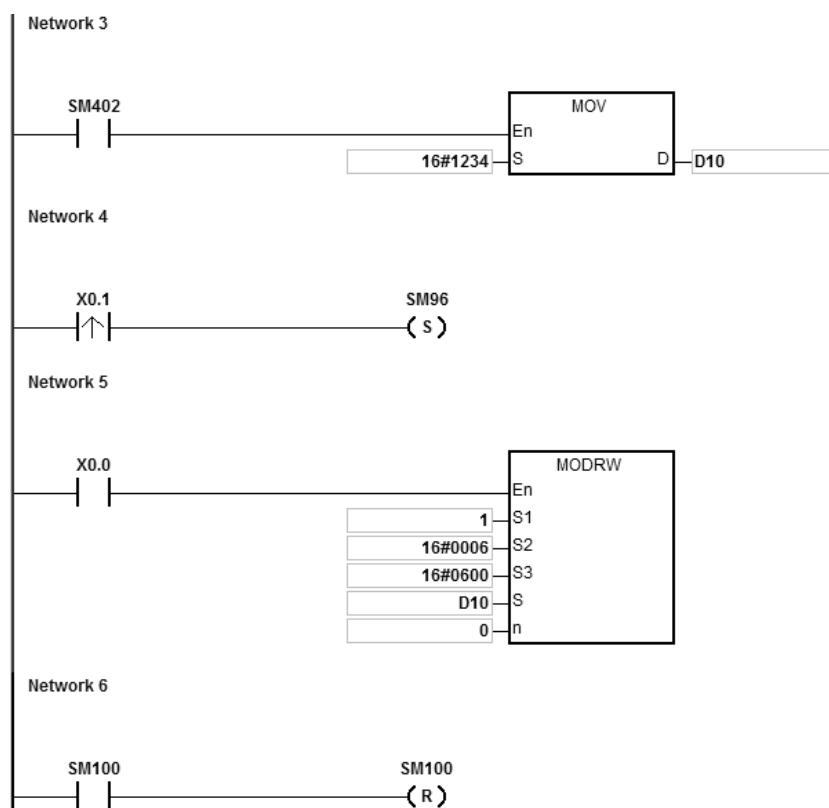
- Команда коммуникации для отправления данных: "01 05 05 00 FF 00 8C F6"
- Команда коммуникации для приема данных: "01 05 05 00 FF 00 8C F6"

Если формат корректен, включается флаг SM100.

3. По завершении приема данных, отправленных с DVP-ES2, будет подтвержден их формат, и будет определена корректность формата. Если в формате ошибок нет, то будет включен соответствующий флаг SM100, если формат некорректен, то включается флаг SM102.
4. Когда DVP-ES2 получает эту инструкцию, включается выход Y0.
5. 5. Параметр n здесь не используется, так как этот код функции здесь предназначен для записи.

## Пример 4:

1. Функциональный код (16#06): ПЛК записывает состояние в один словный регистр.



2. ЦПУ ПЛК серии AS подключается к ПЛК серии DVP-ES2.
3. Предположим, D10 равно 16#55AA (ожидание записи данных в регистр T0 в DVP-ES2).

Когда SM96 и X0.0 включены, ПЛК записывает данные в T0 ПЛК DVP-ES2. Адрес T0 равен 16#0600.

Операнды инструкции MODRW:

Операнд	Описание	Значение
S <sub>1</sub>	Адрес устройства	1
S <sub>2</sub>	Функциональный код	16#0006
S <sub>3</sub>	Адрес данных	16#0600
S	Регистр, используемый при чтении/записи данных	D10
n	Длина данных (здесь не используется)	0

**Режим ASCII:**

Коды ASCII необязательно преобразовываются намеренно и имеют значения в формате 16#.

- Команда коммуникации для отправления данных: “ : 01 06 06 00 55 AA F4 CR LF”
- Команда коммуникации для приема данных: “ : 01 06 06 00 55 AA F4 CR LF”

**RTU mode:**

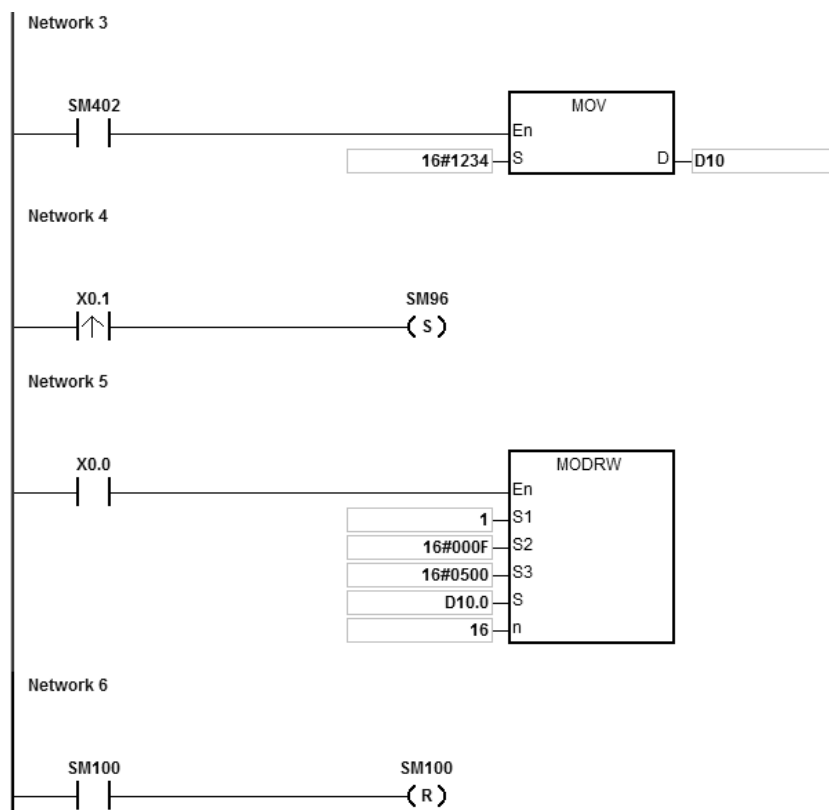
- Команда коммуникации для отправления данных: “01 06 06 00 55 AA 36 6D”
- Команда коммуникации для приема данных: “01 06 06 00 55 AA 36 6D”

Если формат корректен, включается флаг SM100.

4. По завершении приема данных, отправленных с DVP-ES2, будет подтвержден их формат, и будет определена корректность формата. Если в формате ошибок нет, то будет включен соответствующий флаг SM100, если формат некорректен, то включается флаг SM102.
5. Когда DVP-ES2 получает эту инструкцию, данные сохраняются в D10 для регистра T0 ПЛК DVP-ES2.
6. Параметр n здесь не используется, так как этот код функции здесь предназначен для записи.

**Пример 5:**

1. Функциональный код 0F (16#0F): ПЛК записывает состояния в несколько битовых регистров.



2. ЦПУ ПЛК серии AS подключается к ПЛК серии DVP-ES2.

- Предположим, D10.15~D10.0=16#04D2 (ожидание записи состояния выходов Y0~Y17 ПЛК DVP-ES2)

<b>Регистр</b>	D10.7	D10.6	D10.5	D10.4	D10.3	D10.2	D10.1	D10.0
<b>Состояние</b>	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ
<b>Значение</b>	D				2			
<b>Регистр</b>	D10.15	D10.14	D10.13	D10.12	D10.11	D10.10	D10.9	D10.8
<b>Состояние</b>	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ
<b>Значение</b>	0				4			

Когда SM96 и X0.0 включены, ПЛК устанавливает состояние выходов Y0~Y17 для DVP-ES2. Адрес Y0: 16#0500.

Операнды инструкции MODRW:

Операнд	Описание	Значение
S <sub>1</sub>	Адрес устройства	1
S <sub>2</sub>	Функциональный код	16#000F
S <sub>3</sub>	Адрес регистра Y0	16#0500
S	Регистры Y0...Y17, используемые при чтении/записи данных	D10.0
n	Длина данных	16

**Режим ASCII:**

Коды ASCII необязательно преобразовываются намеренно и имеют значения в формате 16#.

- Команда коммуникации для отправления данных: " : 01 0F 0500 0010 02 D2 04 03 CR LF"
- Команда коммуникации для приема данных: " : 01 0F A0 00 00 10 40 CR LF"

**Режим RTU:**

- Команда коммуникации для отправления данных: "01 0F 05 00 00 10 02 D2 04 EA 43"
- Команда коммуникации для приема данных: "01 0F A0 00 00 10 76 07"

Если формат корректен, включается флаг SM100.

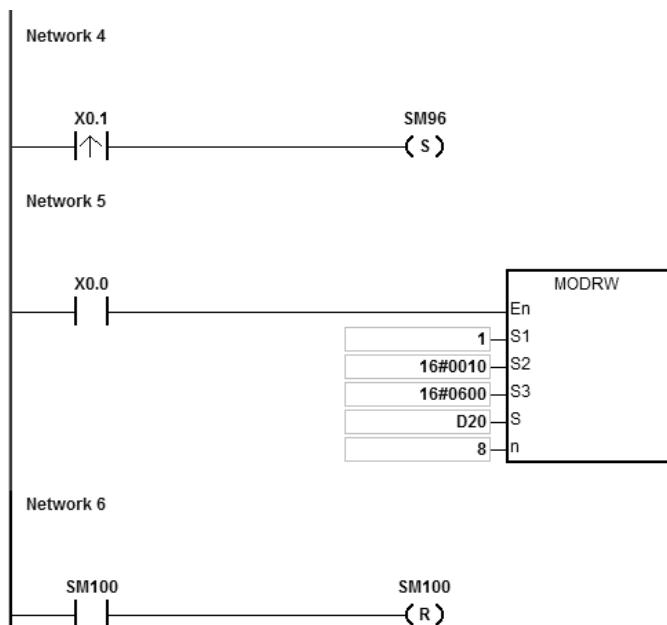
- 6
- По завершении приема данных, отправленных с DVP-ES2, будет подтвержден их формат, и будет определена корректность формата. Если в формате ошибок нет, то будет включен соответствующий флаг SM100, если формат некорректен, то включается флаг SM102.

<b>Устройство</b>	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
<b>Состояние</b>	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ
<b>Значение</b>	D				2			
<b>Устройство</b>	Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10
<b>Состояние</b>	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	ВЫКЛ
<b>Значение</b>	0				4			

- Параметр n здесь не используется, так как этот код функции здесь предназначен для записи.

**Пример 6:**

1. Функциональный код 10 (16#10): ПЛК записывает состояние в несколько словных регистров.



2. ЦПУ ПЛК серии AS подключается к ПЛК серии DVP-ES2.
3. Предположим, значения D20~27 приведены ниже (ожидание записи данных в T0~7 ПЛК DVP-ES2).

Объект	D20	D21	D22	D23	D24	D25	D26	D27
Значение (16#)	1234	5678	1122	3344	5566	7788	99AA	BBCC

Когда SM96 и X0.0 включены, ПЛК записывает данные в T0~7 ПЛК DVP-ES2. Адрес T0: 16#0600.

Операнды инструкции MODRW:

Операнд	Описание	Значение
S <sub>1</sub>	Адрес устройства	1
S <sub>2</sub>	Функциональный код	16#0010
S <sub>3</sub>	Адрес регистра T0	16#0600
S	Регистры T0...T17, используемые при чтении/записи данных	D20
n	Длина данных (здесь не используется)	8

**Режим ASCII:**

Коды ASCII необязательно преобразовываются намеренно и имеют значения в формате 16#.

- Команда коммуникации для отправления данных: “ : 01 10 0600 00 08 10 1234 5678 1122 3344 5566 7788 99AA BBCC 8F CR LF”
- Команда коммуникации для приема данных: “ : 01 10 06 00 00 08 E1 CR LF”

**Режим RTU:**

- Команда коммуникации для отправления данных: “01 10 06 00 00 08 10 1234 5678 1122 3344 5566 7788 99AA BBCC 0B 0C”
- Команда коммуникации для приема данных: “01 10 06 00 00 08 C1 47”

Если формат корректен, включается флаг SM100.

5. По завершении приема данных, отправленных с DVP-ES2, будет подтвержден их формат, и будет определена корректность формата. Если в формате ошибок нет, то будет включен соответствующий флаг SM100, если формат некорректен, то включается флаг SM102. Когда DVP-ES2 примет данную инструкцию, он сохранит данные в D20~27 для регистров T0~7 ПЛК DVP-ES2.

Объект	T0	T1	T2	T3	T4	T5	T6	T7
Значение (16#)	1234	5678	1122	3344	5566	7788	99AA	BBCC

6. Параметр n здесь не используется, так как этот код функции здесь предназначен для записи.

**Примечание:**

1. Если значение **S<sub>1</sub>** или **S<sub>2</sub>** выходит за пределы диапазона, возникает ошибка, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если операнд **S** недостаточен для содержания n частей данных, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
3. Если значение n выходит за пределы диапазона, возникает ошибка, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#200B.
4. Если функциональный код в **S<sub>2</sub>** связан с битовым регистром, то указанный в **S** регистр, должен быть битовым. В противном случае, возникает ошибка, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
5. Если функциональный код в **S<sub>2</sub>** связан со словным регистром, то указанный в **S** регистр, должен быть словным. В противном случае, возникает ошибка, инструкция не выполняется, включается флаг SM0 и в

SR0 записывается код ошибки 16#2003.

6. Если команда коммуникации равна 0x05 или 0x06, **n** не работает. Состояние или данные записываются в один битовый или один словный регистр.
7. Если флаги передачи SM96 и SM97 не включены, инструкция MODRW не выполняется.
8. Если при коммуникации возникает пауза, флаги задержки SM104 и SM105 включены, а флаги приема SM98 и SM99 отключены.
9. Если ошибка возникает при приеме данных, включены флаги ошибки SM102 и SM103, а флаги приема SM98 и SM99 отключены.
10. Если функциональный код в **S<sub>2</sub>** связан с словным объектом, объект во внешнем оборудовании, с которым осуществляет связь ПЛК, должен быть также словным. Если функциональный код в **S<sub>2</sub>** связан с битовым объектом, объект во внешнем оборудовании, с которым осуществляет связь ПЛК, должен быть также битовым.
11. Дополнительную информацию по регистрам коммуникации (SM, SR) см. Раздел 6.19.3.



API	Код инструкции			Операнды						Функция							
1812		COMRS		$S_1 \cdot S_2 \cdot S_3 \cdot D_1 \cdot D_2$						Автоопределение параметров отправки и приема данных коммуникации							

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●							
S <sub>3</sub>								●	●				○	○		
D <sub>1</sub>								●								
D <sub>2</sub>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
D <sub>1</sub>		●			●	●							
D <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**

COMRS	
En	
S1	D1
S2	D2
S3	

- S<sub>1</sub> : Номер коммуникационного порта (1~2, 11~12)
- S<sub>2</sub> : Источник отправляемых данных
- S<sub>3</sub> : Длина отправляемых данных
- D<sub>1</sub> : Начальный операнд сохранения данных коммуникации
- D<sub>2</sub> : Условие прекращения приема данных

**Описание:**

1. S<sub>1</sub> – номер коммуникационного порта (COM1: номер 1, COM2: номер 2, Card1: номер 11 Card2: номер 12). Если данные выходят за пределы диапазона для портов связи, инструкция не будет выполнять отправку или получение данных.
2. Если определенный символ или определенные символы используются в качестве условия прекращения приема данных, подразумевается, что инструкция работает с ASCII данными. Если инструкция использует не ASCII данные, то условием прекращения приема данных будет пауза в коммуникации.
3. S<sub>2</sub>: Источник отправляемых данных

**S<sub>3</sub>**: Длина отправляемых данных

Если значение в **S<sub>2</sub>** равно D100 и значение в **S<sub>3</sub>** равно 10, значения в младших байтах в D100~D109 будут передаваться через коммуникационный порт, заданный в **S<sub>1</sub>**.

4. Если заданное значение **S<sub>3</sub>** равно 0, данные не будут отправляться. Максимальная длина данных для передачи – 256 слов.

5. **D<sub>1</sub>**: Относится к длине принятых данных.

**D<sub>1+1</sub>~D<sub>1+n</sub>**: Операнды сохранения принятых данных.

Если значение в **D<sub>1</sub>** равно D200, значение в **D<sub>2</sub>** равно 3 и значение в **D<sub>2+1</sub>** равно 16#0D0A, принятые данные будут сохранены в младших байтах в регистрах, начиная с D201 (старшие байты не будут изменяться), прием данных продолжится до тех пор, пока не будут получены последовательные символы прекращения приема 16#0D и 16#0A, после приема 16#0D и 16#0A длина принятых данных будет сохранена в D200, а после прекращения приема данных включится флаг завершения.

6. **D<sub>2</sub>**: Режим приема данных

**D<sub>2+1</sub>**: Условие прекращения приема данных

Настройки **D<sub>2</sub>** и **D<sub>2+1</sub>** показаны ниже.

<b>D<sub>2</sub></b>	<b>Режим приема данных</b>	<b>Заданное значение D<sub>2+1</sub></b>	<b>Примечание</b>
<b>0</b>	Приема данных коммуникации нет	Не используется	После окончания отправки данных включается флаг выполнения.
<b>1</b>	Если пауза перед приемом следующего пакета данных превышает задание в <b>D<sub>2+1</sub></b> , прием данных считается выполненным.	Заданное значение в <b>D<sub>2+1</sub></b> – время. Ед. изм. – мс. Задание в <b>D<sub>2+1</sub></b> должно быть в диапазоне 5...3000.	Если задание превышает 3000 мс, значение в <b>D<sub>2+1</sub></b> будет равно 3000. Если задание меньше 5 мс, значение в <b>D<sub>2+1</sub></b> будет равно 5.
<b>2</b>	Условием прекращения приема данных является определенный символ.	Заданное значение в <b>D<sub>2+1</sub></b> – определенный символ.	Если определенный символ равен 16#0A, значение в <b>D<sub>2+1</sub></b> будет равно 16#000A.
<b>3</b>	Условием прекращения приема данных являются два определенных символа.	Заданное значение в <b>D<sub>2+1</sub></b> – два определенных символа	Если определенными символами являются 16#0D и 16#0A, значение в <b>D<sub>2+1</sub></b> будет равно 16#0D0A.
<b>4</b>	Прием данных начинается с определенного символа. Если пауза перед приемом	Определенный символ сохраняется в старшем байте <b>D<sub>2+1</sub></b> , время	Если определенный символ равен 16#3A, а время 15 мс, значение в <b>D<sub>2+1</sub></b> будет равно 16#3A0F.

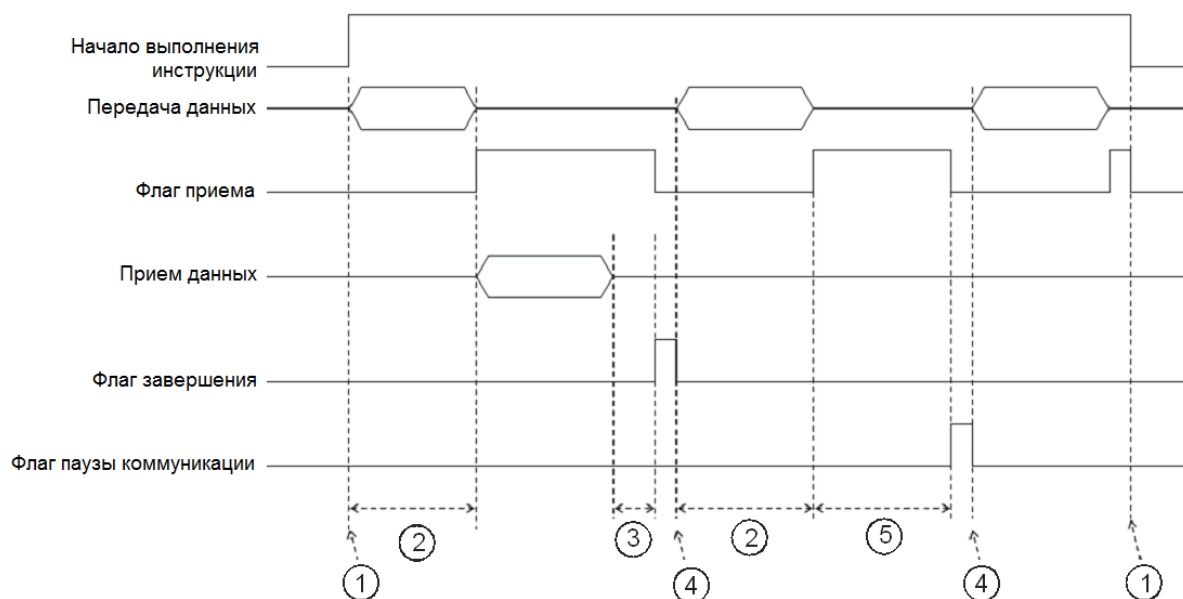
	следующего пакета данных превышает задание в $D_{2+1}$ , прием данных считается выполненным.	сохраняется в младшем байте $D_{2+1}$ . (Диапазон задания времени – от 5 до 255 мс).	
5	Прием данных начинается с определенного символа и условием прекращения приема данных является определенный символ.	Заданное значение в $D_{2+1}$ – два определенных символа – стартовый и конечный	Если стартовый определенный символ 16#3A и конечный определенный символ 16#0A, значение в $D_{2+1}$ будет равно 16#3A0A.
6	Получение определенного количества данных.	Заданное значение в $D_{2+1}$ это длина принятых данных. Диапазон 1~256.	Если необходимо принять 10 символов, значение в $D_{2+1}$ = 10.
7	Условием прекращения приема данных является определенный символ и после генерируется прерывание коммуникации.	Заданное значение в $D_{2+1}$ – определенный символ.	Если конечный символ равен 16#0A, значение в $D_{2+1}$ будет равно 16#000A.
8	Получение определенного количества данных и генерирования прерывания коммуникации.	Заданное значение в $D_{2+1}$ – длина принятых данных. Диапазон 1~256.	Если необходимо принять 10 символов, значение в $D_{2+1}$ = 10.
9	Полученные данные заканчиваются определенным символом или определенным количеством полученных данных; когда выполняется любое условие, передача завершена.	Определенный символ сохраняется в старшем байте $D_{2+1}$ , время сохраняется в младшем байте $D_{2+1}$ . (диапазон задания времени от 1 до 255 мс)	Если конечный символ равен 16#0A, а время 15 мс, длина данных 15 слов, значение в $D_{2+1}$ будет равно 16#0A0F.
Другие	Если используется не поддерживаемый режим, инструкция выполняться не будет.		

7. За исключением режимов 6 и 8, когда принятые данные в  $D_2$  превышают максимальные 256 слов и нет определенного конечного символа, инструкция прекращает свою работу и будет сообщено об ошибке приема.  $D_{1+0}$  равно 0 и  $D_{1+1}$  не входят в полученные данные.
8. Связь между коммуникационным портом, связанными с ним вспомогательными реле и связанным с ним специальным регистром данных описана в Разделе 6.19.3.
9. Временная диаграмма

- Режим приема данных: 0

Когда данные отправляются, пользователь не может отменить отправку данных. Если условный контакт, предшествующий инструкции, не активирован, данные все равно будут отправляться, но после выполнения отправки данных флаг завершения не будет включен.

- Режим приема данных: 1 or 4



Описание:

① → Запуск / останов выполнения инструкции пользователем.

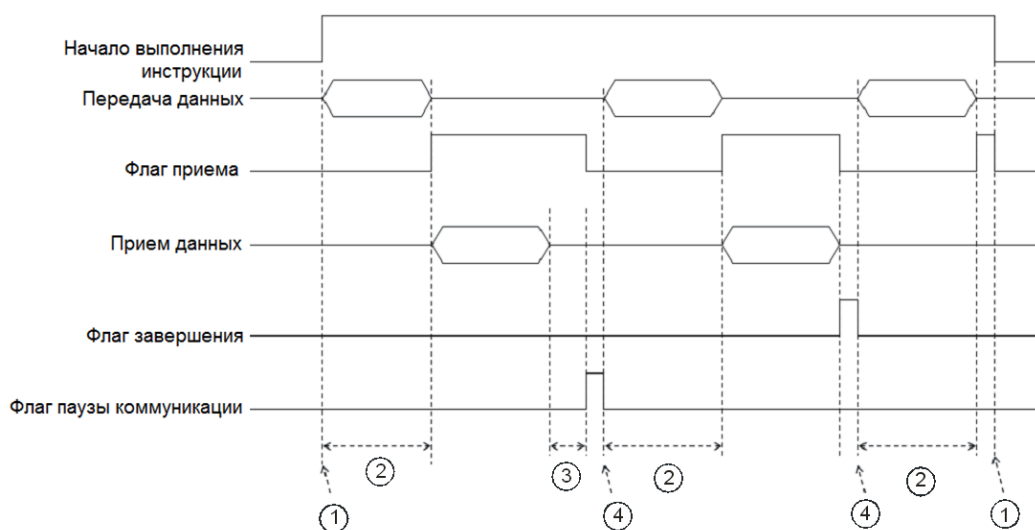
② → Время отправки данных

Период времени, в который данные отправляются, не измеряется.

③ → После приема первого символа будет измерено время, прошедшее до приема следующего символа. Всякий раз, когда символ получен, измеренное время очищается. Флаг завершения не включается до тех пор, пока измеренное время не будет больше заданного значения в  $D_{2+1}$ .

- ④ → Если инструкция по-прежнему выполняется после сброса пользователем флага завершения или флага коммуникации, последующие данные коммуникации отправляются автоматически, когда инструкция сканируется в следующем цикле.
- ⑤ → Когда ПЛК начинает принимать данные, он измеряет время приема данных. Пауза коммуникации не устанавливается до тех пор, пока измеренное время не превысит заданный период ожидания. Предполагается, что заданное время паузы коммуникации должно быть больше времени, заданного в  $D_2+1$ .

- Режим приема данных: 2, 3, 5, 6, от 9.



Описание:

- ① → Запуск / останов выполнения инструкции пользователем.
- ② → Время отправки данных  
Период времени, в который данные отправляются, не измеряется.
- ③ → После приема первого символа будет измерено время, прошедшее до приема следующего символа. Всякий раз, когда символ получен, измеренное время очищается. Флаг завершения не включается до тех пор, пока измеренное время не будет больше заданного значения в  $D_2+1$ .
- ④ → Если инструкция по-прежнему выполняется после сброса пользователем флага завершения или флага коммуникации, последующие данные коммуникации отправляются автоматически, когда инструкция сканируется в следующем цикле.

10. Режим отправки данных / режим приема данных

8-битный режим: Редактируемая команда сохраняется в исходном устройстве, а отправляемая команда, включает в себя начальный и конечный коды. 16-битные данные делятся на старшие и младшие 8-битные



11. Для режимов 7 и 8 соответствующий коммуникационный порт и номер прерывателя перечислены ниже.

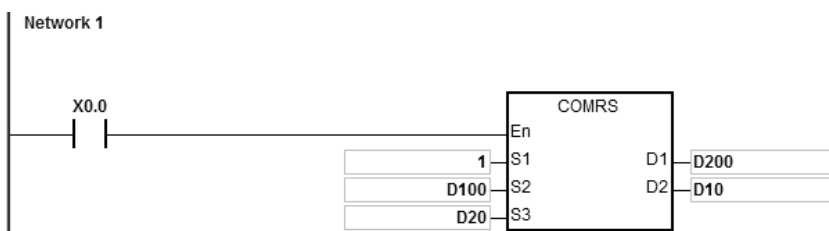
Коммуникационный порт	COM1	COM2	Плата1	Плата2
Номер прерывателя	I300	I302	I304	I306

Эта инструкция не может использоваться в языке программирования ST, задачах прерывания или назначенных функциональных блоках,

В качестве примеров используется порт COM1 (RS485).

**Пример 1:**

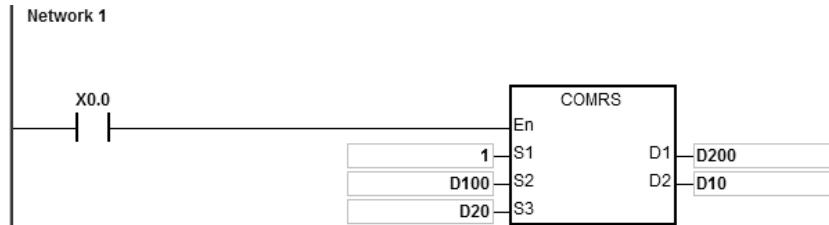
Режим в **D<sub>2</sub>** задан как 0 (приема данных коммуникации нет) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



1. Длина данных для передачи: D20=4.
2. Содержимое данных для передачи: D100=16#0031, D101=16#0032, D102=16#0033, D103=16#0034.
3. Задание D10=16#0000 (только отправка данных, без приема).
4. Активация контакта X0.0.
5. ПЛК передает 4 фрагмента данных.
6. Передача данных: ПЛК→Внешнее устройство 31 32 33 34.
7. Поскольку прием данных не требуется, после того, как ПЛК отправит все данные, операция завершается. SM100 = 0.
8. Для другой передачи данных пользователь может выключить флаг SM100, операция передачи запустится снова.

**Пример 2:**

Режим в  $D_2$  задан как 1 (задание времени в диапазоне 5~3000 мс) и задание режима отправки данных / режима приема данных – 16-битные данные (флаг SM106 выключен).



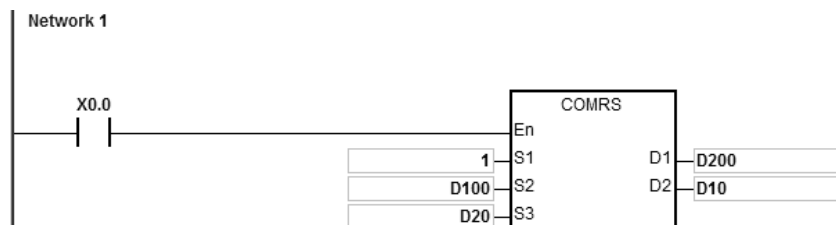
1. Длина данных для передачи:  $D20=4$ .
2. Содержимое данных для передачи:  $D100=16\#3231$ ,  $D101=16\#3433$ .
3. Задание  $D10=16\#0001$  (режим: 1),  $D11=300$  (задание времени 300 мс).
4. Активация контакта X0.0.
5. ПЛК передает 4 фрагмента данных.
6. Передача данных: ПЛК→Внешнее устройство 31 32 33 34.
7. После того как внешнее оборудование получит данные от ПЛК, 5 последовательных фрагментов данных будут отправлены в ПЛК, и каждое сообщение будет отправляться менее 20 мс. Внешнее оборудование 35 36 37 38 39.
8.  $D200=5$  (число фрагментов принятых данных), содержимое принятых данных:  $D201=16\#3635$ ,  $D202=16\#3837$ ,  $D203=16\#0039$ .
9. SM100 включен: прием данных выполнен.
10. Для другой передачи данных пользователь может выключить флаг SM100, операция передачи запустится снова.

**ПРИМЕЧАНИЕ:** Когда передача данных будет завершена, будет включен флаг приема SM98, а затем ПЛК начнет принимать данные. Интервал времени приема каждого фрагмента данных устанавливается в D11. Когда время приема превышает заданное время, и данные не поступают, SM100 будет включен.



**Пример 3:**

Режим в D<sub>2</sub> задан как 2 (условием прекращения приема данных является определенный символ) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



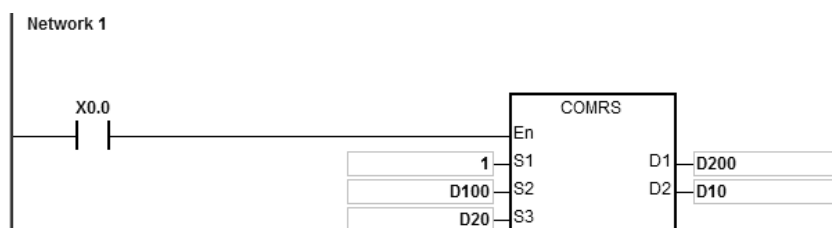
1. Длина данных для передачи: D20=0, ПЛК не будет отправлять данные, а только получает их.
2. Задайте D10=16#0002 (режим: 2), D11=16#000A (конечный определенный символ равен 16#0A).
3. Включите контакт X0.0.
4. ПЛК ждет приема данных от внешнего оборудования(D20 = 0, что означает, что ПЛК не будет отправлять данные на внешнее оборудование).
5. Внешнее оборудование отправляет данные в ПЛК. Внешнее оборудование → ПЛК 31 32 33 34 35 0A.
6. D200=6 (количество принятых данных), содержимое принятых данных: D201=16#0031, D202=16#0032, D203=16#0033, D204=16#0034, D205=16#0035, D206=16#000A
7. SM100 включен: прием данных выполнен.
8. Для другой передачи данных пользователь может выключить флаг SM100, операция передачи запустится снова.

**ПРИМЕЧАНИЕ:** Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.

**Пример 4:**

Режим в  $D_2$  задан как 3 (условием прекращения приема данных являются два определенных символа) и задание режима отправки данных / режима приема данных – 16-битные данные (флаг SM106 выключен).

В этом примере DVP-ES2 используется как внешнее оборудование, и мы собираемся записать значение N1234 в регистр D100 в DVP-ES2.



1. Длина данных для передачи:  $D20=17$ .
2. Содержимое передаваемых данных:  $D100=16\#303A$ ,  $D101=16\#3031$ ,  $D102=16\#3136$ ,  $D103=16\#3630$ ,  $D104=16\#3134$ ,  $D105=16\#3332$ ,  $D106=16\#3334$ ,  $D107=16\#0D46$ ,  $D108=16\#000A$ .
3. Задайте  $D10=16\#0003$  (режим: 3),  $D11=16\#0D0A$  (конечные определенные символы равны  $16\#0D$  и  $16\#0A$ ).
4. Включите контакт X0.0
5. ПЛК отправляет 17 фрагментов данных.

Отправляемые данные: ПЛК → Внешнее устройство 3A 30 31 30 36 31 30 36 34 31 32 33 34 33 46 0D 0A

(код ASCII: 0106106412343FCRLF)

6. Внешнее оборудование получает данные от ПЛК, а конечные определенные символы –  $16\#0D$  и  $16\#0A$ .

Внешнее оборудование → ПЛК 3A 30 31 30 36 31 30 36 34 31 32 33 34 33 46 0D 0A

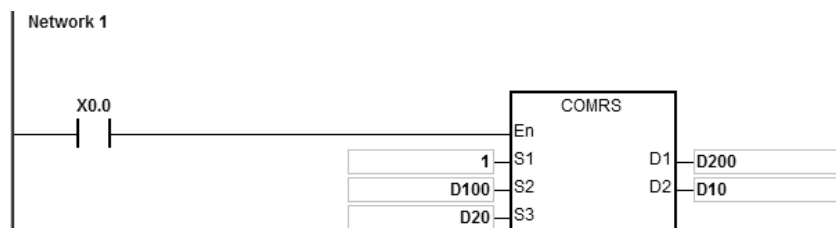
(код ASCII: 0106106412343FCRLF)

7.  $D200=17$  (количество принятых данных), содержимое принятых данных:  $D201=16\#303A$ ,  $D202=16\#3031$ ,  $D203=16\#3136$ ,  $D204=16\#3630$ ,  $D205=16\#3134$ ,  $D206=16\#3332$ ,  $D207=16\#3334$ ,  $D208=16\#0D46$ ,  $D209=16\#000A$ .
8. SM100 включен: прием данных выполнен.
9. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

ПРИМЕЧАНИЕ: Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.

**Пример 5:**

Режим в D<sub>2</sub> задан как 4 (получение данных начинается с определенного символа и значение времени приема задается в диапазоне 5 ~ 255 мс) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



1. Длина данных для передачи: D20=4.
2. Содержимое данных для передачи: D100=16#0031, D101=16#0032, D102=16#0033, D103=16#0034.
3. Задайте D10=16#0004 (режим: 4), D11=16#3A0F (стартовый символ – 16#3A и задание значения времени - 16#0F, что означает 15 мс).
4. Включите контакт X0.0.
5. ПЛК передает 4 фрагмента данных.

Передаваемые данные: ПЛК→Внешнее устройство 31 32 33 34

6. Внешнее оборудование получает данные от ПЛК и затем отправляет 7 последовательных слов в ПЛК с интервалом в 1 мс между каждой отправкой.

Внешнее оборудование → ПЛК 30 3A 35 36 37 38 39

7. D200=6 (число принятых данных), содержимое принятых данных: D201=16#003A, D202=16#0035, D203=16#0036, D204=16#0037, D205=16#0038, D206=16#0039.

8. SM100 включен: прием данных выполнен.

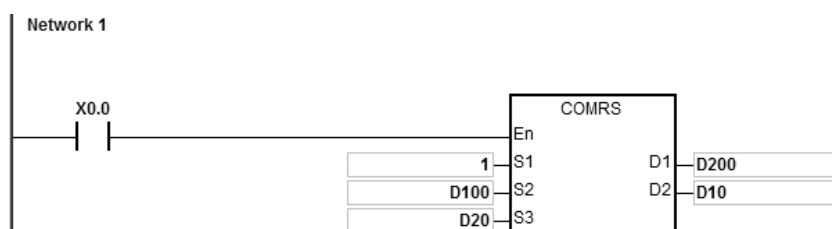
9. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

**ПРИМЕЧАНИЕ:** Когда передача данных завершена, будет включен флаг приема SM98, а ПЛК будет готов к приему данных. При получении стартового символа 16#3A ПЛК начнет принимать данные. Интервал времени каждого приема данных задается в D11. Когда интервал времени между приемами превышает заданное время 16#0F (15 мс), будет включен флаг SM100.

**Пример 6:**

Режим в  $D_2$  задан как 5 (Прием данных начинается с определенного символа и условием прекращения приема данных является определенный символ) и задание режима отправки данных / режима приема данных – 16-битные данные (флаг SM106 выключен).

В этом примере DVP-ES2 используется как внешнее оборудование, и мы собираемся записать данные в регистр D100 в DVP-ES2.

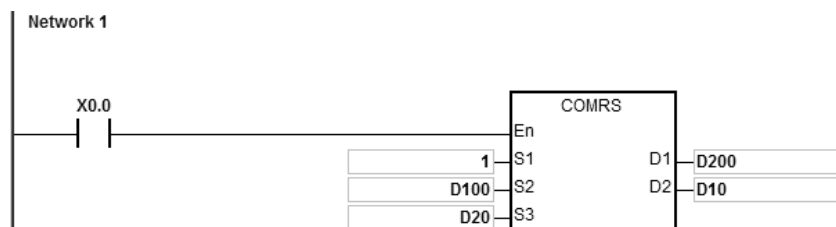


1. Длина данных для передачи:  $D20=17$ .
2. Содержимое данных для передачи:  $D100=16\#303A$ ,  $D101=16\#3031$ ,  $D102=16\#3133$ ,  $D103=16\#3630$ ,  $D104=16\#3034$ ,  $D105=16\#3030$ ,  $D106=16\#3831$ ,  $D107=16\#0D37$ ,  $D108=16\#000A$
3. Задайте  $D10=16\#0005$  (режим: 5),  $D11=16\#3A0A$  (стартовый символ  $16\#3A$  и конечный символ  $16\#0A$ ).
4. Включите контакт X0.0.
5. ПЛК передает 17 фрагментов данных.  
Отправляемые данные: ПЛК→Внешнее оборудование 3A 30 31 30 36 31 30 36 34 31 32 33 34 33 46 0D 0A  
(код ASCII: 0106106412343FCRLF)
6. Внешнее оборудование получает данные от ПЛК, а последние 2 фрагмента данных  $16\#0D$  и  $16\#0A$ .  
Внешнее оборудование → ПЛК 3A 30 31 30 36 31 30 36 34 31 32 33 34 33 46 0D 0A  
(код ASCII: 0106106412343FCRLF)
7.  $D200=15$  (число принятых данных), содержимое принятых данных:  $D201=16\#303A$ ,  $D202=16\#3031$ ,  $D203=16\#3033$ ,  $D204=16\#3132$ ,  $D205=16\#3332$ ,  $D206=16\#4234$ ,  $D207=16\#0D34$ ,  $D208=16\#000A$ .
8. SM100 включен: прием данных выполнен.
9. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

**ПРИМЕЧАНИЕ:** Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.

**Пример 7:**

Режим в D<sub>2</sub> задан как 6 (Получение определенного количества данных) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



1. Длина данных для передачи: D20=4.
2. Содержимое данных для передачи: D100=16#0031, D101=16#0032, D102=16#0033, D103=16#0034.
3. Задайте D10=16#0006 (режим: 6), D11=16#0008 (8 фрагментов данных для приема).

4. Включите контакт X0.0.

5. ПЛК передает 4 фрагмента данных.

Передаваемые данные: ПЛК→Внешнее устройство 31 32 33 34

6. Внешнее устройство принимает данные от ПЛК и затем отправляет 8 последовательных данных в ПЛК.

Внешнее устройство → ПЛК 32 33 34 35 36 37 38 39

7. D200=8 (число принятых данных), содержимое принятых данных: D201=16#0032, D202=16#0033, D203=16#0034, D204=16#0035, D205=16#0036, D206=16#0037, D207=16#0038, D208=16#0039.

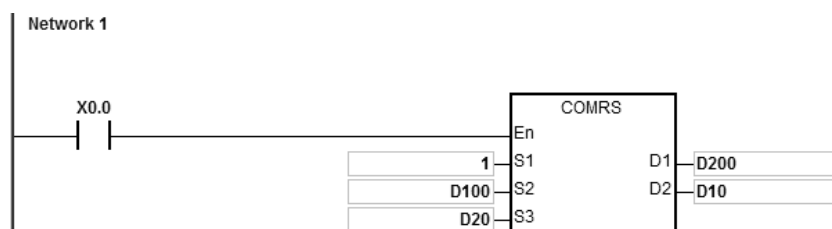
8. SM100 включен: прием данных выполнен.

9. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

**ПРИМЕЧАНИЕ:** Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.

**Пример 8:**

Режим в  $D_2$  задан как 7 (Условием прекращения приема данных является определенный символ и после генерируется прерывание коммуникации) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



Программа прерывания коммуникации:

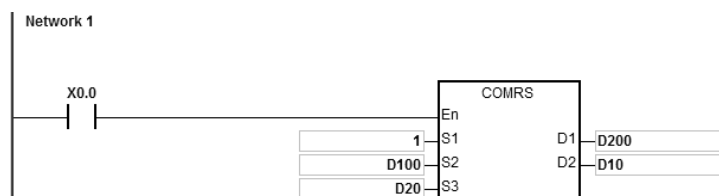


1. Сброс прерывания:  $D30=0$
2. Длина данных для передачи:  $D20=4$ .
3. Содержимое данных для передачи:  $D100=16\#0031$ ,  $D101=16\#0032$ ,  $D102=16\#0033$ ,  $D103=16\#0034$ .
4. Задайте  $D10=16\#0007$  (режим: 7),  $D11=16\#000A$  ( $16\#0A$  – конечный символ).
5. Включите контакт  $X0.0$ .
6. ПЛК передает 4 фрагмента данных. Передаваемые данные: ПЛК→Внешнее устройство 31 32 33 34
7.  $D30=0$  (программа прерывания не выполняется)
8. Внешнее устройство передает данные в ПЛК. Внешнее устройство → ПЛК 31 32 33 34 35 0A
9.  $D200=6$  (число принятых данных), содержимое принятых данных:  $D201=16\#0031$ ,  $D202=16\#0032$ ,  $D203=16\#0033$ ,  $D204=16\#0034$ ,  $D205=16\#0035$ ,  $D206=16\#000A$ .
10. SM100 включен: прием данных выполнен.
11.  $D30=1$  (прерывание запускается, а затем выполняется INC D30)
12. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

ПРИМЕЧАНИЕ: Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.

**Пример 9:**

Режим в D<sub>2</sub> задан как 8 (Получение определенного количества данных и генерирование прерывания коммуникации) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



Программа прерывания коммуникации:

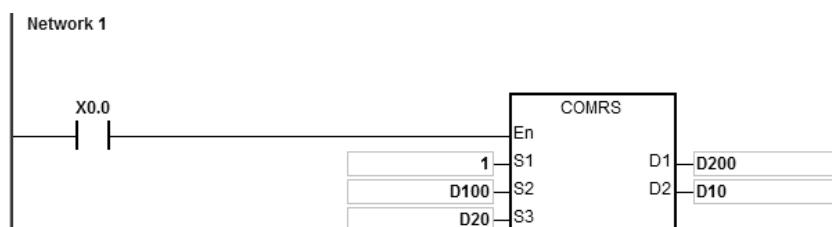


1. Сброс прерывания: D30=0
2. Длина данных для передачи: D20=4.
3. Содержимое данных для передачи: D100=16#0031, D101=16#0032, D102=16#0033, D103=16#0034.
4. Задайте D10=16#0008 (режим: 8), D11=16#0008 (8 фрагментов данных для приема).
5. Включите контакт X0.0.
6. ПЛК передает 4 фрагмента данных. Передаваемые данные: ПЛК→Внешнее устройство 31 32 33 34
7. D30=0 (программа прерывания не выполняется)
8. Внешнее устройство принимает данные от ПЛК и передает в ПЛК 8 последовательных фрагментов данных.  
Внешнее устройство → ПЛК 32 33 34 35 36 37 38 39
9. D200=8 (число принятых данных), содержимое принятых данных: D201=16#0032, D202=16#0033, D203=16#0034, D204=16#0035, D205=16#0036, D206=16#0037, D207=16#0038, D208=16#0039.
10. SM100 включен: прием данных выполнен.
11. D30=1 (прерывание запускается, а затем выполняется INC D30)
12. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

ПРИМЕЧАНИЕ: Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.

**Пример 10:**

Режим в  $D_2$  задан как 9 (Полученные данные заканчиваются определенным символом или определенным количеством полученных данных; когда выполняется любое условие, передача завершена) и задание режима отправки данных / режима приема данных – 8-битные данные (флаг SM106 включен).



1. Длина данных для передачи:  $D20=4$ .
2. Содержимое данных для передачи:  $D100=16\#0031$ ,  $D101=16\#0032$ ,  $D102=16\#0033$ ,  $D103=16\#0034$ .
3. Задайте  $D10=16\#0009$  (режим: 9),  $D11=16\#0A0F$  (конечный символ -  $16\#0A$  и длина данных -  $16\#0F$ ).
4. Включите контакт X0.0.
5. ПЛК передает 4 фрагмента данных. Передаваемые данные: ПЛК→Внешнее устройство 31 32 33 34
6. Внешнее устройство принимает данные от ПЛК и передает в ПЛК 15 последовательных фрагментов данных.  
Внешнее устройство → ПЛК 31 32 33 34 35 0A 41 42 43 44 45 46 47 48 49
7.  $D200=6$  (число принятых данных), содержимое принятых данных:  $D201=16\#0031$ ,  $D202=16\#0032$ ,  $D203=16\#0033$ ,  $D204=16\#0034$ ,  $D205=16\#0035$ ,  $D206=16\#000A$ .  
ПЛК прекращает прием данных после получения 6-го фрагмента данных.
8. SM100 включен: прием данных выполнен.
9. Для другой передачи данных пользователь может выключить флаг SM100, операция запустится снова.

ПРИМЕЧАНИЕ: Когда передача данных будет завершена, включается флаг приема SM98, а затем ПЛК начнет принимать данные до тех пор, пока не примет конечный символ. Когда прием данных завершен, включается флаг SM100. Если возникает пауза коммуникации, но конечный символ еще не принят, будет включен флаг паузы коммуникации SM104.



**Примечание:**

1. Нет ограничений на количество выполнений инструкции коммуникации COMRS. Однако каждый коммуникационный порт может быть включен только одной инструкцией, последующие инструкции коммуникации на этом порту выполнены не будут.
2. Когда выполняется инструкция COMRS, контрольная сумма не используется. Если необходима контрольная сумма, можно использовать вместе с инструкцией COMRS другую инструкцию.
3. Если значение в  $D_2$  равно 2, 3, 5, 6 или 9, рекомендуется установить период ожидания. После того, как установлен такой период, отправка данных будет повторена, если не получен конечный определенный символ.
4. Инструкция автоматически не сбрасывает значения в  $D_1 \sim D_{1+n}$  при каждом выполнении или, если ПЛК начинает получать новые данные коммуникации. Получение данных и количество полученных данных коммуникации можно узнать только после включения флага завершения. Для сброса значений в  $D_1 \sim D_{1+n}$  можно использовать инструкцию ZRST.
5. Если значение в  $S_1$  находится вне допустимого диапазона, инструкция не будет выполнена.
6. Если количество регистров, начиная с  $S_2$  не равно значению в  $S_3$ , инструкция не будет выполнена, включится флаг SM0 и в SR0 будет записан код ошибки 16#2003.
7. Если значение в  $D_2$  не находится в диапазоне от 0 до 9, инструкция не будет выполнена, включится флаг SM0 и в SR0 будет записан код ошибки 16#200B.
8. Если значение в  $D_2$  равно 6, 8 или 9, а количество регистров, начиная с  $D_1$  не равно значению в  $D_2+1$ , инструкция не будет выполнена, включится флаг SM0 и в SR0 будет записан код ошибки 16#2003.
9. Если количество полученных фрагментов данных больше количества регистров в объектах, начиная с  $D_1$ , данные, которые не могут быть сохранены, будут игнорироваться.
10. Если флаг завершения включен, ПЛК прекратит прием данных. Если на коммуникационный порт направлять данные при включенном флаге завершения, приниматься данные не будут.
11. Если значение настройки в  $S_3$  меньше 0 или больше 256, инструкция не будет выполнена, включится флаг SM0 и в SR0 будет записан код ошибки 16#200B.
12. Когда режим в  $D_2$  равен 6 или 8, а значение  $D_2+1$  меньше 1 или больше 256, инструкция не будет выполнена, включится флаг SM0 и в SR0 будет записан код ошибки 16#200B.

API	Код инструкции			Операнды							Функция						
1813		COMDF	P	$S_1 \cdot S_2 \cdot S_3 \cdot S_4 \cdot S_5 \cdot D$							Настройка коммуникации для последовательного коммуникационного порта						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$													<input type="radio"/>	<input type="radio"/>		
$S_2$													<input type="radio"/>	<input type="radio"/>		
$S_3$													<input type="radio"/>	<input type="radio"/>		
$S_4$													<input type="radio"/>	<input type="radio"/>		
$S_5$													<input type="radio"/>	<input type="radio"/>		
D													<input type="radio"/>	<input type="radio"/>		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$													
$S_2$													
$S_3$													
$S_4$													
$S_5$													
D													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**

COMDF		COMDFP	
En		En	
S1		S1	
S2		S2	
S3		S3	
S4		S4	
S5		S5	
D		D	

$S_1$  : Скорость обмена данными (шаг:100 бит/с)

$S_2$  : Число битов данных

$S_3$  : Бит четности

$S_4$  : Число конечных битов

$S_5$  : Выбор формата MODBUS

D : Номер коммуникационного порта

**Описание:**

1. Инструкция предоставляет только способ непосредственного задания значений параметров вместо задания переменной.
2.  $S_1$  задает скорость обмена данными (шаг: 100 бит/с). Например, значение 96 задает скорость 9600 бит/с.
3.  $S_2$  задает число битов данных. Значение 7 задает 7 битов данных, 8 – 8 битов данных. Если значение в  $S_2$  равно 7 or 8, инструкция выполняется со значением по умолчанию.

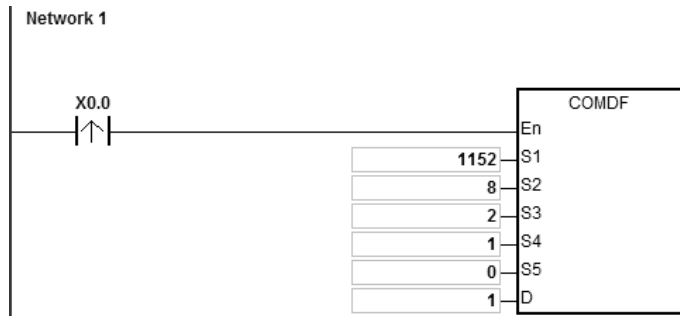
4. **S<sub>3</sub>** задает бит четности. Значение 0 означает отсутствие бита четности. 1 означает проверку нечетного бита. Значение 2 означает проверку четного бита. Если значение в **S<sub>3</sub>** не равно 0, 1 или 2, инструкция выполняется со значением по умолчанию.
5. **S<sub>4</sub>** задает число конечных битов. Значение 1 означает 1 бит, 2 – 2 бита. Если значение в **S<sub>4</sub>** не равно 1 или 2, инструкция выполняется со значением по умолчанию.
6. **S<sub>5</sub>** задает режим связи на базе протокола Modbus. Значение 0 задает режим ASCII (по умолчанию), а 1 задает режим RTU. Если значение в **S<sub>5</sub>** не равно 0 или 1, инструкция выполняется со значением по умолчанию.
7. **D** задает номер коммуникационного порта. Номер порта COM1 равен 1, COM2 - 2, платы1 - 11 и платы2 - 12. Если значение настройки не соответствует допустимому диапазону, инструкция не будет выполнять настройки коммуникационного порта.
8. Пользователь может настроить коммуникационный порт через ПО **ISPSoft**—>**HWCONFIG**—>**COM Port** или **специальные регистры** (для настройки HWCONFIG, см. Руководство ПО ISPSoft. Раздел 6.19.3 настоящего Руководства описывает специальные регистры SR и SM, относящиеся к коммуникации)
9. Связь на фактическом коммуникационном порту изменится сразу после использования инструкции. Если какое-то сообщение выполняется в настоящий момент, оно будет принудительно отменено. Кроме того, будут также изменены соответствующие настройки SM / SR. Подробнее о SM / SR см. Раздел 6.19.3 настоящего Руководства.
10. Инструкция не будет менять какие-либо настройки для фактического коммуникационного порта, если настройка формата связи совпадает с предыдущей настройкой.

**Пример:**

1. Возьмем для примера порт COM1. Настройка других коммуникационных портов ПЛК аналогична.
2. Стартовый контакт: X0.0.
3. Формат связи для COM1 (RS485) задан как 115200, 8, E и 1.
4. Режим связи для COM1 (RS485) задан как ASCII.
5. Описание инструкции COMDF:

Операнд	Описание		Значение
<b>S<sub>1</sub></b>	Скорость обмена данными	115200 bps	1152
<b>S<sub>2</sub></b>	Число битов данных	8	8
<b>S<sub>3</sub></b>	Бит четности	E	2

<b>S<sub>4</sub></b>	Число конечных битов	1	1
<b>S<sub>5</sub></b>	Выбор режима MODBUS	ASCII	0
<b>D</b>	Номер коммуникационного порта	COM1	1



API	Код инструкции			Операнды				Функция								
1814		VFDRW		$S_1 \cdot S_2 \cdot S_3 \cdot S$				Инструкция по последовательной коммуникации для ПЧ Delta								

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$								●	●		○	○	○	○		
$S_2$								●	●		○	○	○	○		
$S_3$								●	●		○	○	○	○		
$S$								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							
$S$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**

VFDRW
En
S1
S2
S3
S

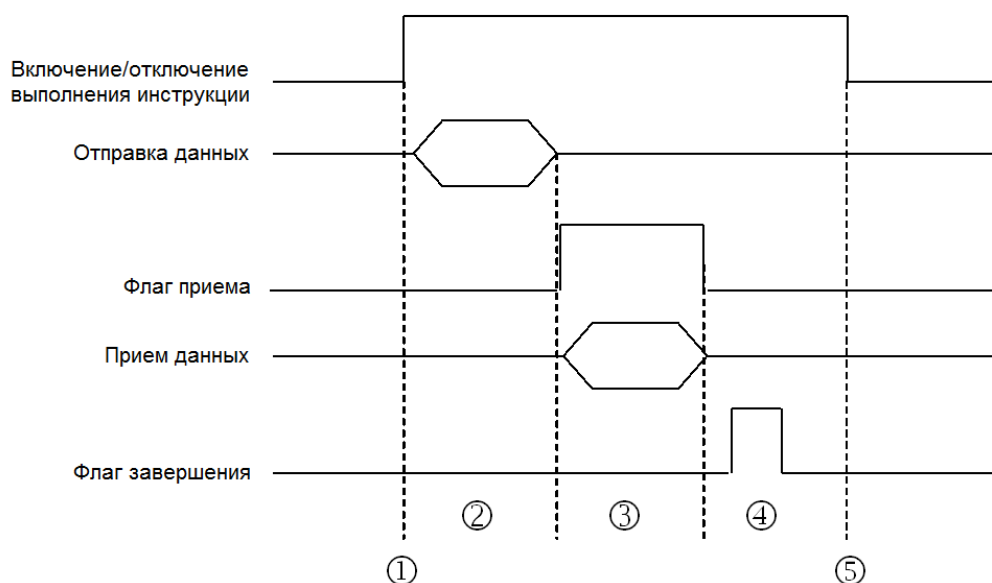
- $S_1$  : Номер коммуникационного порта
- $S_2$  : Адрес станции ПЧ
- $S_3$  : Функциональный код
- $S$  : Источник и приемник данных

**Описание:**

1.  $S_1$  задает номер коммуникационного порта. Номер COM1 - 1, COM2 - 2, Card1 - 11 и Card2 - 12. Если значение настройки не соответствует допустимому диапазону, инструкция не будет выполнять настройки коммуникационного порта.
2.  $S_2$  задает адрес станции ПЧ VFD. Если адрес станции равен 0, это означает, что выбран широковещательный режим. Диапазон настройки составляет 0~254. Если значение адреса находится вне диапазона, инструкция выполнена не будет.
3.  $S_3$  – функциональный код, а  $S$  – источник или приемник данных, согласно таблице ниже.

S <sub>3</sub> функц. код	S <sub>3</sub> функция	S источник и приемник данных	Примечание
0	Сброс	Не используется	В S может быть введено любое значение.
1	Команда вращения по часовой стрелке	Значение скорости	См. Руководство по эксплуатации ПЧ серии VFD.
2	Команда вращения против часовой стрелки	Значение скорости	
3	Останов	Не используется	В S может быть введено любое значение.
4	Команда Jog по часовой стрелке	Не используется	См. Руководство по эксплуатации ПЧ серии VFD для информации по настройке Jog режима.
5	Команда Jog против часовой стрелки	Не используется	
6	Чтение состояния	Полученные значения состояния	См. Руководство по эксплуатации ПЧ серии VFD для информации по значениям 5-битных адресов H2100 ~ H2104 в ПЧ серии VFD.

## 4. Временная диаграмма для отправляемых и принимаемых данных:



Описание:

- ① → Пользователь запускает/останавливает выполнение инструкции.
- ② → Передача данных начинается после запуска инструкции. В течение этого времени время паузы связи не измеряется.

- ③ → Флаг приема включен. С момента получения первого символа до момента получения следующего символа будет измеряться период времени. Всякий раз, когда символ получен, измеренное время очищается. Флаг паузы связи будет включен, если измеренное время больше, чем значение задания паузы связи.
  - ④ → Когда получение данных будет завершено, будет включен флаг завершения. Сбрасывается этот флаг пользователем.
  - ⑤ → Прежде чем команда связи будет отправлена еще раз, необходимо остановить действие инструкции в течение одного цикла после просмотра флага завершения. В этом случае инструкция запускается в следующем цикле.
5. Нет ограничений на количество использований инструкции. Один коммуникационный порт может использоваться каждый раз только для вывода и выполнения одной команды связи. Если получение и отправка данных завершены, необходимо отключить инструкцию для возможности управления связью.

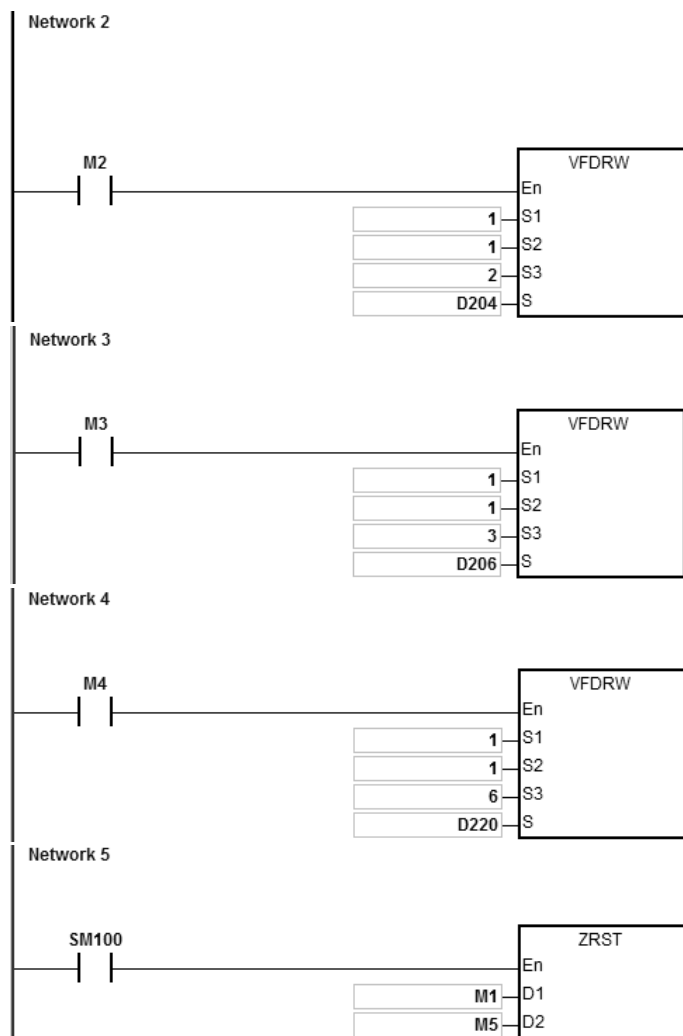
**Пример настройки протокола коммуникации:**

1. С помощью HWCONFIG задайте для коммуникационного порта COM1 (RS485) адрес станции 2 и формат связи: ASCII, 115200, 7, N, 2.
2. Сделайте базовую настройку с помощью пульта управления ПЧ Delta C2000 в соответствии со следующими шагами.
  - A. Задайте параметру 09-00 значение 1, адрес станции ПЧ задается как 1.
  - B. Задайте параметру 09-01 значение 115.2, скорость обмена данными RS485 задается как 115200.
  - C. Задайте параметру 09-04 значение 1, формат связи RS485: 7, N, 2.
  - D. Задайте параметру 09-20 значение 1, ввод частотного задания через RS485.
  - E. Задайте параметру 09-21 значение 2, ввод режима работы ПЧ через RS485.

**Пример:**

Используя инструкцию VFDRW для управления скоростью, зададим ПЧ VFD вращение вперед на частоте 120 Гц, обратное вращение на частоте 180 Гц, а затем останов.





1. Подключите AS COU к ПЧ VFD.

Сначала задайте значение D202=12000. Когда M1 включен, ПЧ начинает разгон в направлении по часовой стрелке и вращается с частотой 120 Гц.

2. Сначала задайте значение D204=18000. Когда M2 включен, ПЧ после приема команды вращения против часовой стрелки будет замедляться, а затем вращаться против часовой стрелки на частоте 180 Гц.

3. Когда M3 включен (в данный момент значение в D206 неэффективно), ПЧ после приема команды Стоп останавливается.

4. Когда M4 включен, значения в H2100~H2104 ПЧ считываются в D220~224.

Регистр	D220	D221	D222	D223	D224
Содержимое	Код ошибки	Состояние ПЧ	Частотное задание	Выходная частота	Выходной ток



Состояние ПЧ:

Когда бит2 =1, ПЧ работает в режиме Jog. Если Bit4~3= 11В, ПЧ осуществляет вращение против часовой стрелки. Если задание частоты равно 18000, ПЧ работает на частоте 180 Гц.

(См. Руководство по эксплуатации для ПЧ для дополнительной информации о параметрах и адресах коммуникации).

5. Когда флаг завершения SM100 включен, значения M1 ~ M5 очищаются, чтобы избежать влияния на выполнение следующей команды связи.

После того, как будут получены отправленные в ПЧ данные, будет проверен формат данных, отправленных из ПЧ. Если формат данных верен, включается флаг SM100. В противном случае включается флаг SM102.

API	Код инструкции			Операнды							Функция						
1815		ASDRW		$S_1 \cdot S_2 \cdot S_3 \cdot S$							Инструкция по последовательной коммуникации для сервопривода Delta						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$								●	●		○	○	○	○		
$S_2$								●	●		○	○	○	○		
$S_3$								●	●		○	○	○	○		
$S$								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							
$S$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**

ASDRW
En
S1
S2
S3
S

- $S_1$  : Номер коммуникационного порта
- $S_2$  : Адрес станции сервопривода
- $S_3$  : Функциональный код
- $S$  : Источник и приемник данных

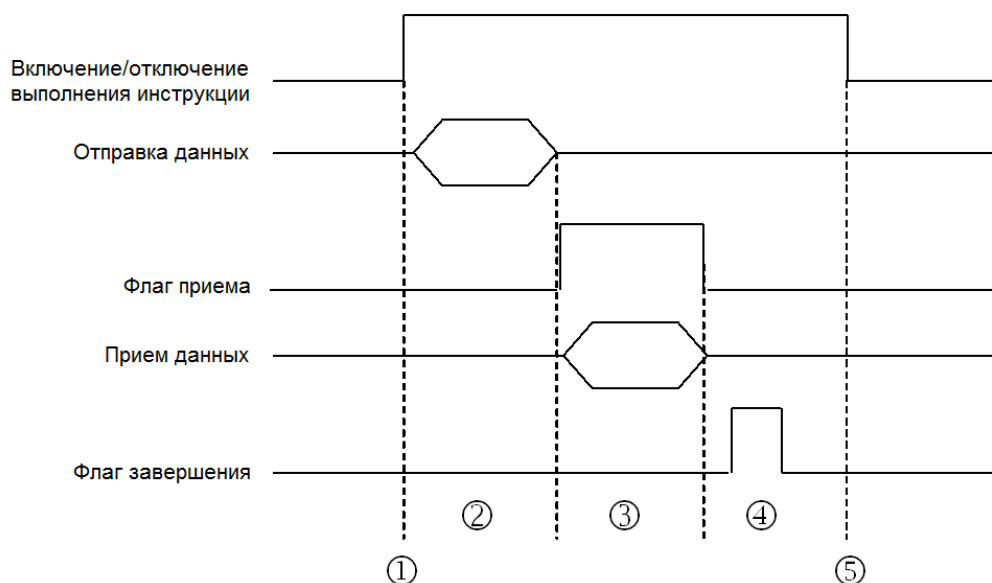
**Описание:**

- $S_1$  задает номер коммуникационного порта. Номер COM1 - 1, COM2 - 2, Card1 - 11 и Card2 - 12. Если значение настройки не соответствует допустимому диапазону, инструкция не будет выполнять настройки коммуникационного порта.
- $S_2$  задает адрес станции сервопривода. Если адрес станции равен 0, это означает, что выбран широкопередаточный режим. Диапазон настройки составляет 0~254. Если значение адреса находится вне диапазона, инструкция выполнена не будет.
- См. Руководство по эксплуатации на сервоприводы Delta ASD для информации по параметрам сервопривода.
- $S_3$  – функциональный код, а  $S$  – источник или приемник данных, согласно таблице ниже.

Применимо к серии B2			
S <sub>3</sub> функц. код	S <sub>3</sub> функция	S источник и приемник данных	Примечание
0	Чтение состояния сервопривода	Занимает 5 последовательных объектов S~S+4	Чтение состояния из параметров P0-04~P0-08
1	Чтение значения регистра сервопривода	Занимает 8 последовательных объектов S~S+7	Чтение состояния из параметров P0-09~P0-16
2	Запись значения в регистр сервопривода	Занимает 8 последовательных объектов S~S+7	Запись данных в регистры для P0-09~P0-16
3	Ввод скорости режима Jog, вращение по часовой стрелке, вращение против часовой стрелки и останов	Диапазон скорости: 1~3000; 4999 (по часовой); 4998 (против часовой); 5000 (останов)	Запись данных в регистр для P4-05
4	ВКЛ/ОТКЛ сервопривода (сигнал Servo ON/OFF)	1: ВКЛ; Другое значение: ОТКЛ	Запись данных в регистр для P2-30
5	Задание скорости	Диапазон: -5000~5000	Запись данных в регистры для P1-09 ~P1-11
6	Задание момента	Диапазон: -300~300	Запись данных в регистры для P1-12 ~ P1-14

Применимо к серии A2			
S <sub>3</sub> функц. код	S <sub>3</sub> функция	S источник и приемник данных	Примечание
16	Чтение состояния сервопривода	Занимает 10 последовательных объектов S~S+9	Чтение состояния из параметров P0-09 ~ P0-13 (32-битные значения)
17	Запись значения в регистр сервопривода	Занимает 8 последовательных объектов S~S+7	Запись данных в регистры для P0-17 ~ P0-20 (32-битные значения)
18	Запись значения параметра сопоставления	Занимает 8 последовательных объектов S~S+7	Запись данных в регистры для P0-25 ~ P0-28 (32-битные значения)
19	Ввод скорости режима Jog, вращение по часовой стрелке, вращение против часовой стрелки и останов	Диапазон скорости: 1~3000; 4999 (по часовой); 4998 (против часовой); 0 (останов)	Запись данных в регистр для P4-05
20	ВКЛ/ОТКЛ сервопривода (сигнал Servo ON/OFF)	1: ВКЛ; Другое значение: ОТКЛ	Запись данных в регистр для P2-30
21	Задание скорости (3 набора)	Занимает 6 последовательных объектов с диапазоном задания: -60000~60000	Запись данных в регистры для P1-09 ~ P1-11 (32- битные значения)
22	Задание момента (3 набора)	Занимает 6 последовательных объектов с диапазоном задания: -300~300	Запись данных в регистры для P1-12 ~ P1-14 (32- битные значения)
23	Задание целевых параметров сопоставления сервопривода	Занимает 8 последовательных объектов S~S+7	Запись данных в регистры для P0-35 ~ P0-38 (32- битные значения)

## 5. Временная диаграмма для отправляемых и принимаемых данных:



Описание:

- ① → Пользователь запускает/останавливает выполнение инструкции.
- ② → Передача данных начинается после запуска инструкции. В течение этого времени время паузы связи не измеряется.
- ③ → Флаг приема включен. С момента получения первого символа до момента получения следующего символа будет измеряться период времени. Всякий раз, когда символ получен, измеренное время очищается. Флаг паузы связи будет включен, если измеренное время больше, чем значение задания паузы связи.
- ④ → Когда получение данных будет завершено, будет включен флаг завершения. Сбрасывается этот флаг пользователем.
- ⑤ → Прежде чем команда связи будет отправлена еще раз, необходимо остановить действие инструкции в течение одного цикла после просмотра флага завершения. В этом случае инструкция запускается в следующем цикле

- 6. Нет ограничений на количество использований инструкции. Один коммуникационный порт может использоваться каждый раз только для вывода и выполнения одной команды связи. Если получение и отправка данных завершены, необходимо отключить инструкцию для возможности управления связью.

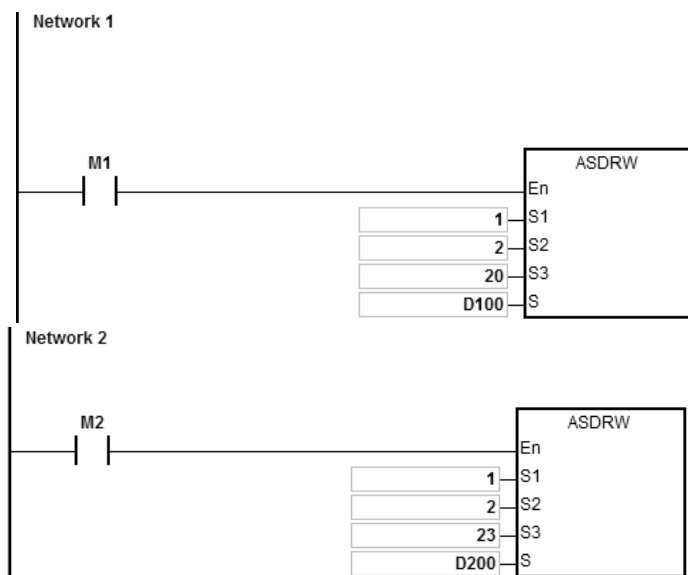
**Пример настройки протокола коммуникации:**

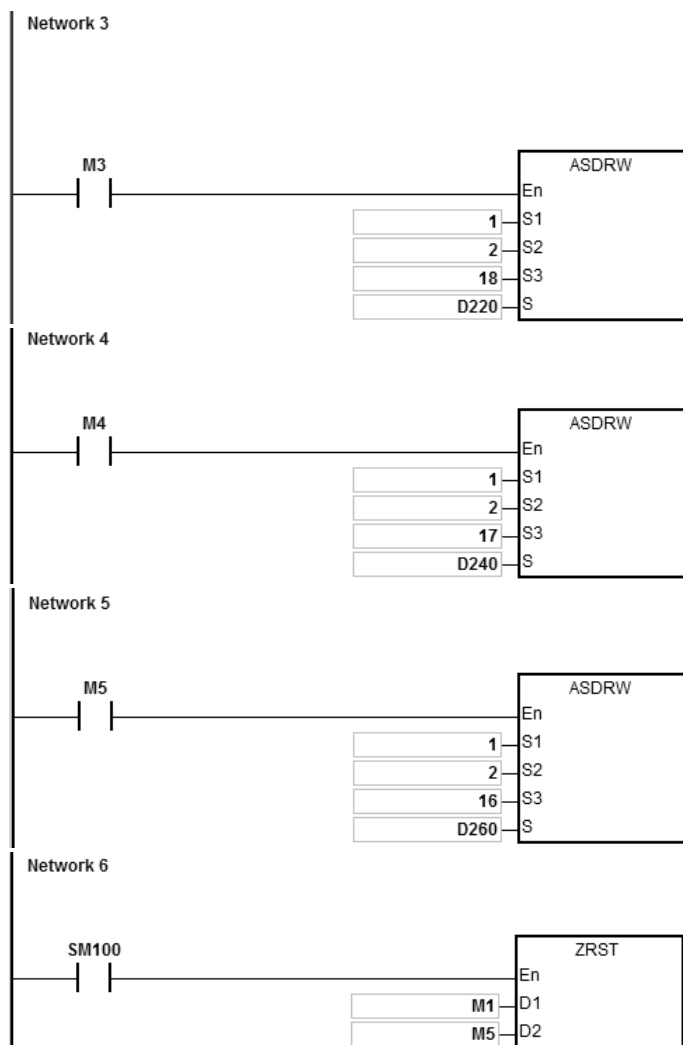
1. С помощью HWCONFIG задайте для коммуникационного порта COM1 (RS485) формат связи ASCII, 115200, 8, E, 1.
2. Сделайте базовую настройку с помощью пульта управления сервопривода Delta ASDA-A2 в соответствии со следующими шагами.
  - A. Заданием параметру P2-08 значения 10 восстанавливаем заводскую настройку.
  - B. Перезапускаем питание сервопривода.
  - C. задаем P1-01 значение 0001 (PR режим).
  - D. Задаем P3-00 значение 2, устанавливая адрес станции сервопривода равный 2.
  - E. Задаем P3-01 значение 0205, скорость обмена данными для RS485 устанавливается как 115200 бит/с.
  - F. Задаем P3-02 значение 0004, формат связи для RS485 сервопривода будет: 8, E, 1.
  - G. Включаем сервопривод снова после завершения установки.

**Пример:**

Использование инструкции ASDRW для управления скоростью, задание относительного позиционирования 5000000PUU ускорением в течение 400 мс до скорости 3000 об/мин и затем замедление в течение 200 мс.

6





1. Подключите модуль ЦПУ ПЛК AS к сервоприводу ASDA-A2.

Сначала задайте D100=1. Когда M1 включен, ASDA-A2 получает сигнал Servo ON.

2. Задайте значения в D200~D207 согласно таблице ниже, которые будут записаны в параметры P0-35 ~ P0-38 сервопривода ASDA-A2. Они занимают при этом 8 последовательных регистров.

Регистр	D200	D201	D202	D203	D204	D205	D206	D207
Заданное значение (16#)	05140515		06020602		06030603		053C0507	

Когда M2 включен, значения в D200~D207 записываются в параметры P0-35~P0-38 сервопривода ASDA-A2.

Заданные значения параметров P0-35~P0-38 используются для установки целевых параметров сопоставления для параметров P0-25~P0-28. См. Руководство по эксплуатации сервоприводов Delta.

ASDA-A2	P0-35	P0-36	P0-37	P0-38
Заданное значение (16#)	0514 0515	0602 0602	0603 0603	053C 0507

Когда значение P0-38 задано как 053C 0507, это означает, что целевыми параметрами сопоставления для P0-28 будут P5-60 (16 битов) и P5-07 (16 битов).

ASDA-A2	P0-25	P0-26	P0-27	P0-28
Целевой параметр сопоставления	P5-20, P5-21	P6-02	P6-03	P5-60, P5-07
Наименование параметра	Время разгона/торможения 1 / время разгона/торможения 2	Тип траектории 1	Данные траектории 1	Заданная скорость и триггер командного задания режима PR

3. Задание значений в D220~D227 показано ниже.

Регистр	D220	D221	D222	D223	D224	D225	D226	D227
Заданное значение (16#)	0190 00C8		0000 1083		4C4B40		7530 0001	

Когда M3 включен, значения в D220~D227 записываются в параметры P0-25~28 сервопривода ASDA-A2.

ASDA-A2	P0-25	P0-26	P0-27	P0-28
Заданное значение (16#)	0190 00C8	0000 1083	4C4B40	7530 0001

Сервопривод запускается со временем разгона = 0190 (400 мс), временем торможения =00C8 (200 мс), тип траектории =1083, команда позиционирования =4C4B40 (5000000PUU), заданная скорость =7530 (3000 об/мин) и триггер командного задания режима PR =1.

4. Задание значений в D240~D247 показано ниже, значения записываются в параметры P0-17~P0-20 сервопривода ASDA-A2. Они занимают при этом 8 последовательных регистров.

Регистр	D240	D241	D242	D243	D244	D245	D246	D247
Заданное значение (10#)	41		0		0		0	

Когда M4 включен, значения в D240~D247 записываются в параметры P0-17~20 сервопривода ASDA-A2.

Заданные значения в параметрах P0-17~20 используются для установки содержимого в параметрах P0-09~12.

(Для настройки отображения содержимого см. Руководство по эксплуатации сервоприводов Delta)

ASDA-A2	P0-17	P0-18	P0-19	P0-20
Заданное значение (10#)	41	0	0	0

Когда настройка параметра P0-17 равна 41, отображаемое содержимое в P0-09 является состоянием сервопривода.

ASDA-A2	P0-09	P0-10	P0-11	P0-12
Содержимое	Состояние сервопривода	Число импульсов обратной связи двигателя	Число импульсов обратной связи двигателя	Число импульсов обратной связи двигателя

5. Когда M5 включен, значения в P0-17~ P0-20 сервопривода ASDA-A2 считываются в D260~D267.

Регистр	D260	D262	D264	D266
Содержимое	Состояние сервопривода	Число импульсов обратной связи двигателя	Число импульсов обратной связи двигателя	Число импульсов обратной связи двигателя

Когда бит состояния Бит4 равен1, это показывает достижение заданной позиции.

(см. Руководство по эксплуатации сервоприводов Delta для информации по параметру P0-46.)

6. Когда включен флаг выполнения SM100, значения M1 ~ M5 очищаются во избежание влияния на выполнение следующей команды связи.

В конце сеанса коммуникации будет проверен формат данных, отправленных на ПЛК из сервопривода ASDA-A2. Если формат данных верен, включается флаг SM100. Если формат данных неверен, включается флаг SM102.



API	Код инструкции		Операнды									Функция					
1816		CCONF	P	S <sub>1</sub> ~S <sub>11</sub>									Установка параметров в таблице обмена данными коммуникационного порта				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●				○	○		
S <sub>3</sub>								●	●				○	○		
S <sub>4</sub>								●	●				○	○		
S <sub>5</sub>								●	●				○	○		
S <sub>6</sub>								●	●				○	○		
S <sub>7</sub>			●					●								
S <sub>8</sub>								●	●				○	○		
S <sub>9</sub>								●	●				○	○		
S <sub>10</sub>								●	●				○	○		
S <sub>11</sub>			●					●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub> ~S <sub>11</sub>	См. описание инструкции												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Symbol:

CCONF	CCONFP
En	En
S1	S1
S2	S2
S3	S3
S4	S4
S5	S5
S6	S6
S7	S7
S8	S8
S9	S9
S10	S10
S11	S11

S<sub>1</sub>~S<sub>11</sub> : См. описание инструкции

## Описание:

1. Наименование и описание операндов  $S_1$ – $S_{11}$  представлены в таблице ниже.

Операнд	Наименование	Описание	Тип данных	Примеч.
$S_1$	Номер коммуникационного порта	1=COM1, 2=COM2. Если задано другое значение, инструкция выполняться не будет.	WORD	
$S_2$	Номер позиции в таблице обмена данными	Диапазон: 1~32. Если задано другое значение, инструкция выполняться не будет.	WORD	
$S_3$	Адрес станции удаленного оборудования	Диапазон: 0 и 1~240. Если значение больше 240, автоматически будет установлено 240. Если значение меньше или равно 0, будет сохранено оригинальное значение адреса для устройства.	WORD	Отрицательное значение будет отображаться как 0.
$S_4$	Функциональный код для чтения данных	Поддерживаются 16#01, 02, 03, 04 и 17. Если задано другое значение, инструкция выполняться не будет. Если $S_4=16\#17$ , значение $S_8$ также будет равно 16#17.	WORD	
$S_5$	Чтение удаленного адреса связи	16#0000~16#FFFF	WORD	
$S_6$	Длина считываемых данных	Битовый функциональный код: поддержка 0~256 битов; Словный функциональный код: поддержка 0~100 слов. Значение 0 показывает, что данные не считываются. Если значение больше максимального, будет задано максимальное значение.	WORD	
$S_7$	Локальный регистр для хранения полученных данных	Битовый функциональный код: выбирается только объект M. Словный функциональный код: выбирается только объект D.	BOOL WORD	
$S_8$	Функциональный код для записи данных	Поддерживаются 16#05, 06, 0F, 10. Если задано другое значение, инструкция выполняться не будет.	WORD	
$S_9$	Запись удаленного адреса связи	16#0000~16#FFFF	WORD	
$S_{10}$	Длина записываемых данных	Битовый функциональный код: поддержка 0~256 битов; Словный функциональный код: поддержка 0~100 слов. Значение 0 показывает, что данные не записываются. Если значение больше максимального, будет задано максимальное значение.	WORD	
$S_{11}$	Локальный регистр для записи данных удаленного устройства	Битовый функциональный код: выбирается только объект M. Словный функциональный код: выбирается только объект D.	BOOL WORD	

2. Инструкция используется как импульсная инструкция.

3. Подробные сведения о функциональных кодах Modbus в **S<sub>4</sub>** и **S<sub>8</sub>** приведены ниже.

ПЛК считывает данные с нескольких битовых регистров (кроме дискретных входов) = 1 (16#01).

ПЛК считывает данные с нескольких битовых регистров (только дискретные входы) = 2 (16#02).

ПЛК считывает данные с нескольких словных регистров (кроме входных регистров) = 3 (16#03).

ПЛК считывает данные с нескольких словных регистров (только входные регистры) = 4 (16#04).

ПЛК записывает данные в несколько битовых регистров = 5 (16#05).

ПЛК записывает данные в несколько словных регистров = 6 (16#06).

ПЛК записывает состояние в несколько битовых регистров = 15 (16#0F).

ПЛК записывает состояние в несколько словных регистров = 16 (16#10).

ПЛК синхронно считывает и записывает данные на несколько словных регистров = 23 (16#17).

Поддерживаются только упомянутые выше функциональные коды. Для других значений функциональных кодов, например 0, значение либо будет считаться недействительным (адрес связи, длину и начальный регистр), либо инструкция будет выполнять функцию обмена данными на основе параметров связи по умолчанию.

4. Когда в **S<sub>4</sub>** (функциональный код чтения) задано значение 16#17 (синхронизированные чтение и запись), операнд **S<sub>8</sub>** (функциональный код записи) недействителен и значение 16#17 автоматически реализует запись данных.
5. Когда в **S<sub>8</sub>** (функциональный код записи) задано значение 16#05 или 16#06, операнд **S<sub>10</sub>** (длина записываемых данных) недействителен и 16#05 или 16#06 автоматически реализуют запись фрагмента данных.
6. Значения параметров, указанные инструкцией, действительны только во время работы ПЛК. После перезагрузки ПЛК данные в таблице обмена, установленные через HWCONFIG, сбрасываются на значения по умолчанию. Если в ходе выполнения программы необходимо изменение значений параметров, для их изменения потребуется выполнение данной инструкции.
7. Инструкция используется для немедленного задания параметров для коммуникационного соединения, когда функция обмена данными еще не запущена. Во время выполнения обмена данными новое значение параметра коммуникации может быть изменено, но не может быть активировано до конца текущего цикла.
8. Инструкция реализует только функцию изменения параметров коммуникации. Обратите внимание на флаги в таблице ниже для запуска и закрытия функции коммуникации, если программа ПЛК используется для запуска или закрытия соединения коммуникации.

Когда пользователь устанавливает функцию автоматического сканирования через программное обеспечение для редактирования, флаг запуска / остановки соединения автоматически обновляет состояние запуска / остановки один раз после того, как функция обмена данными завершит выполнение сканирования.

Подробная информация по флагам SM содержится в таблице ниже.

SM	Атрибут	Описание параметров связи для COM1
SM750	R/W	Флаг запуска обмена данными
SM752 ~ SM783	R/W	Флаг запуска соединения 1~32 для обмена данными
SM784 ~ SM815	R	Флаг успешного чтения данных при соединении 1~32
SM816 ~ SM847	R	Флаг ошибки обмена данными при соединении 1~32
SM	Атрибут	Описание параметров связи для COM2
SM862	R/W	Флаг запуска обмена данными
SM864 ~ SM895	R/W	Флаг запуска соединения 1~32 для обмена данными
SM896 ~ SM927	R	Флаг успешного чтения данных при соединении 1~32
SM928 ~ SM959	R	Флаг ошибки обмена данными при соединении 1~32

Флаг успешного чтения данных включается при корректном чтении данных. При возникновении ошибки или прерывания связи включается флаг ошибки обмена данными (при этом пользователь может просмотреть код ошибки). Т.к. оба флага при одном соединении не могут быть включены одновременно, ПЛК не сбрасывает эти флаги во время обмена данными.

9. В таблице ниже представлены регистры (только для чтения) SR для коммуникации.

SR	Описание
SR1335	Наличие цикла фактического соединения 1~32 по COM1
SR1336	Номер соединения по COM1 в настоящее время
SR1340 ~ SR1371	Коды ошибок соединения 1~32 по COM1
SR1375	Наличие цикла фактического соединения 1~32 по COM1
SR1376	Номер соединения по COM2 в настоящее время
SR1380 ~ SR1411	Коды ошибок соединения 1~32 по COM2

10. Функция обмена данными не предусматривает флаг успешной записи данных. Оценку успешной записи можно провести по факту смены номера соединения или с использованием специального флага успешной записи.

Например, когда номер выполняемого соединения в SR1336 равен 3, сначала считывают данные, а затем после завершения чтения данные связи записывают. При успешной записи данных номер подключения изменяется на 4.

11. Если длина данных в **S<sub>6</sub>** и значение в **S<sub>7</sub>** превышают диапазоны для объектов D или M, значение в **S<sub>6</sub>** будет автоматически изменено на значение в пределах допустимого диапазона. Например, если длина данных в **S<sub>6</sub>** равна 100 и значение в **S<sub>7</sub>** равно M8182, значение в **S<sub>6</sub>** автоматически будет исправлено на 10.
12. В нижеприведенных случаях возникает ошибка, параметры в HWCONFIG не будут изменены, включится флаг SM0 и в SR0 будет отображен код ошибки 16#200B.
  - Если значения в **S<sub>1</sub>**, **S<sub>2</sub>**, **S<sub>4</sub>** и **S<sub>8</sub>** превышают допустимый диапазон, т.е. это ошибка ввода.
  - Когда функциональные коды в **S<sub>4</sub>** или **S<sub>8</sub>** для чтения или записи данных битового типа, регистр в операндах **S<sub>7</sub>** или **S<sub>11</sub>** не является регистром M. Это также ошибка ввода.
  - Когда функциональные коды в **S<sub>4</sub>** или **S<sub>8</sub>** для чтения или записи данных словного типа, регистр в операндах **S<sub>7</sub>** или **S<sub>11</sub>** не является регистром D. Это также ошибка ввода.

**Пример: Порт ПЛК COM1 (RS485):**

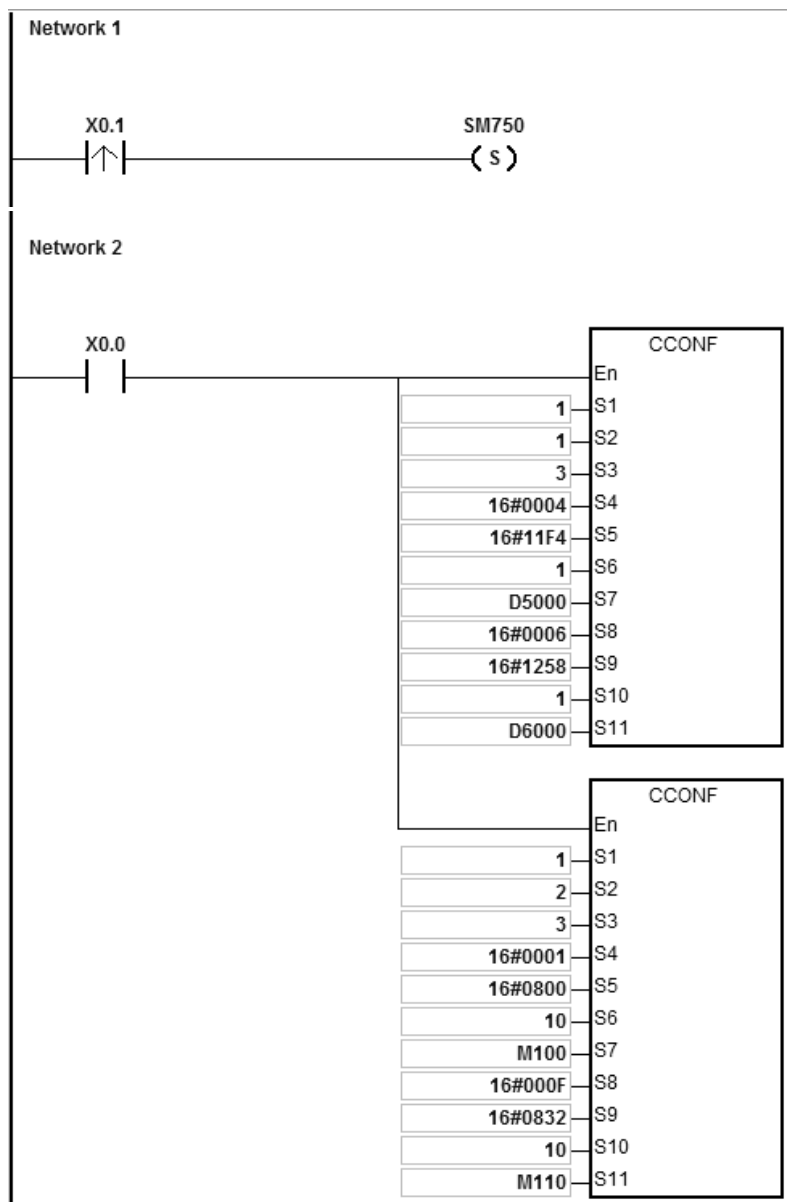
1. Для обмена данными между ЦПУ AS и ЦПУ DVP-ES2 через COM1, таблица обмена данными в ПО ISPSOFT HWConfig показана ниже.

№	Метод обнаружения	Адрес удаленной станции	Регистр сохранения		Удаленный регистр	Длина	Функциональный код	Старт
1	Назначенное соединение	2	D50	←	D50	50	H03	Программно
			D100	→	D100	50	H10	
2	Назначенное соединение	2	D150	←	D150	50	H03	Программно
			D200	→	D200	50	H10	

2. Данные для обмена между ЦПУ AS и ЦПУ DVP-ES2 приведены ниже.

ПЛК AS (Master)	Данные	ПЛК ES2 (Slave)	Данные
D50~D99	0	D50~D99	1~50
D100~D149	100~149	D100~D149	0
D150~D199	0	D150~D199	150~199
D200~D249	200~249	D200~D249	0

3. Обмен данными начинается после включения X0.1.



4. После запуска обмена данными соответствующие данные в ЦПУ AS и ЦПУ DVP-ES2 изменяются, как показано в таблице ниже.

ПЛК AS (Master)	Данные	ПЛК ES2 (Slave)	Данные
D50~D99	1~50	D50~D99	1~50
D100~D149	100~149	D100~D149	100~149
D150~D199	150~199	D150~D199	150~199
D200~D249	200~249	D200~D249	200~249

5. Когда X0.0 включен, таблица обмена данными по COM1 в ЦПУ AS изменяется следующим образом.

№	Метод обнаружения	Адрес удаленной станции	Регистр сохранения		Удаленный регистр	Длина	Функциональный код	Старт
1	Назначенное соединение	3	D5000	←	D500	1	H04	Программно
			D6000	→	D600	1	H06	
2	Назначенное соединение	3	M100	←	M0	10	H01	Программно
			M110	→	M50	10	H0F	

6. Ввиду изменения таблицы обмена данными по COM1 в ЦПУ AS, данные при обмене выглядят следующим образом.

ПЛК AS (Master)	Данные	ПЛК ES2 (Slave)	Данные
D5000	3000	D500	3000
D6000	4000	D600	4000
M100~M109	ON	M0~M9	ON
M110~M119	OFF	M50~M59	OFF

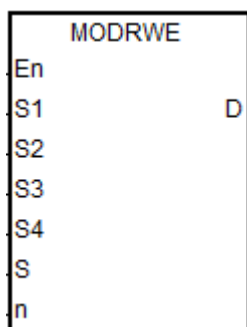
API	Код инструкции			Операнды							Функция						
1817		MODRWE		$S_1 \cdot S_2 \cdot S_3 \cdot S_4 \cdot S \cdot n \cdot D$							Чтение и запись данных Modbus без использования какого-либо флага						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●				○	○		
S <sub>3</sub>								●	●				○	○		
S <sub>4</sub>								●	●				○	○		
S								●								
n								●	●				○	○		
D		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
S <sub>4</sub>		●			●	●							
S	●	●			●	●							
n		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:



- S<sub>1</sub> : Номер порта коммуникации
- S<sub>2</sub> : Адрес устройства
- S<sub>3</sub> : Функциональный код
- S<sub>4</sub> : Адрес данных
- S : Регистр участвует в чтении и записи данных
- n : Длина данных
- D : Флаг завершения чтения и записи данных

Описание:

- S<sub>1</sub> задает номер коммуникационного порта. Номер COM1 - 1, COM2 - 2, Card1 - 11 и Card2 - 12. Если значение настройки не соответствует допустимому диапазону, инструкция не будет выполнять настройки коммуникационного порта.
- См. описание инструкции API1808 MODRW для сведений по операндам S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, S и n.



3. **D** устанавливает флаги завершения коммуникации. Флаги занимают 3 последовательных регистра. Сбрасываются флаги пользователем. См. таблицу ниже.

Операнд	Описание
<b>D</b>	Флаг удачного завершения коммуникации.
<b>D +1</b>	Ошибка приема данных.
<b>D +2</b>	Флаг паузы при приеме данных.

ПРИМЕЧАНИЕ: Включается только один флаг, соответствующие специальные флаги (SM) также будут включены при завершении связи. См. Раздел 6.19.3 по применению специальных флагов.

4. Время отправки команды коммуникации - это начало работы инструкции. Пользователь должен отключить инструкцию после завершения коммуникации. Затем может быть запущена следующая инструкция.
5. Последовательность действий и последовательность управления инструкцией аналогичны инструкции MOD1W API1808. Единственная разница между ними заключается в том, что отправка команды связи может быть выполнена без контроля пользователем флага отправки данных.
6. Эта инструкция не может использоваться в языке программирования ST, для прерывания или в разово назначенных функциональных блоках.

**Пример:**

Сравним инструкции MODRW и MODRWE, используя коммуникационный порт ПЛК COM1 и функциональный код 03, для считывания 8 фрагментов данных из регистра D20 ПЛК DVP-ES2. Аналогично для других коммуникационных портов ПЛК. Для использования других функциональных кодов см. описание инструкции API1808 MODRW и следующий пример.

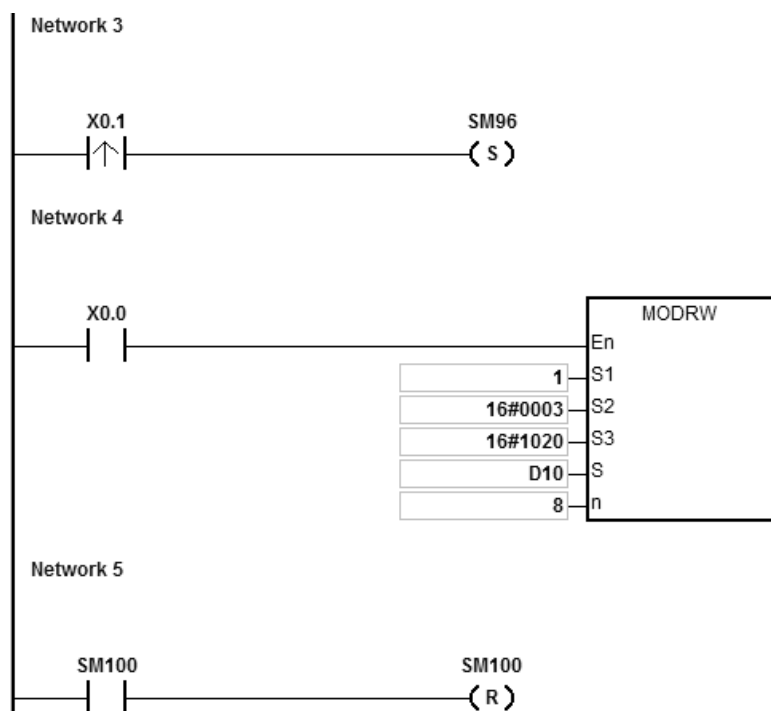
1. Адрес объекта D20 в ЦПУ DVP-ES2: 16#1020 и содержимое в D20~D27 показано ниже.

Объект	D20	D21	D22	D23	D24	D25	D26	D27
Значение (16#)	1234	5678	1122	3344	5566	7788	99AA	BBCC

2. ПЛК серии AS считывает содержимое из D20~D27 в ЦПУ ПЛК DVP-ES2 посредством коммуникации.

Метод 1: Применение инструкции MODRW

Данные в D20~D27 DVP-ES2 считываются, когда включены флаг SM96 и контакт X0.0.



Описание операндов инструкции MODRW

Операнд	Описание	Объект
S <sub>1</sub>	Адрес устройства	16#0001
S <sub>2</sub>	Функциональный код	16#0003
S <sub>3</sub>	Считывание адреса объекта D20	16#1020
S	Начальный регистр сохранения считанных данных	D10
n	Считывание длины данных	8

Отклик коммуникации между AS и DVP-ES2

**Режим ASCII:** (Коды ASCII необязательно принудительно преобразовывать, они выражены в значениях в формате 16#)

- AS отправляет команду связи: "01 03 10 20 00 08 C4 CR LF"
- AS принимает команду связи: "01 03 10 12 34 56 78 11 22 33 44 55 66 77 88 99 AA BB CC AA CR LF"

**Режим RTU:**

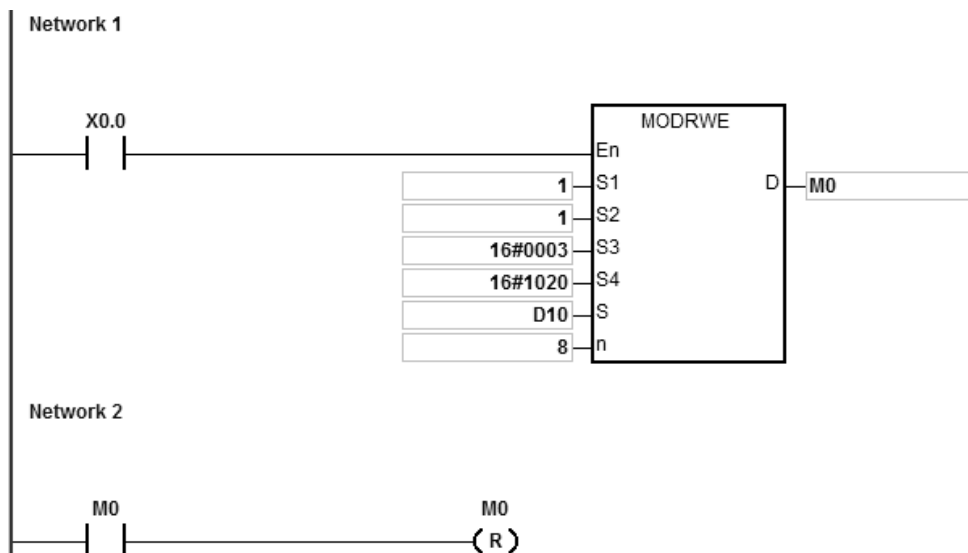
- AS отправляет команду связи: “ 01 03 10 20 00 08 41 06”
- AS принимает команду связи: “01 03 10 12 34 56 78 11 22 33 44 55 66 77 88 99 AA BB CC 90 FE”

При отсутствии ошибок связи включается флаг SM100.

После приема данных от ПЛК DVP-ES2 будет проверена корректность принятых данных. Если данные корректны, включается флаг SM100. Если данные неправильны, включается флаг ошибки SM102. В случае отсутствия отклика включается флаг паузы SM104.

**Метод 2: Применение инструкции MODRWE**

Данные в D20~D27 ПЛК DVP-ES2 считываются при включении контакта X0.0.



Описание операндов MODRWE

Операнд	Описание	Объект
S1	Номер коммуникационного порта	16#0001
S2	Адрес устройства	16#0001
S3	Функциональный код	16#0003
S4	Считывание адреса объекта D20	16#1020
S	Начальный регистр сохранения считанных данных	D10
n	Считывание длины данных	8
D	Флаги приема/передачи данных	M0

Отклик коммуникации между AS и DVP-ES2

**Режим ASCII:** (Коды ASCII необязательно принудительно преобразовывать, они выражены в значениях в формате 16#)

- AS отправляет команду связи: "01 03 10 20 00 08 C4 CR LF"
- AS принимает команду связи: "01 03 10 12 34 56 78 11 22 33 44 55 66 77 88 99 AA BB CC AA CR LF"

**Режим RTU:**

- AS отправляет команду связи: "01 03 10 20 00 08 41 06"
- AS принимает команду связи: "01 03 10 12 34 56 78 11 22 33 44 55 66 77 88 99 AA BB CC 90 FE"

При отсутствии ошибок связи включается M0.

После приема данных от ПЛК DVP-ES2 будет проверена корректность принятых данных. Если данные корректны, включается M0. Whereas, Если данные неправильны, включается M1. В случае отсутствия отклика включается M2. Соответствующие флаги SM100, SM102 и SM104 также включены.

### 3. Содержимое в D10~D17 ЦПУ AS CPU:

Объект	D10	D11	D12	D13	D14	D15	D16	D17
Значение (16#)	1234	5678	1122	3344	5566	7788	99AA	BBCC

**Примечание:**

1. Если операнд **D** задается через ISPSOft, тип данных: ARRAY [3] для BOOL.
2. Если значение в **D+2** превышает допустимое, инструкция выполняться не будет, включается флаг ошибки SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнды						Функция					
1818	DNETRW			S <sub>1</sub> , S <sub>2</sub> , S <sub>3</sub> , S <sub>4</sub> , S <sub>5</sub> , S <sub>6</sub> , S <sub>7</sub> , S <sub>8</sub> , S <sub>9</sub> , S <sub>10</sub> , D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> , D <sub>5</sub>						Чтение и запись данных коммуникации DeviceNet					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●				○	○		
S <sub>3</sub>								●	●				○	○		
S <sub>4</sub>								●	●				○	○		
S <sub>5</sub>								●	●				○	○		
S <sub>6</sub>								●	●				○	○		
S <sub>7</sub>								●	●				○	○		
S <sub>8</sub>								●								
S <sub>9</sub>								●	●				○	○		
S <sub>10</sub>								●	●				○	○		
D <sub>1</sub>		●	●	●												
D <sub>2</sub>		●	●	●												
D <sub>3</sub>								●								
D <sub>4</sub>								●								
D <sub>5</sub>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
S <sub>4</sub>		●			●	●							
S <sub>5</sub>		●			●	●							
S <sub>6</sub>		●			●	●							
S <sub>7</sub>		●			●	●							
S <sub>8</sub>		●			●	●							
S <sub>9</sub>		●			●	●							
S <sub>10</sub>		●			●	●							
D <sub>1</sub>	●												
D <sub>2</sub>	●												
D <sub>3</sub>		●			●	●							
D <sub>4</sub>		●			●	●							
D <sub>5</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символ:**

DNETRW	
En	
S1	D1
S2	D2
S3	D3
S4	D4
S5	D5
S6	
S7	
S8	
S9	
S10	

- S<sub>1</sub>** : Номер модуля, отправляющего данные по коммуникации DeviceNet
- S<sub>2</sub>** : DeviceNet MAC ID
- S<sub>3</sub>** : Служебный код
- S<sub>4</sub>** : Класс ID
- S<sub>5</sub>** : Идентификатор ID
- S<sub>6</sub>** : Атрибут ID
- S<sub>7</sub>** : Длина записанных данных
- S<sub>8</sub>** : Операнд сохранения записанных данных
- S<sub>9</sub>** : Время паузы коммуникации
- S<sub>10</sub>** : Количество повторных передач
- D<sub>1</sub>** : Флаг выполнения
- D<sub>2</sub>** : Флаг ошибки
- D<sub>3</sub>** : Код ошибки
- D<sub>4</sub>** : Длина считываемых данных
- D<sub>5</sub>** : Операнд сохранения считанных данных

**Описание:**

- S<sub>1</sub>** – это номер модуля справа от ПЛК. Первый – номер 1, второй - номер 2 и т. д. Все модули справа от ПЛК должны быть пронумерованы. Диапазон настройки составляет 1~32. Если значение настройки превышает диапазон (<1 или > 32), инструкция будет работать с минимальным или максимальным значением соответственно.
- S<sub>2</sub>** – это идентификатор (ID) Mac DeviceNet? диапазон: 0 ~ 63. Это может быть идентификатор Mac для Slave устройства, который Master устройство будет считывать и записывать, а также идентификатор Mac Master устройства, при этом чтение и запись данных будет происходить в Master устройстве.
- S<sub>3</sub>** является служебных кодом DeviceNet:

Служебный код	Описание
0x01	Чтение всех атрибутов (Get_Attribute_All)
0x02	Задание всех атрибутов (Set_Attribute_All)
0x0E	Чтение одиночного атрибута (Get_Attribute_Single)
0x10	Задание одиночного атрибута (Set_Attribute_Single)

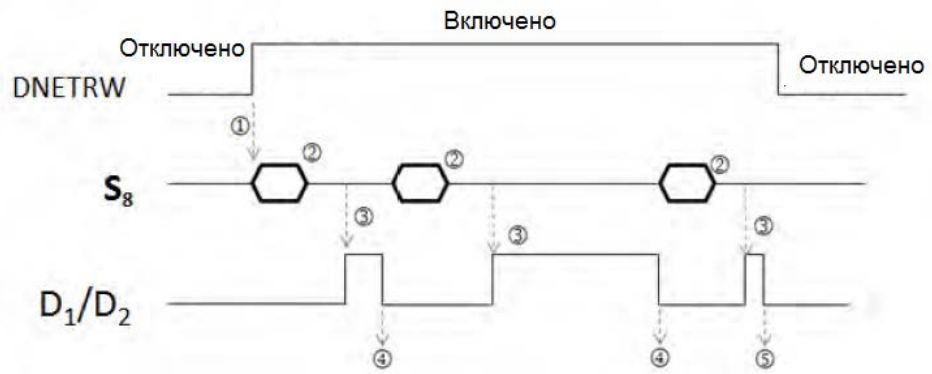
- S<sub>4</sub>**, **S<sub>5</sub>** и **S<sub>6</sub>** – это соответственно класс, идентификатор и атрибут ID для указания пути параметров в EPATH протокола DeviceNet.

- 5. **S<sub>7</sub>** – это длина записанных данных в байтах.
- 6. **S<sub>8</sub>** – это начальный адрес объекта, где записанные данные хранятся в порядке от младшего до старшего бита.
- 7. **S<sub>9</sub>** – это пауза коммуникации. Диапазон: 1~100, ед.изм.: 0,1 сек.
- 8. **S<sub>10</sub>** – количество повторных передач, диапазон настройки: 0 ~ 3. Когда возникает пауза коммуникации, данные будут передаваться снова.
- 9. **D<sub>3</sub>** отображает коды ошибок:

Код ошибки		Описание
Код 1 (старший байт)	Код 2 (младший байт)	
XX	FF	Не соответствует стандарту DeviceNet
20	01	Целевое Slave устройство не существует
20	02	Соединение с Slave устройством не установлено
20	03	Не удалась отправка сообщения
16	00	Превышение времени ожидания при паузе связи

- 10. **D<sub>4</sub>** – это длина считанных данных в байтах.
- 11. **D<sub>5</sub>** – это начальный адрес объекта, где считанные данные хранятся в порядке от младшего до старшего бита.
- 12. **D<sub>1</sub>** – флаг завершения связи, а **D<sub>2</sub>** – флаг ошибки. См. подробную процедуру отправки сообщений и временную диаграмму:

- ① Инструкция DNETRW запускается в первый раз, команда будет отправлена немедленно, если инструкция не будет занята другой связью DeviceNet.
- ② Команда отправляется.
- ③ Передача данных завершена, в соответствии с ответом включается флаг завершения или флаг ошибки.
- ④ Следующее сообщение готово к отправке. Следующая команда отправляется сразу после того, как флаг завершения или флаг ошибки будет сброшен.
- ⑤ Передача команды завершена, и инструкция DNETRW отключена.



13. Эта инструкция не может использоваться в языке программирования ST, задачах прерывания или однократно назначенных функциональных блоках.



API	Код инструкции			Операнды							Функция						
1819	CANRS			S <sub>1</sub> , S <sub>2</sub> , S <sub>3</sub> , S <sub>4</sub> , D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub>							Пользовательская настройка коммуникации по CAN, прием и отправка данных						

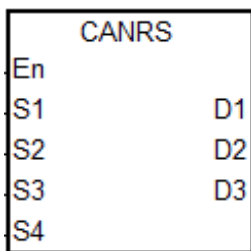
Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●							
S <sub>3</sub>								●	●				○	○		
S <sub>4</sub>								●	●							
D <sub>1</sub>								●								
D <sub>2</sub>		●	●	●				●								
D <sub>3</sub>		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
S <sub>4</sub>		●			●	●							
D <sub>1</sub>		●			●	●							
D <sub>2</sub>	●												
D <sub>3</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

6

**Символ:**



- S<sub>1</sub> : Задание режима коммуникации
- S<sub>2</sub> : ID коммуникации (MsgID)
- S<sub>3</sub> : Длина отправленных данных
- S<sub>4</sub> : Стартовый операнд сохранения отправленных данных
- D<sub>1</sub> : Стартовый операнд сохранения принятых данных
- D<sub>2</sub> : Флаг завершения связи
- D<sub>3</sub> : Флаг ошибки связи

**Описание:**

1. Инструкция CANRS применима к ПЛК серии AS с прошивкой версии V1.04.00 и выше.
2. Перед выполнением инструкции CANRS убедитесь, что HWCONFIG в конфигурации оборудования выбрал CANopen DS301 в качестве режима для платы расширения 2 (Function Card 2).
3. S<sub>1</sub> устанавливает режим коммуникации. Ниже представлены режимы, поддерживаемые инструкцией. Если значение параметра неверно, включается флаг ошибки D<sub>3</sub>, код ошибки SR659 устанавливается на 1.

Коммуникационный код		Заданное значение $S_1$	Формат коммуникации	Описание
Старш. 8 бит	Младш. 8 бит			
0	0	16#0000	2.0A 11-бит ID	Master-режим. Master устройство будет ждать получения ответного сообщения после отправки данных связи.
	1	16#0001		Master-режим. После отправки команды широковещательной передачи Master устройство переходит в режим приема сообщений от нескольких Slave устройств до тех пор, пока не возникнет пауза приема или количество принятых сообщений не превысит 100.
	2	16#0002		Режим Slave. Slave устройство сначала получает сообщение, а затем отправляет ответное сообщение.
	3	16#0003		Режим Slave. Slave устройство только получает сообщение.
1	0	16#0100	2.0B 29-бит ID	Master-режим. Master устройство будет ждать получения ответного сообщения после отправки данных связи.
	1	16#0101		Master-режим. После отправки команды широковещательной передачи Master устройство переходит в режим приема сообщений от нескольких Slave устройств до тех пор, пока не возникнет пауза приема или количество принятых сообщений не превысит 100.
	2	16#0102		Режим Slave. Slave устройство сначала получает сообщение, а затем отправляет ответное сообщение.
	3	16#0103		Режим Slave. Slave устройство только получает сообщение.

4.  $S_2$  – идентификатор переданного сообщения (ID коммуникации (MsgID)). В соответствии с протоколом 2.0A или 2.0B передаваемые данные автоматически занимают регистры буфера D.

Когда выбран формат 2.0A,  $S_2$  составляет 11 бит идентификационного кода со следующим форматом передачи данных:

№ $S_2$	$S_2$
Описание	Msg. ID

Когда выбран формат 2.0B,  $S_2$  (младш. слово) и  $S_{2+1}$  (старш. слово) являются 29 битами идентификационного кода:

№ $S_2$	$S_2$	$S_{2+1}$
Описание	Msg. ID (младш. слово)	Msg. ID (старш. слово)

5.  $S_3$  – длина передаваемого сообщения. Значение настройки должно быть в диапазоне 0 ~ 8 с единицей байт (8 бит). Если значение настройки выходит за пределы диапазона (<0 или > 8), инструкция будет работать с минимальным значением 0 или максимальным 8. Если длина переданного сообщения равна 0, режим связи автоматически переключается в Slave режим (принимать сообщения без отправки каких-либо данных). Этот режим можно использовать для контроля пакета коммуникации.
6.  $S_4$  является стартовым объектом, в котором сохраняются переданные данные и он использует только

следующие 8 бит данных.

Например, с D10 в качестве стартового объекта передаются 4 сообщения. Последовательность передачи данных показана ниже:

№ S <sub>4</sub>	D10	D11	D12	D13
Описание	Данные 1	Данные 2	Данные 3	Данные 4

7. Если S<sub>1</sub> определяет Master-режим, в котором Master устройство будет ожидать приема данных после отправки или Slave-режим, когда принятые данные будут непосредственно сохранены в объекте D<sub>1</sub>. D100 указан в D<sub>1</sub>. См. формат сохраненного содержимого:

Режим 2.0A:

№ D <sub>1</sub>	D100	D101	D102 ~ D109 (младш. 8 бит)
Описание	Msg. ID	Длина данных	Данные 1 ~ Данные 8

Режим 2.0B:

№ D <sub>1</sub>	D100	D101	D102	D103 ~ D110 (младш. 8 бит)
Описание	Msg. ID (младш. слово)	Msg. ID (старш. слово)	Длина данных	Данные 1 ~ Данные 8

Примечание. Если принимаемый Msg.ID необходимо указать на этапе приема данных, предварительно задайте значение D<sub>1</sub> / D<sub>1</sub>+1 в режимах 2.0A / 2.0B. Если Msg.ID не указан, перед приемом данных сбросьте значение D<sub>1</sub> / D<sub>1</sub>+1 на 0.

8. Если S<sub>1</sub> является Master-режимом с широковещанием, принятые данные будут непосредственно сохранены в объекте D<sub>1</sub>. D100 указан в D<sub>1</sub>. См. формат сохраненного содержимого:

Режим 2.0A: (Прием данных из двух Slave устройств. Для других данных увеличьте номер устройства, указанный в D<sub>1</sub>):

Последовательность отклика	Данные от 1 Slave устройства				Данные от 2 Slave устройства		Данные от 3 Slave устройства
	№ D <sub>1</sub>	D100	D101	D102	D103 ~ D110 (младш. 8 бит)	D111	D112~D120
Описание	Номер полученного пакета	Msg. ID	Длина данных	Данные 1 ~ Данные 8	Msg. ID	Длина, данные	Длина, данные, ID

Режим 2.0B: (Прием данных из двух Slave устройств. Для других данных увеличьте номер устройства, указанный в  $D_1$ ):

Последовательность отклика		Данные от 1 Slave устройства				Данные от 2 Slave устройства
№ $D_1$	D100	D101	D102	D103	D104 ~ D111 (младш. 8 бит)	D112~D122
Описание	Номер полученного пакета	Msg. ID (младш. слово)	Msg. ID (старш. слово)	Длина данных	Данные 1 ~ Данные 8	Длина, данные, ID

9.  $D_2$  и  $D_3$  – это флаг завершенности приема и флаг ошибки соответственно. В зависимости от текущего состояния связи при сканировании инструкции эти два флага будут определять, завершено ли получение данных или возникла ошибка. Если получение данных будет завершено или произойдет ошибка, будет включен соответствующий флаг. При каждом повторном включении инструкции оба флага будут очищены и автоматически выключатся.
10. Когда инструкция установлена в Master-режим, рекомендуется использовать настройку времени ожидания приема в HWCONFIG. Таким образом, если пакет связи не был получен полностью в течение указанного периода времени, включится  $D_3$  и в SR659код будет записан ошибки 2, Если время ожидания приема установлено на 0, это означает, что пауза при связи не ограничена и статус может быть применен к Slave-режиму.

Примечание. Если установлен метод, в котором нет ограничения на паузу связи, право управления коммуникацией не может быть изменено до тех пор, пока инструкция не будет отключена вручную.

11. Количество инструкций в программе не ограничено. Но каждый раз может быть отправлена только одна команда связи по CAN. Если команда отправляется или принимается в настоящее время, следующая инструкция CANRS не может быть включена в это время.
12. Формат CAN BUS и каждый бит контента для Msg. ID объясняется ниже:

При выборе протокола 2.0A, значение  $S_2$  равно 16#0123, содержимое Msg. ID показано ниже:

№ бита	15 ~ 11	10 ~ 8	7 ~ 4	3 ~ 0
Значение $S_2$ (16 бит)	-	1	2	3

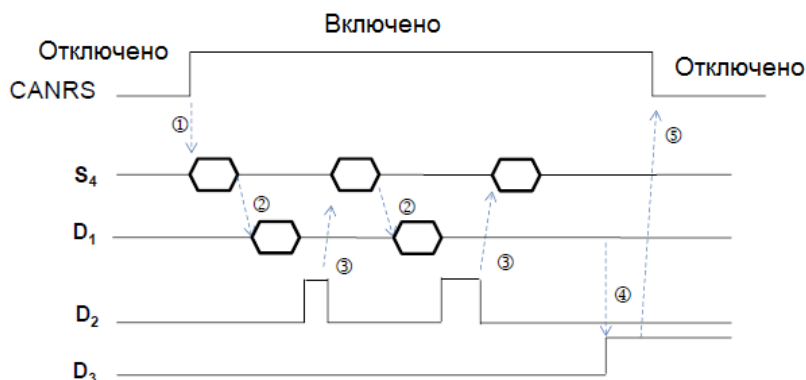
При выборе протокола 2.0B, значения  $S_2$  равно 16#1234 (младш. слово) и  $S_2+1$  равно 16#0567 (старш. слово), содержимое Msg. ID показано ниже:

№ бита	31 ~ 29	28	27 ~ 24	23 ~ 20	19 ~ 16	15 ~ 0
Значение $S_2$ (32 бит)	-	0	5	6	7	1234

13. Если операнд  $D_1$  задан как переменная, рекомендуется использовать матрицу словного типа.

14. Ниже см. Диаграмму последовательности связи CANRS и пояснения к ней:

- ① Инструкция CANRS включена. Если никакая другая инструкция CANRS не имеет права на управление коммуникацией, флаги D<sub>2</sub> и D<sub>3</sub> сбрасываются, затем данные сразу отправляются.
- ② Передача данных завершена, затем начинается прием данных. По завершении приема включается флаг завершения D<sub>2</sub>.
- ③ Инструкция отправляет данные снова после сброса флага D<sub>2</sub>.
- ④ При возникновении паузы при приеме, включается флаг D<sub>3</sub>.
- ⑤ Если возникла ошибка, вы можете отключить инструкцию. Инструкцию можно запустить заново после сброса флага D<sub>3</sub>.



**Пример 1:**

**Master-режим**

Используя инструкцию CANRS, 6-байтовые данные в D10 ~ D15 передаются на Slave устройство, а данные ответа от Slave устройства хранятся в операндах, начиная с D100. Когда передача и прием завершены включается M100.

Описание соответствующих параметров:

Master mode_MsgID=2	
Режим коммуникации	0 = Master-режим, ожидает получения ответа от Slave устройства после отправки данных
ID коммуникации	MsgID отправленных данных, D0=1
Длина отправленных данных	6 байт
Начальный операнд, в котором сохраняются отправленные данные	D10
Начальный операнд, в котором сохраняются принятые данные	D100 сохраняет MsgID, D101 сохраняет число принятых пакетов D102.... сохраняют принятые данные

Флаг завершения коммуникации	M100
Флаг ошибки коммуникации	M101

1. Когда M1 включен, установите MsgID отправленного сообщения как 1 и MsgID принятого сообщения как 2. Затем начинается отправка данных.

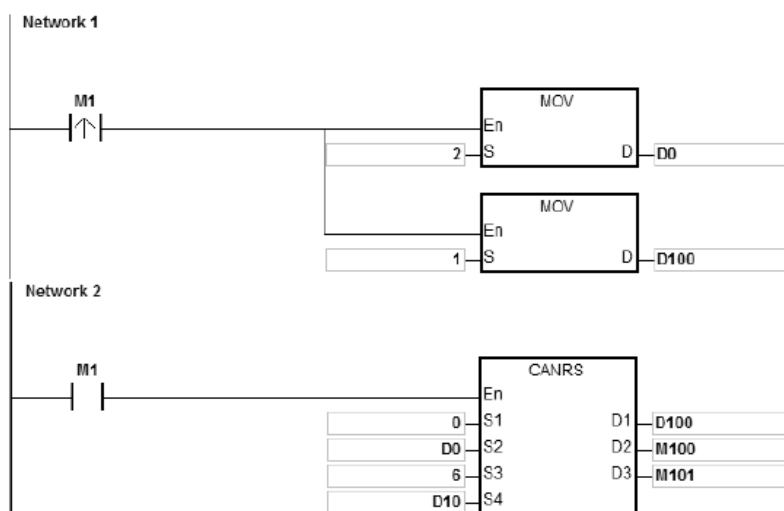
A. Выполняется отправка данных в D10 ~ D15 (6 байт) на Slave устройство.

Операнд	D10 ~ D15
Содержимое	H0A (определяется пользователем)

B. Полученные данные ответа от Slave устройства сохраняются в D100 ~ D109. После приема включается M100.

Операнд	D100	D101	D102 ~ D109
Содержимое	2	8	H0B

2. Пример программы Master-режима:



**Пример 2:**

**Slave-режим**

Используя инструкцию CANRS, полученные данные от ведущего устройства сохраняются в операндах, начиная с D120, и 8-байтовые данные в D20 ~ D27 отправляются обратно. Когда передача и прием завершены, включается M110.

Описание соответствующих параметров:

Master mode_MsgID=1	
Режим коммуникации	2 = Slave режим, отправляет данные после приема данных

ID коммуникации	MsgID отправленных данных, D0=2
Длина отправленных данных	8 байт
Начальный операнд, в котором сохраняются отправленные данные	D20
Начальный операнд, в котором сохраняются принятые данные	D120 сохраняет MsgID, D121 сохраняет число принятых пакетов D122.... сохраняют принятые данные
Флаг завершения коммуникации	M110
Флаг ошибки коммуникации	M111

1. Когда M1 включен, установите MsgID отправленного сообщения как 1 и MsgID принятого сообщения как 2. Затем начинается отправка данных.

A. Выполняется отправка данных в D120 ~ D127 (6 байт) на Slave устройство.

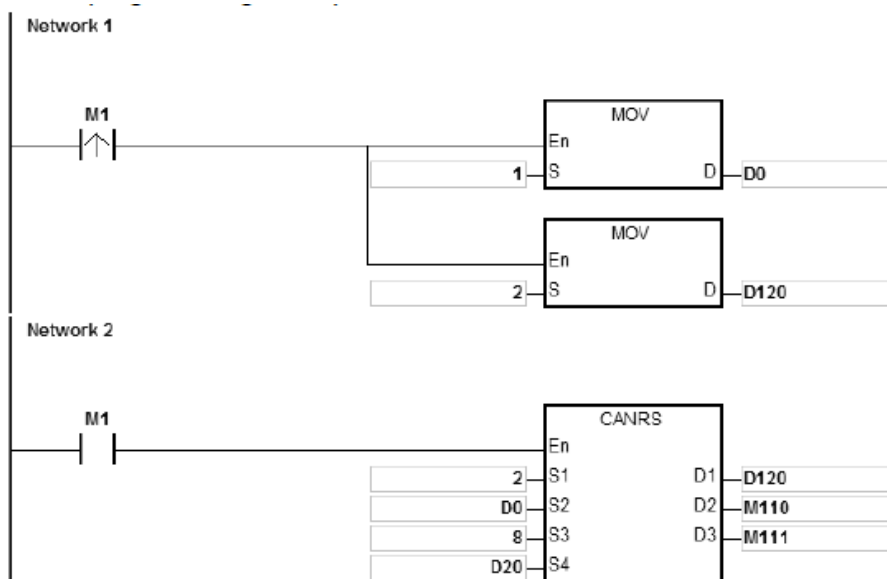
Операнд	D120	D121	D122 ~ D179
Содержимое	1	6	H0A

B. Затем 8-байтовые данные в D20 ~ D27 начинают пересылаться обратно Master устройству. После выполнения отправки данных включается M110.

Операнд	D10 ~ D15
Содержимое	H0B (определяется пользователем)

6

2. Пример программы Slave-режима:



API	Код инструкции		Операнды					Функция										
1820		DMVSH																Разрешение обнаружения и коммуникации для Delta DMV

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Mode								●					○	○		
Start1		●	●	●				●								
Start2		●	●	●				●								
Ready	●															
ComNo								●					○	○		
Id_Ip								●					○	○		
Address								●					○	○		
Length								●					○	○		
Shoot1		●														
Shoot2		●														
RdData								●								
Ok		●	●	●				●								
Err		●	●	●				●								
ErrCode								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Mode		●											
Start1	●												
Start2	●												
Ready	●												
ComNo		●			●	●							
Id_Ip		●			●	●							
Address		●			●	●							
Length		●			●	●							
Shoot1	●												
Shoot2	●												
RdData		●			●	●							
Ok	●												
Err	●												
ErrCode		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-



**Символ:**

DMVSH	
En	
Mode	Shoot1
Start1	Shoot2
Start2	RdData
Ready	Ok
ComNo	Err
Id_Ip	ErrCode
Address	
Length	

- Mode** : Метод запуска и приема
- Start1** : Триггерный сигнал для старта, настройка 1
- Start2** : Триггерный сигнал для старта, настройка 2
- Ready** : Сигнал завершения приема данных ответа от DMV
- ComNo** : Последовательный порт или сетевой порт связи ПЛК для отправки или получения данных
- Id\_Ip** : Адрес последовательной связи или сетевой IP-адрес DMV
- Address** : Стартовый операнд, в котором сохраняются данные DMV для считывания
- Length** : Длина данных для считывания
- Shoot1** : Выходной сигнал, когда ПЛК уведомляет DMV, чтобы включить обнаружение, настройка 1
- Shoot2** : Выходной сигнал, когда ПЛК уведомляет DMV, чтобы включить обнаружение, настройка 2
- RdData** : Стартовый операнд, в котором сохраняются данные, которые ПЛК получает от DMV
- Ok** : Флаг завершения связи
- Err** : Флаг ошибки связи
- ErrCode** : Код ошибки связи

**Описание:**

1. Инструкция DMVSH применима к ПЛК серии AS с прошивкой версии V1.06.00 и выше.
2. **Mode** устанавливает метод, посредством которого ПЛК запускает контроллер машинного зрения DMV, включая DMV1000 и DMV2000, и метод приема данных. Есть два режима: 0 и 1. Если настройка превышает допустимые значения, ПЛК будет автоматически запускаться в режиме 0.
3. Пожалуйста, обратитесь к следующему примеру, объясняющему функции **Start1**, **Start2**, **Ready**, **Shoot1**, **Shoot2**, **RdData** и **Ok** и процесс запуска обнаружения.
4. **ComNo** устанавливает номер порта связи ПЛК. Число 1 представляет COM1, номер 2 представляет COM2, номер 11 представляет Card1, номер 12 представляет Card2, а номер 4 представляет порт Ethernet. Если параметр не является одним из указанных выше номеров или представляет собой порт связи, который ПЛК не поддерживает, включается флаг ошибки **Err** и код ошибки в **ErrCode** будет равен 16#0001.
5. **Id\_Ip** устанавливает адрес станции связи (Mac ID) Slave DMV или сетевой IP. **Address** – это адрес связи, по которому считывается результат обнаружения DMV. **Length** – это длина данных обнаружения считывания.
6. Коды ошибок **ErrCode** приведены в таблице ниже.

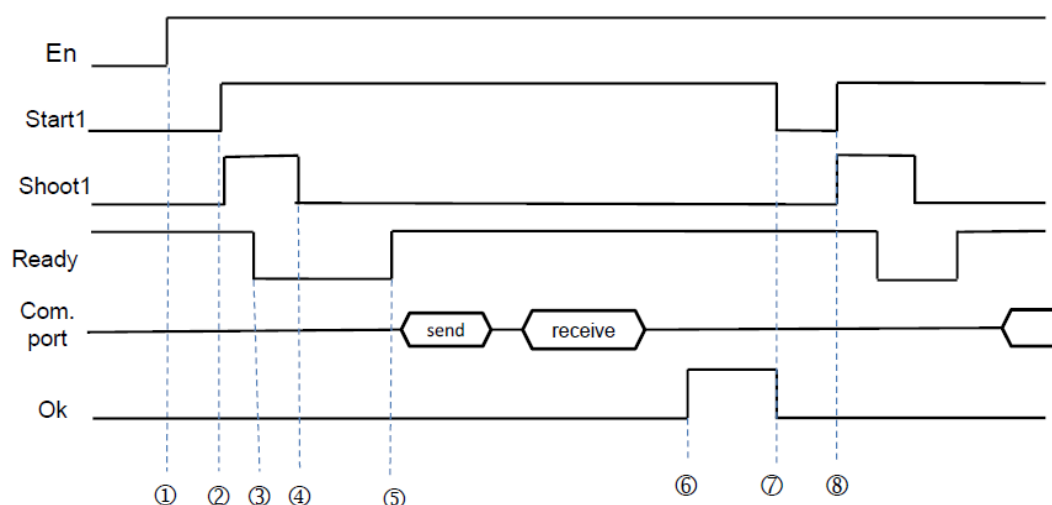
Код ошибки	Описание	Метод устранения
16#0001	Указанный порт связи неверен	Укажите другой порт связи

16#0002	Ошибка ответа DMV	Проверьте адрес связи
16#0003	Нет ответа от DMV	Проверьте правильность подключения кабеля или контроллера DMV
16#0004	Ожидание ответа от DMV; обнаружение DMV не может быть активировано повторно	Измените время активации обнаружения DMV

**Пример 1****Mode = 0**

Для запуска обнаружения DMV и приема данных ответа DMV используется только настройка 1.

См. диаграмму последовательности ниже:

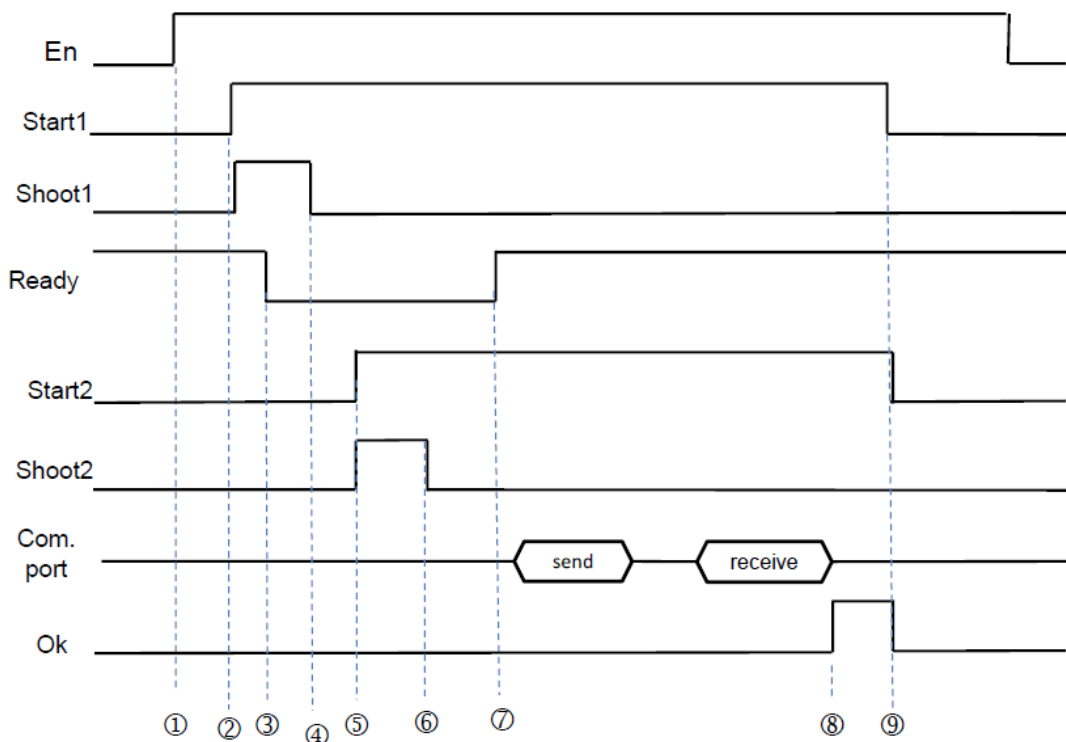


- ① Инструкция DMVSH запущена.
- ② Включение **Start1**, инструкция выдает выходной сигнал **Shoot1** (включен около 10 мс) и уведомляет DMV о включении функции обнаружения камеры.
- ③ После того, как DMV получает триггерный сигнал, **Ready** выключается.
- ④ ПЛК автоматически сбрасывает **Shoot1**.
- ⑤ После того, как обнаружение DMV закончено, **Ready** будет включен. ПЛК определяет включение режима **Ready** после ожидания в течение времени в 1,5 раза больше времени входной фильтрации. ПЛК получает сообщение о включении **Ready** и отправляет команду на считывание через Modbus 0x03.
- ⑥ Флаг **Ok** включается после того, как ПЛК принимает данные по коммуникации от контроллера DMV. Полученные на данный момент данные могут быть обработаны.
- ⑦ Флаги **Start1** и **Ok** можно сбросить вручную.
- ⑧ Если необходимо выполнить повторную проверку обнаружения, сразу включите **Start1** (Вернитесь к шагу 2, чтобы продолжить).

**Пример 2****Mode = 1**

Настройки 1 и 2 используются для запуска обнаружения DMV отдельно. Но команда чтения данных связи не может быть отправлена до тех пор, пока не включатся оба **Ready** (Пример применим к обнаружению DMV1000 с двумя камерами).

См. диаграмму последовательности ниже:



- ① Инструкция DMVSH запущена.
- ② Включение **Start1**, инструкция выдает выходной сигнал **Shoot1** (включен около 10 мс) и уведомляет DMV о включении функции обнаружения первой камеры.
- ③ После того, как DMV получает триггерный сигнал, **Ready** выключается.
- ④ ПЛК автоматически сбрасывает **Shoot1**.
- ⑤ Включение **Start2**, инструкция выдает выходной сигнал **Shoot1** (включен около 10 мс) и уведомляет DMV о включении функции обнаружения второй камеры.
- ⑥ ПЛК автоматически сбрасывает **Shoot2**.
- ⑦ После того, как обнаружение DMV закончено, **Ready** будет включен. ПЛК определяет включение режима **Ready** после ожидания в течение времени в 1,5 раза больше времени входной фильтрации. ПЛК получает сообщение о

включении **Ready** и отправляет команду на считывание через Modbus 0x03.

⑧ Флаг **Ok** включается после того, как ПЛК принимает данные по коммуникации от контроллера DMV. Полученные на данный момент данные могут быть обработаны.

⑨ Флаги **Start1**, **Start2** и **Ok** можно сбросить вручную.

## 6.19.2 Описание флагов и регистров для коммуникации

### Флаги (SM)

Флаг				Описание	Действие
COM1	COM2	Card1	Card2		
SM96	SM97	SM76	SM77	Флаг запроса отправки данных Если используется инструкция для отправки и получения данных, необходимо использовать импульсную инструкцию, чтобы включить этот флаг. Когда инструкция выполняется, ПЛК отправляет и принимает данные. По завершении отправки данных система автоматически выключает флаг	Пользователь включает, а система выключает флаг автоматически
SM98	SM99	SM78	SM79	Когда флаг включен, ПЛК ожидает приема данных	Система включает и выключает флаг автоматически
SM100	SM101	SM80	SM81	Флаг завершения приема По завершении приема данных система автоматически включает флаг. Когда флаг включен, полученные данные могут быть обработаны. После завершения обработки полученных данных пользователь должен отключить флаг	Система включает флаг автоматически, пользователь выключает флаг
SM102	SM103	SM82	SM83	Флаг ошибки приема данных Ошибка при приеме данных с использованием команды приема данных.	Система включает флаг автоматически, пользователь выключает флаг
SM104	SM105	SM84	SM85	Флаг превышения паузы связи Если пользователи устанавливают паузу связи (в SR), и после периода ожидания данные не принимаются, флаг включен. После решения проблемы пользователь должен отключить флаг	Система включает флаг автоматически, пользователь выключает флаг
SM106	SM107	SM86	SM87	Переключение между 8-битным и 16-битным режимами ВКЛ: 8-битный режим ВЫКЛ: 16-битный режим	Пользователь включает и выключает флаг
SM210	SM212	-	-	Режим связи ВКЛ: режим RTU ВЫКЛ: режим ASCII Настройку можно выполнить в HWCONFIG ПО ISPSOFT, недоступна для Card1 и Card2.	Пользователь включает и выключает флаг
SM209	SM211	SM90	SM91	Флаг изменения протокола связи Протокол связи изменяется в соответствии с настройками в SR. Если флаг включен, протокол связи изменяется в соответствии с настройками в SR, а затем система автоматически отключает флаг. ПРИМЕЧАНИЕ: это изменение не повлияет на параметры, установленные в HWCONFIG. Когда ПЛК включится снова, он будет работать в соответствии с протоколом связи, установленным в HWCONFIG	Пользователь включает и выключает флаг

ПРИМЕЧАНИЕ: вышеупомянутые флаги являются энергозависимыми.

## Регистры (SR)

Специальные регистры данных				Описание
COM1	COM2	Card1	Card2	
SR201	SR202	SR176	SR178	Адрес коммуникационного порта
SR209	SR212	SR177	SR179	Протокол связи Для получения дополнительной информации см. таблицу ниже, чтобы настроить формат связи для порта последовательной коммуникации
SR210	SR213	SR182	SR183	Пауза коммуникации, ед. изм: мс Предположим, что значение настройки больше 0. Когда команда связи выполнена и ПЛК находится в состоянии приема, но данные не принимаются после истечения периода ожидания или время между приемом двух символов превышает настроенное время, включается флаг превышения паузы связи. Если настройка равна 0, мониторинг пауз связи будет отключен. В инструкции MODRW диапазон задания времени: 100 ~ 32767 (мс)

Настройка формата связи для последовательного коммуникационного порта

b0	Длина данных		7 (значение=0)		8 (значение=1)	
	b2~b1	Бит четности		00	:	Нет
01				:	Нечетный	
10				:	Четный	
b3	Стоп-бит		1 бит (значение=0)		2 бита (значение=1)	
b7~b4	0001	( 16#1 )	:	4800		
	0010	( 16#2 )	:	9600		
	0011	( 16#3 )	:	19200		
	0100	( 16#4 )	:	38400		
	0101	( 16#5 )	:	57600		
	0110	( 16#6 )	:	115200		
	0111	( 16#7 )	:	230400		Не применяется для RS-232
	1000	( 16#8 )	:	500000		Не применяется для RS-232
	1001	( 16#9 )	:	921000		Не применяется для RS-232
1111	( 16#F )	:	Пользовательская настройка*1			
b8~b15	Не используется (зарезервирован)					

\*1: Пользователь может настроить скорость обмена данными через HWCONFIG ПО ISPSOft.

Скорость обмена данными следующая

Скорость (бит/с)	Таймер паузы режима RTU (мс)	Скорость (бит/с)	Таймер паузы режима RTU (мс)
4800	9	115200	1
9600	5	230400	1
19200	3	-	-
38400	2	-	-
57600	1	-	-

## 6.20 Прочие инструкции

### 6.20.1 Описание прочих инструкций

API	Код инструкции		Операнды	Функция
1900	WDT	P	—	Сторожевой таймер

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	—

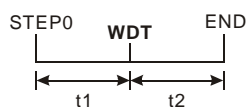
Символ:



Описание:

1. В ПЛК серии AS имеется сторожевой таймер, который используется для контроля работы системы.
2. Инструкция WDT используется для сброса сторожевого таймера в ПЛК. Если время сканирования программы превышает 200 миллисекунд, горит светодиодный индикатор ошибки и ПЛК прекращает работу.
3. Условия срабатывания сторожевого таймера:
  - Системный сбой
  - Исполнение программы занимает много времени, поэтому время сканирования больше, чем значение настройки сторожевого таймера. Есть два способа улучшения данной ситуации:

(a) Использование инструкции WDT

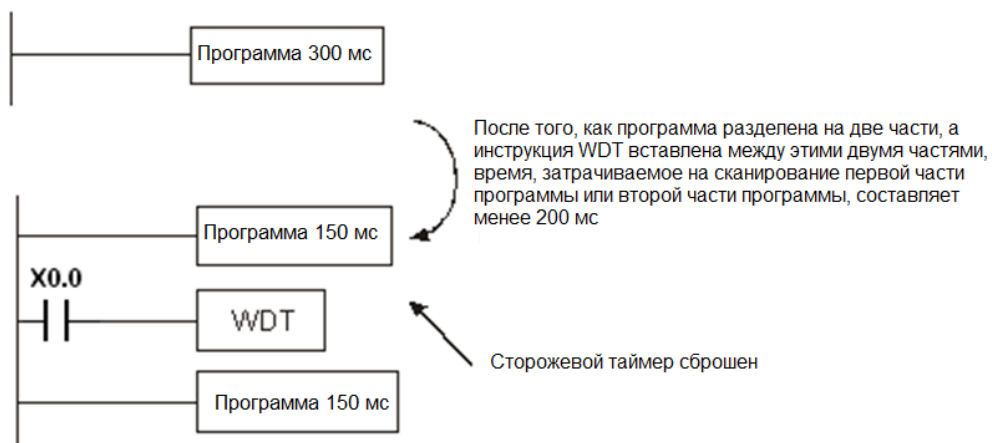


(b) См. Руководство для ПО ISPSOft для получения дополнительной информации об изменении значения настройки сторожевого таймера.



**Пример:**

Предположим, время сканирования программы 300 мс. После того, как программа разделена на две части, а инструкция WDT вставлена между этими двумя частями, время, затрачиваемое на сканирование первой части программы или второй части программы, составляет менее 200 мс.



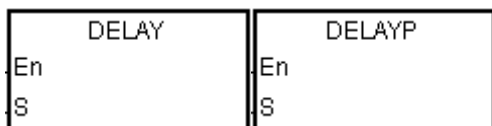
API	Код инструкции			Операнды							Функция						
1901		DELAY	P	S							Задержка выполнения программы						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	—

**Символ:**



**S** : Время задержки

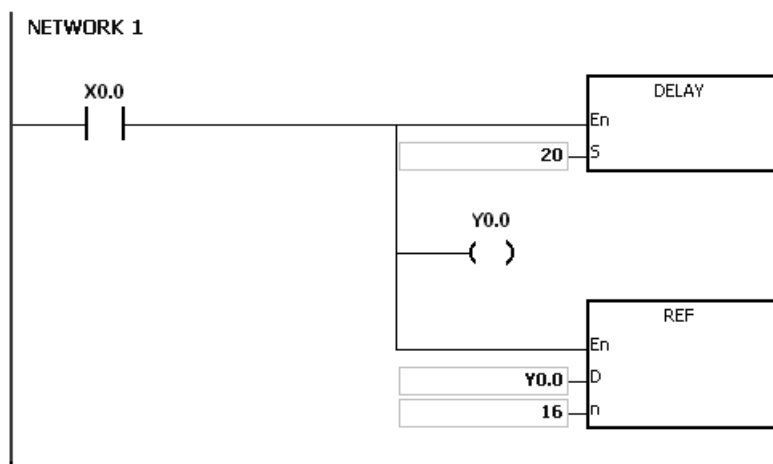
**Описание:**

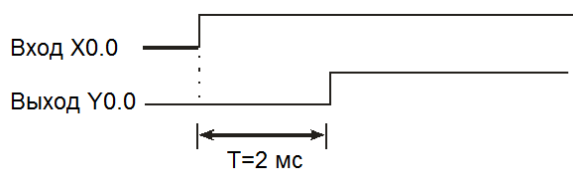
После выполнения инструкции DELAY выполнение программы после инструкции DELAY задерживается на время, указанное пользователем.

Шаг задания **S** равен 0,1 мс.

**Пример:**

Когда X0.0 включен, инструкция DELAY выполняется. Выполнение программы после инструкции DELAY задерживается на 2 мс. Когда Y0.0 включен, состояние выходов Y0.0~Y0.15 обновляется через два миллисекунды после выполнения инструкции DELAY.



**Примечание:**

1. Если значение **S** меньше 0, задержки не происходит.
2. Если значение **S** больше 1000, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
3. Пользователь может настроить задержку в соответствии с практическими условиями.
4. На увеличение задержки может повлиять связь или другие влияющие факторы.

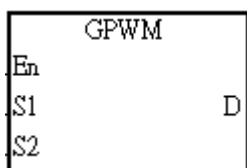
API	Код инструкции			Операнды								Функция					
1902		GPWM		$S_1 \cdot S_2 \cdot D$								Общая команда генерации импульсов ШИМ					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●			○	○				
$S_2$		●			●	●		●								
D		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

**Символ:**



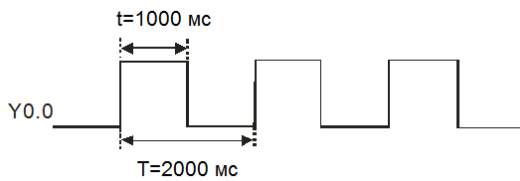
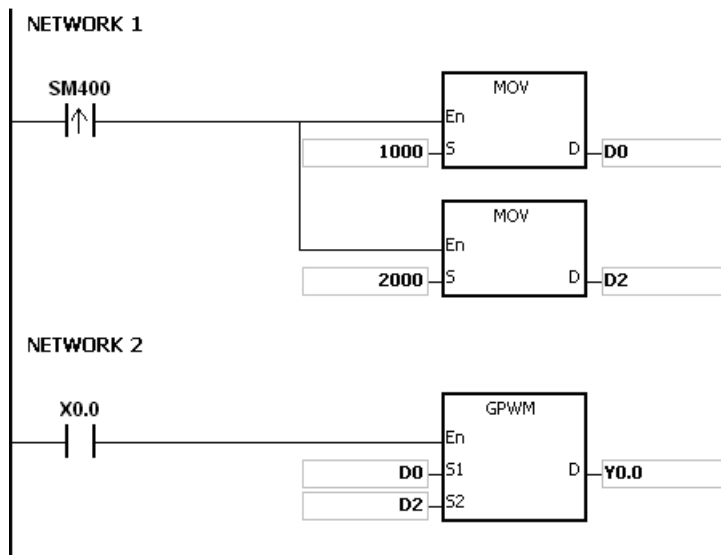
- $S_1$  : Ширина импульса
- $S_2$  : Период импульса
- D : Импульсный выход

**Описание:**

1. Когда команда GPWM выполняется, на импульсном выходе D генерируются импульсы шириной  $S_1$  с периодом  $S_2$ .
2. Ширина импульса в  $S_1$  задается в диапазоне от 0 до 3276 мс.
3. Период импульса задается в  $S_2$  как T. Значение T задается в диапазоне от 1 до 32767 мс.  $S_1$  должно быть меньше  $S_2$ .
4.  $S_{2+1}$  и  $S_{2+2}$  являются системными параметрами. Не редактируйте их.
5. Если значение  $S_1$  меньше 0, выходных импульсов нет. Если  $S_1$  больше  $S_2$ , выход постоянно включен.
6.  $S_1$  и  $S_2$  могут быть изменены во время выполнения инструкции GPWM.
7. Если условный контакт не включен, импульсный выход отсутствует.
8. Когда используется онлайн-редактирование, пожалуйста, выключите условный контакт, чтобы инициализировать инструкцию.

**Пример:**

Во время выполнения программы значения в D0 и D2 равны 1000 и 2000 соответственно. Когда X0.0 включен, показанные ниже импульсы выводятся на Y0.0. Когда X0.0 выключен, Y0.0 также выключен.



**Примечание:**

1. Инструкция рассчитывается по циклу сканирования. Поэтому максимальная ошибка – это один цикл сканирования. Кроме того,  $S_1$ ,  $S_2$  и  $(S_2 - S_1)$  должны быть больше, чем цикл сканирования. В противном случае при выполнении инструкции GPWM возникает ошибка.
2. Если инструкция используется в функциональном блоке или задаче прерывания, точность сигнала на импульсном выходе снижается.
3. Если операнд  $S_2$  задается в ПО ISPSOft, тип данных: ARRAY [3] для WORD/INT.

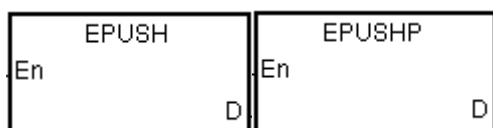
API	Код инструкции			Операнды								Функция					
1904		EPUSH	P	D								Сохранение содержимого индексных регистров					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
D								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	—

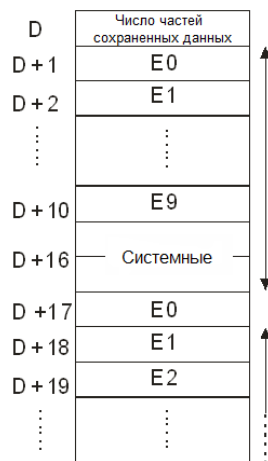
**Символ:**



**D** : Операнд сохранения значения индексных регистров

**Описание:**

1. Значения в E0~E9 сохраняются в **D**. Допустимые значения **D** находятся в диапазоне 0~99. Если значение превышает допустимое, инструкция не выполняется.
2. Для выполнения инструкции необходимо шестнадцать регистров, последние шесть регистров являются системными. Если инструкция выполнена и длина сохраненных данных в **D** равна n, данные в E0 ~ E9 сохраняются в  $D+(16*n+1)~D+(16*n+16)$ , значение в **D** становится n+1.
3. Диапазон номеров операнда **D** должен быть не более  $16 \times 100 + 1$ .
4. Эта инструкция, для взаимодействия со стеком, используется как импульсная и заносит значение в стек. Поэтому пользователю необходимо отключить контакт при следующей операции.
5. Если инструкция используется вместе с инструкцией EPOP, значение, которое хранится последним в регистре, указанном в **D**, считывается сначала, следуя принципу LIFO.



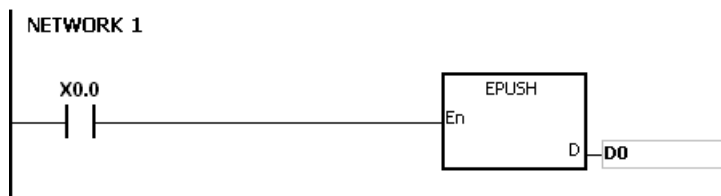
**Пример:**

Предположим, значение в D0 равно 0.

Когда X0.0 включается впервые, данные в E0~E9 передаются в D1~D10 и значение в D0 становится равным 1.

Когда X0.0 включается повторно, данные в E0~E9 передаются в D17~D26 и значение в D0 становится равным 2.

Когда X0.0 включается в n<sup>й</sup> раз, данные в E0~E9 передаются в (n\*16)+1~(n\*16)+10.



**Примечание:**

1. Если значение в **D** меньше 0 или больше 99, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значения регистра недостаточно для  $D + ((\text{значение в } D) + 1) * 16 - 1$ , инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.

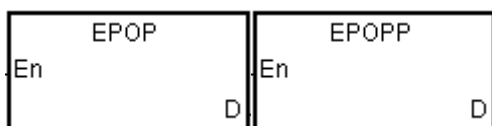
API	Код инструкции			Операнды								Функция					
1905		EPOP	P	<b>D</b>								Считывание данных в индексные регистры					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>D</b>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	—

**Символ**

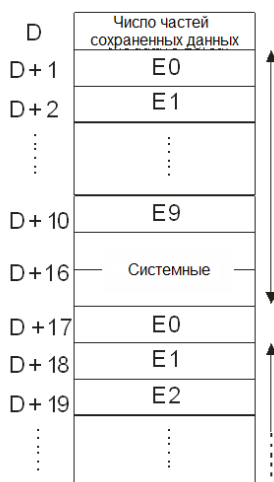


**D** : Операнд, из которого считывается значение

**Описание:**

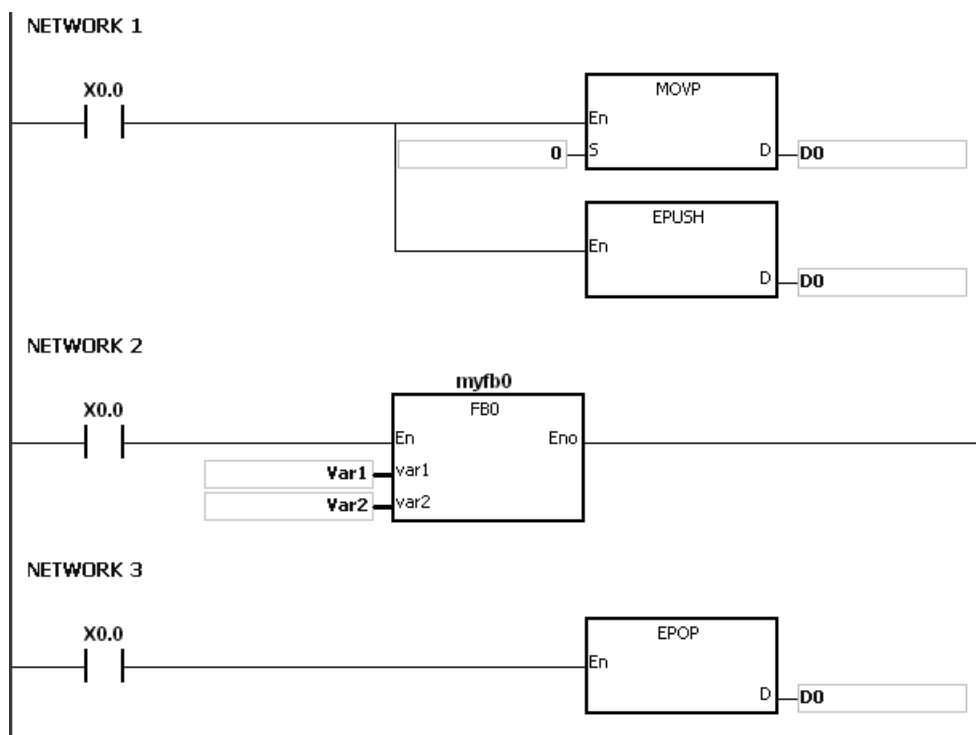
1. Значения в регистрах, заданных в операнде **D** считываются в E0~E9, и значение **D** уменьшается на единицу. Допустимые значения **D** находятся в диапазоне 1~100. Если значение превышает допустимое, инструкция не выполняется.
2. Для выполнения инструкции необходимо шестнадцать регистров, последние шесть регистров являются системными. Если инструкция выполнена и длина сохраненных данных в **D** равна n, данные в E0 ~ E9 сохраняются в  $D+16*(n-1)+1 \sim D+16*(n-1)+10$ , значение в **D** становится n-1.
6. Эта инструкция, для взаимодействия со стеком, используется как импульсная, забирая значение из стека и присваивая его указанной переменной. Поэтому пользователю необходимо отключить контакт при следующей операции.
7. Если инструкция используется вместе с инструкцией EPOP, значение, которое хранится последним в регистре, указанном в **D**, считывается сначала, следуя принципу LIFO.





**Пример:**

Когда X0.0 включен, значение в D0 задано как 0, значения в E0~E9 передаются в D1~D10. Когда выполнение FBO завершено, значения в D1~D9 считываются в E0~E9 с помощью инструкции EPOP.



**Примечание:**

1. Если значение в D меньше 1 или больше 100, инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значения регистра недостаточно для D+((значение в D) \*16-1), инструкция не выполняется, включается флаг SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнды								Функция					
1906		INFO	P	S, D								Считывание системных данных					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S													○			
D								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●				●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	—

## Символ

	INFO	
En		
S		D

	INFOP	
En		
S		D

S : Код чтения системных данных

D : Стартовый операнд, в котором сохраняются считанные системные данные

## Описание:

1. S представляет собой код чтения системной информации, коды перечислены в таблице ниже:

Код S	Описание	Количество последовательных D (число слов)	Примечание	Поддерживаемая версия прошивки
0	Чтение серийного номера устройства	9	Код ASCII	V1.04 и выше
1	Чтение идентификатора Ethernet Mac ID	3	Шестнадцатеричное значение	V1.04 и выше
2	Чтение полного времени подачи питания на устройство	2	32-битное значение, ед.изм.: минуты	V1.04 и выше
Другие	Зарезервированы	0	Инструкция не выполняется	-

2. Если серийный номер устройства отсутствует, это AS332T-AW16360001, код чтения равен 0, а данные считывания сохраняются в операндах, начиная с D0, серийный номер продукта сохраняется в порядке, представленным в таблице ниже:

D0	D1	D2	D3	D4	D5	D6	D7	D8
16#5341	16#3333	16#5432	16#412D	16#3157	16#3336	16#3036	16#3030	16#0031
'SA'	'33'	'T2'	'A-'	'1W'	'36'	'06'	'00'	'1'

3. Если идентификатор Mac ID равен 16#123456789ABC, код чтения равен 1, а данные считывания сохраняются в операндах, начиная с D10, данные идентификатора Mac ID сохраняются в порядке, представленным в таблице ниже:

D10	D11	D12
16#1234	16#5678	16#9ABC

4. Если полное время подачи питания на устройство равно 70 000 минут, код чтения равен 2, а данные считывания сохраняются в операндах, начиная с D20, данные полного времени подачи питания на устройство сохраняются в порядке, представленным в таблице ниже:

D20	D21
16#1170	16#0001

5. При выполнении условия по контакту действительно только первое выполнение инструкции INFO. Поэтому рекомендуется использовать импульсную инструкцию INFOP. Если инструкция INFO используется совместно с Н/О контактом, инструкция будет выполняться автоматически только один раз.

**Примечание:**

Если **D** задан как переменная и код чтения равен 0, 1 и 2, эти величины могут быть объявлены соответственно в ARRAY [9], [3], [2] типа WORD.

## 6.21 Инструкции по обработке строковых переменных

### 6.21.1 Описание инструкций по обработке строковых переменных

API	Код инструкции			Операнд				Функция			
2100	D	BINDA	P	S, D				Преобразование числа в десятичном коде со знаком в код ASCII			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●	●		●	●	●						
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

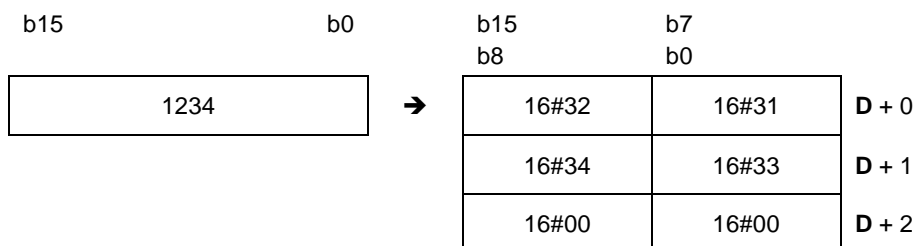
Символьное обозначение:



S : Исходное значение  
D : Регистр, в который записывается результат преобразования

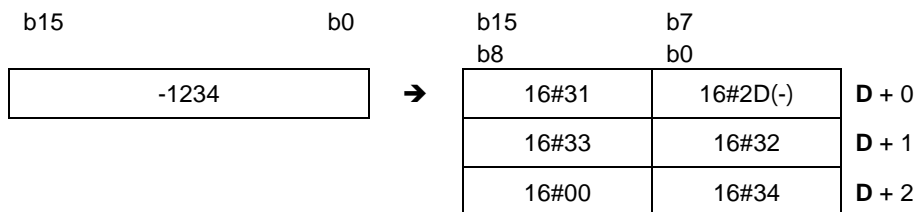
Описание:

1. Число в двоично-десятичном коде со знаком из операнда **S** преобразуется в код ASCII, и результат записывается в операнд **D**.
2. Инструкция поддерживает SM690, который управляет конечным символом.
3. Значение операнда **S** 16-битной инструкции должно быть двоичным числом в диапазоне от -32768 до 32767. Операнд **D** занимает четыре регистра слова. Данные преобразуются следующим образом.

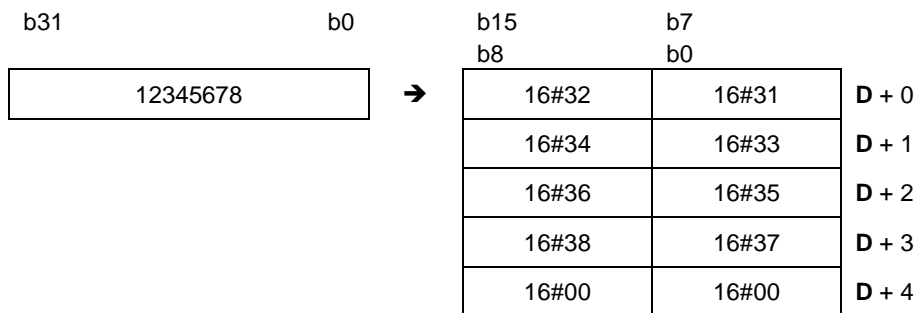


Если SM690 = OFF, символ конца 16#0000 записывается в D+2. Если SM690 = ON, значение D+2 остается без изменений, без символа конца.

Если значение операнда S положительное, в операнде D будет записываться только значение без символа знака. Если значение операнда S отрицательное, символ знака операнда D «-» будет записываться в 16#2D. Например, если S = -12345 и SM690 = OFF, результат преобразования будет следующим.

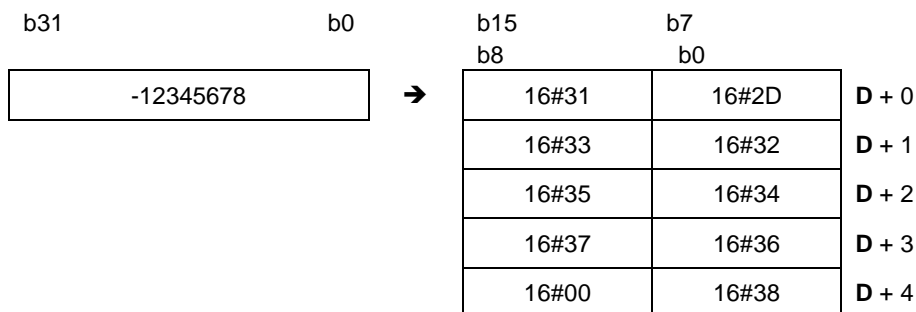


4. Значение операнда S 32-битной инструкции должно быть двоичным числом в диапазоне от -2147483648 до 2147483647. Операнд D занимает шесть регистров слова. Данные преобразуются следующим образом.



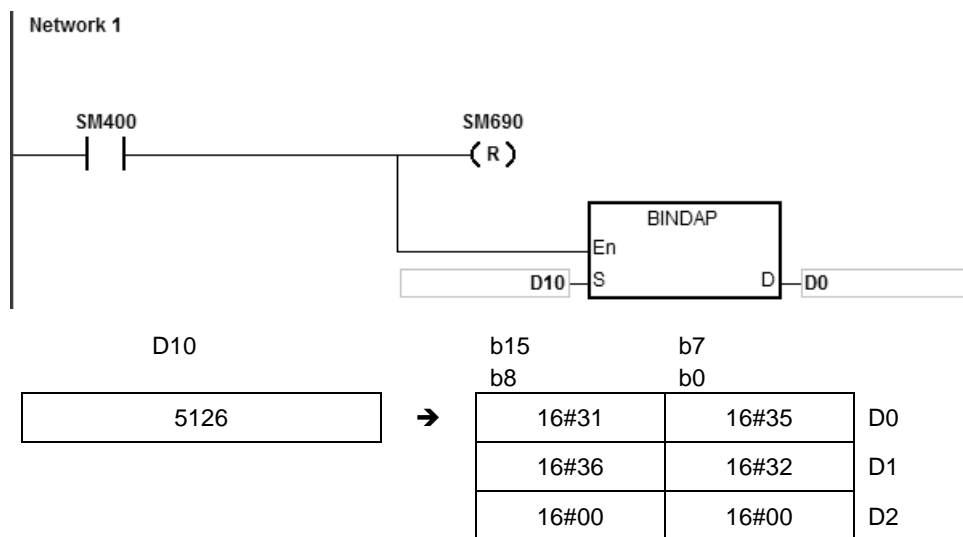
Если SM690 = OFF, символ конца 16#0000 записывается в D+4. Если SM690 = ON, значение D+4 остается без изменений, без символа конца.

Если значение операнда S положительное, в операнде D будет записываться только значение без символа знака, как на предыдущем примере. Если значение операнда S отрицательное, символ знака операнда D «-» будет записываться в 16#2D. Например, если S = -12345678 и SM690 = OFF, результат преобразования будет следующим.



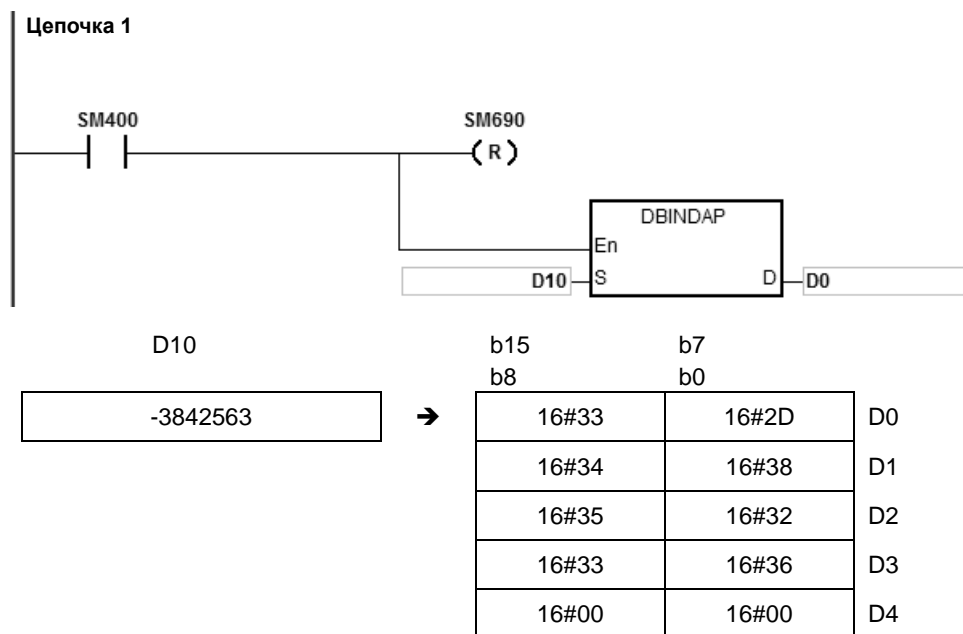
**Пример 1:**

Предположим, что L0 = 5126 и SM690 = OFF. Во время работы ПЛК D0=16#3135, D1=16#3632, D2=16#0000.



**Пример 2:**

Предположим, что D10 = -3842563 и SM690 = OFF. Во время работы ПЛК D0=16#332D, D1=16#3438, D2=16#3532, D3=16#3336, D4=16#0000.



**Дополнительные замечания:**

1. Если значение регистра **D** недостаточное для преобразования,  $SM0 = ON$  и в регистр  $SR0$  записывается код ошибки 16#2003.
2. Если операнд **D**, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа WORD/INT.
3. Если операнд **D**, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [6] типа WORD/INT.

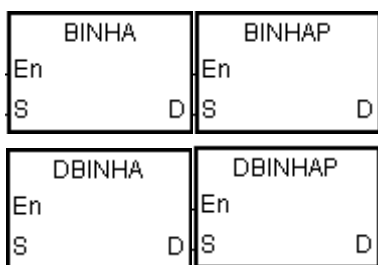
API	Код инструкции			Операнд							Функция						
2101	D	BINHA	P	S, D							Преобразование 16-разрядных двоичных значений в шестнадцатеричные значения в коде ASCII						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●	●		●	●	●						
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

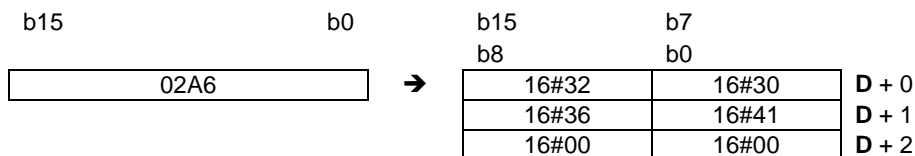
**Символьное обозначение:**



**S** : Исходное значение  
Регистр, в который  
**D** : записывается результат преобразования

**Описание:**

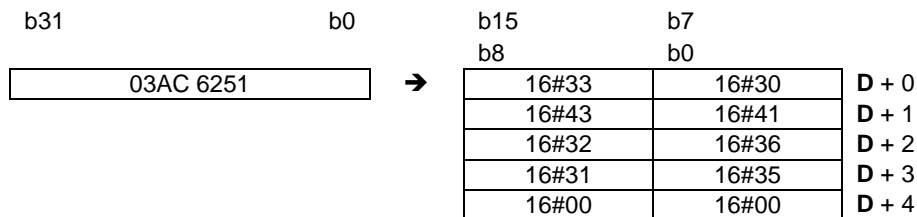
- 16-разрядное двоичное значений из операнда **S** преобразуется в код ASCII, и результат записывается в операнд **D**.
- Инструкция поддерживает SM690, который управляет конечным символом.
- Значение операнда S 16-битной инструкции должно быть двоичным числом в диапазоне от 16#0000 до 16#FFFF и должно быть 4-значным двоичным числом. Операнд D занимает три регистра слова. Данные преобразуются следующим образом.



Если SM690 = OFF, 16#0000 записывается в D+2. Если SM690 = ON, значение D+2 остается без изменений. Например, если S = 16#02A6 и SM690 = OFF, результат преобразования будет следующим.

- Значение операнда S 32-битной инструкции должно быть в диапазоне от 16#00000000 до 16#FFFFFFFF и должно быть 8-значным двоичным числом. Операнд D занимает пять регистров слова. Данные преобразуются следующим образом.



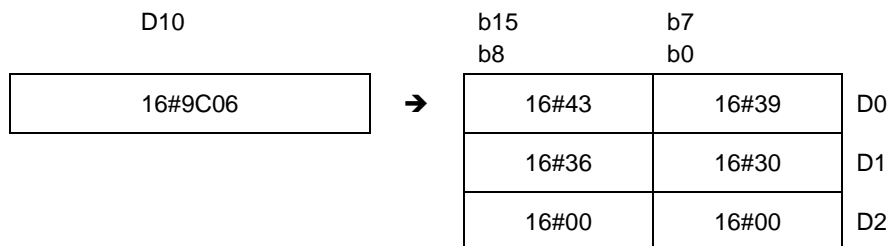
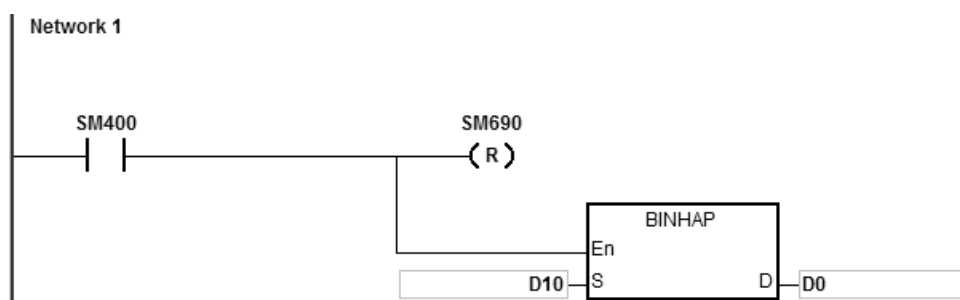


Если SM690 = OFF, 16#0000 записывается в D+4. Если SM690 = ON, значение D+4 остается без изменений.

Например, если S = 16#03AC625E и SM690 = OFF, результат преобразования будет следующим.

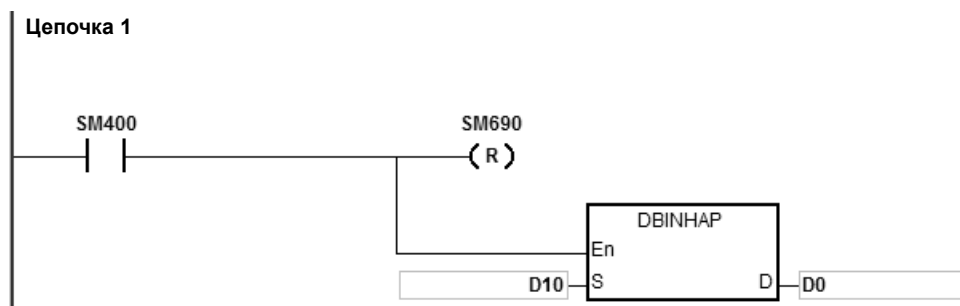
**Пример 1:**

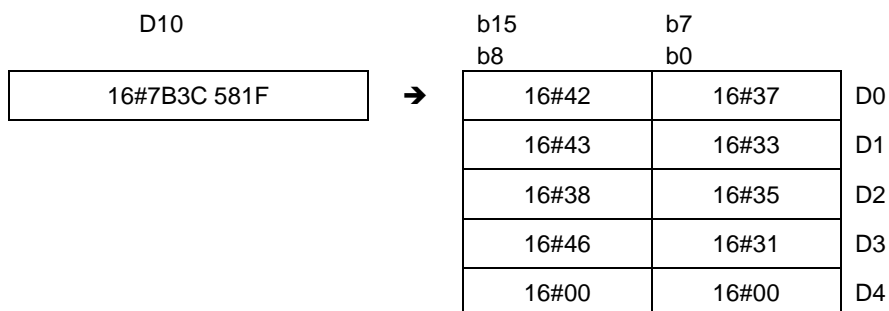
Предположим, что D10 = 16#9C06 и SM690 = OFF. Во время работы ПЛК D0=16#4339, D1=16#3630, D2=16#0000.



**Пример 2:**

Предположим, что D10 = 16#7B3C581F и SM690 = OFF. Во время работы ПЛК D0=16#4237D, D1=16#4333, D2=16#3835, D3=16#4631, D4=16#0000.



**Дополнительные замечания:**

1. Если значение D+2 16-битной инструкции вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
2. Если значение D+4-битной инструкции вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 32#2003.
3. Если операнд **D**, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [3] типа WORD/INT.
4. Если операнд **D**, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [5] типа WORD/INT.

API	Код инструкции			Операнд							Функция					
2102	D	BCDDA	P	S, D							Преобразование значений в двоично-десятичном коде в код ASCII					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●	●	●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●	●		●	●	●						
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**

BCDDA	BCDDAP
En	En
S	D

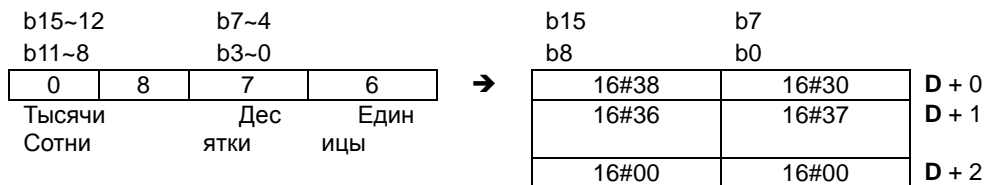
  

DVCDDA	DVCDDAP
En	En
S	D

**S** : Исходное значение  
Регистр, в который записывается результат преобразования  
**D** : Исходное значение  
Регистр, в который записывается результат преобразования

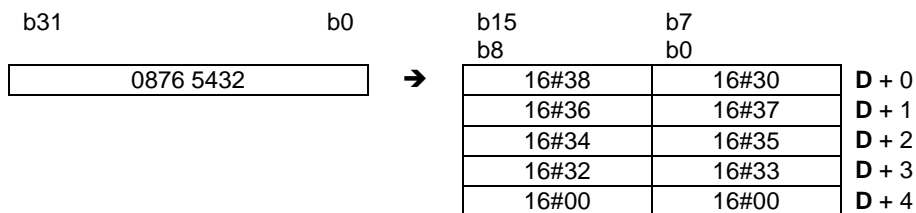
**Описание:**

- Число в двоично-десятичном коде из операнда **S** преобразуется в код ASCII, и результат записывается в операнд **D**.
- Инструкция поддерживает SM690, который управляет конечным символом.
- Число в двоично-десятичном коде операнда S 16-битной инструкции должно быть двоичным числом в диапазоне от 0 до 9999 и должно быть 4-значным числом в двоично-десятичном коде. Операнд D занимает три регистра слова. Данные преобразуются следующим образом.



Если SM690 = OFF, символ конца 16#0000 записывается в **D+2**. Если SM690 = ON, значение **D+2** остается без изменений, без символа конца.

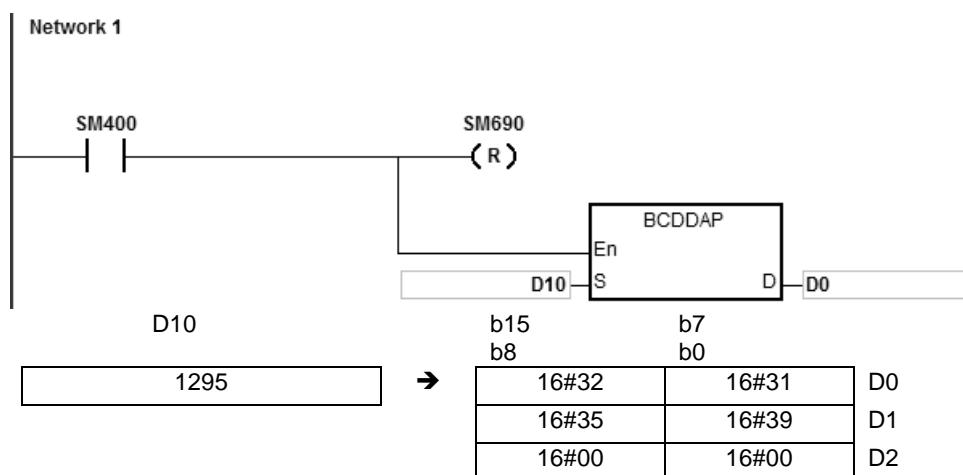
- Число в двоично-десятичном коде операнда S 32-битной инструкции должно быть двоичным числом в диапазоне от 0 до 99999999 и должно быть 8-значным числом в двоично-десятичном коде. Операнд D занимает пять регистров слова. Данные преобразуются следующим образом.



5. Если SM690 = OFF, символ конца 16#0000 записывается в D+5. Если SM690 = ON, значение D+5 остается без изменений, без символа конца.
6. Даже если первая цифра числа в двоично-десятичном коде в операнде S равна 0, оно будет преобразовываться в 0 по коду ASCII (16#30).

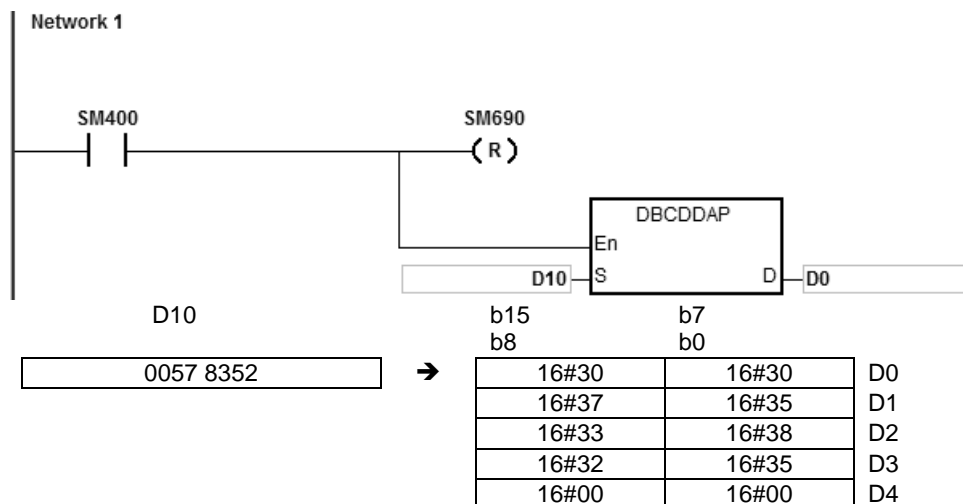
**Пример 1:**

Предположим, что значение регистра D10 в двоично-десятичном коде равно 1295 и SM690 = OFF. Во время работы ПЛК D0=16#3231, D1=16#3539, D2=16#0000.



**Пример 2:**

Предположим, что значение регистра D10 в двоично-десятичном коде равно 00578352 и SM690 = OFF. Во время работы ПЛК D0=16#3030D, D1=16#3735, D2=16#3338, D3=16#3235, D4=16#0000.



### Дополнительные замечания:

1. Если значение операнда S, используемого в 16-битной инструкции, вне диапазона от 0 до 9999, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200D. (Значение в двоично-десятичном коде представлено шестнадцатеричным значением, но один разряд вне диапазона от 0 до 9.)
2. Если значение операнда S, используемого в 32-битной инструкции, вне диапазона от 0 до 99999999, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#200D. (Значение в двоично-десятичном коде представлено шестнадцатеричным значением, но один разряд вне диапазона от 0 до 9.)
3. Если значение D+2 16-битной инструкции вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 16#2003.
4. Если значение D+4-битной инструкции вне пределов допустимого диапазона, инструкция не выполняется, SM0 = ON и у SR0 код ошибки 32#2003.
5. Если операнд D, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [3] типа WORD/INT.
6. Если операнд D, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [5] типа WORD/INT.

API	Код инструкции			Операнд								Функция				
2103	D	DABIN	P	S, D								Преобразование десятичных кодов ASCII со знаком в значения со знаком в двоично-десятичном коде				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S		●			●	●		●	●						○	
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**

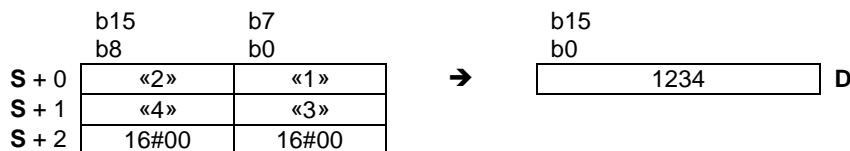


**S** : Исходное значение

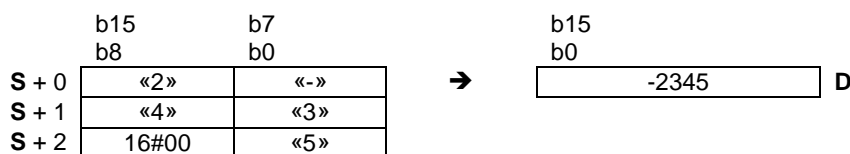
**D** : Регистр, в который записывается результат преобразования

**Описание:**

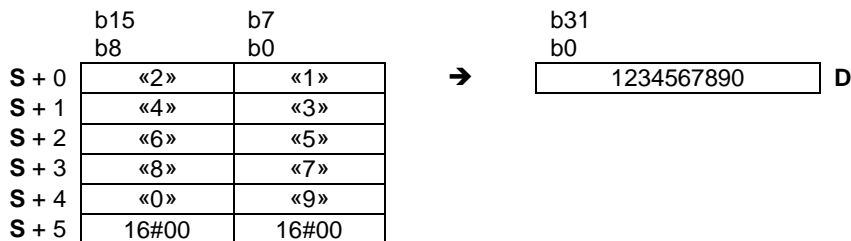
- Десятичное число в коде ASCII со знаком из операнда **S** преобразуется в число в двоично-десятичном коде со знаком, и результат записывается в операнд **D**.
- Операнд **S**, используемый в 16-битной инструкции, занимает три регистра слова, а значение в десятичном коде ASCII операнда **S** должно быть в диапазоне от -32768 до 32767. Если значение операнда **S** является строкой без символа конца 16#00, преобразование может быть до 5 разрядов (без знака).



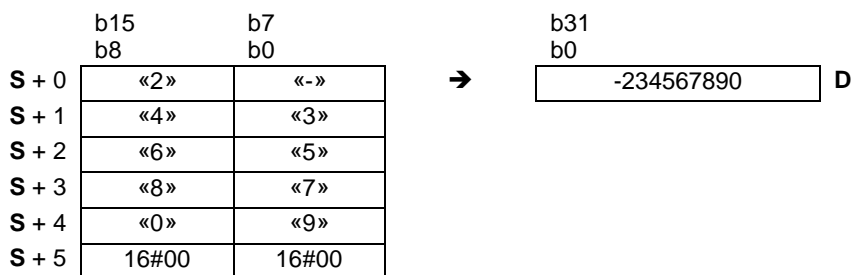
- Если первый символ « » (пробел), будет знак плюса. Если первый символ «-», будет знак минуса. Возьмем для примера строку «2345».



4. Операнд **S**, используемый в 32-битной инструкции, занимает шесть регистров слова, а значение в десятичном коде ASCII операнда **S** должно быть в диапазоне от -2147483648 до 2147483647. Если значение операнда **S** является строкой без символа конца 16#00, преобразование может быть до 10 разрядов (без знака).



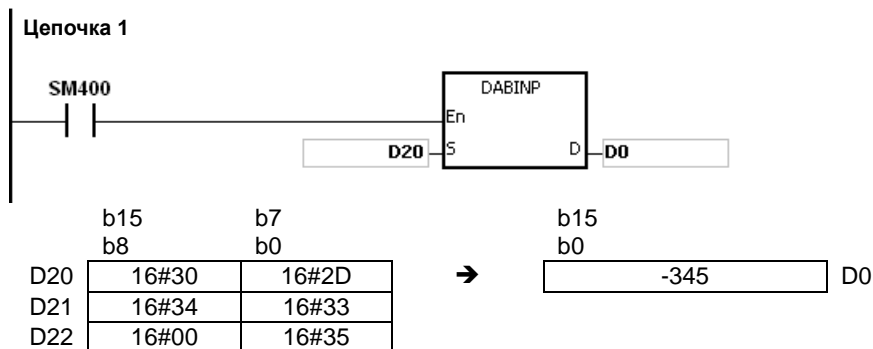
5. Если первый символ « » (пробел), будет знак плюса. Если первый символ «-», будет знак минуса. Возьмем для примера строку «-234567890».



6. Если первый разряд строки регистра операнда **S** пробел (16#20) или знак + (16#2B), это будет расцениваться как 0. Если второй разряд не является числом от 0 до 9, он будет расцениваться как конец строки и сообщения об ошибке не выдается. Например, если порядок слов 16#20→16#31→16#32→16#2B, результат преобразования будет 12.
7. Диапазон строки регистра операнда **S** для 16-битной инструкции будет 1~6 (включая знаки плюса/минуса), а для 32-битной инструкции будет 1~11 (включая знак плюса/минуса).
8. 32-битный счетчик может использоваться только в 32-битной инструкции, а в регистре **E** – не может.

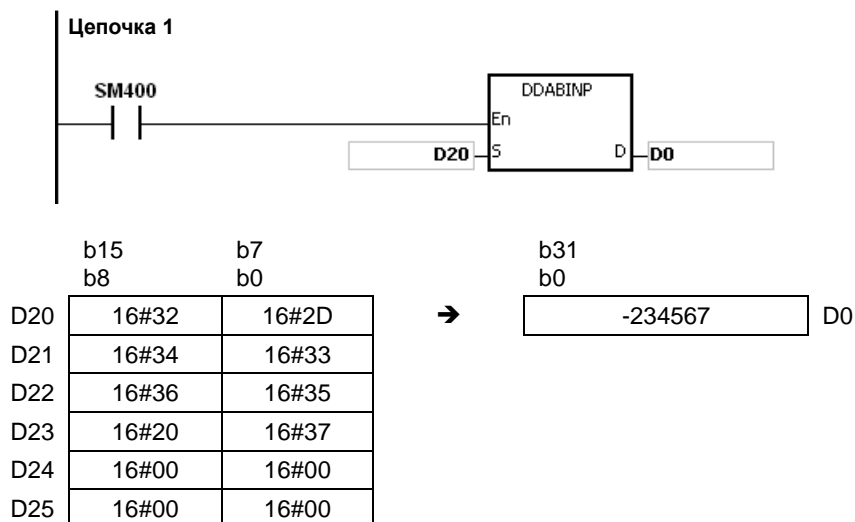
**Пример 1:**

Предположим, что D20=16#302D, D21=16#3433, D22=16#0035. Во время работы ПЛК D0=-345.



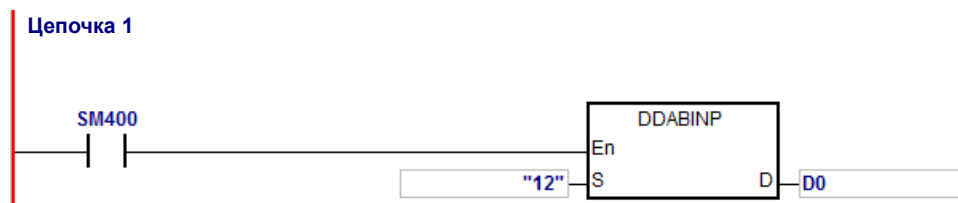
**Пример 2:**

Предположим, что D20=16#322D, D21=16#3433, D22=16#3635, D23=16#2037, D24=16#0000, D25=16#0000. Во время работы ПЛК D0=-234567.



**Пример 3:**

Предположим, что значение строки операнда S равно 12. Во время работы ПЛК D0=12.



**Дополнительные замечания:**

1. Если значение первого слова операнда **S** символ конца (16#00), инструкция будет расценивать значение операнда **S** равным 0 (16#30).
2. Если значение первого разряда операнда **S** равно 16#20 (пробел) или 16#2B (+) или 16#2D (-) и второй разряд равен 16#00, инструкция будет расценивать значение операнда **S** равным 0 (16#30).
3. Даже если первая цифра числа в двоично-десятичном коде в операнде **S** равна 0, оно будет преобразовываться в 0 по коду ASCII (16#30).
4. Значением первого разряда операнда **S** может быть только код ASCII, 16#30~16#39 (0~9), 16#200 (пробел), 16#2D (знак минуса), 16#2B (знак плюса), 16#00 (символ конца). Если значением первого разряда **S** не



является код ASCII, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

5. Если значениями всех других разрядов кроме первого в операнде **S** не являются коды ASCII, 16#30~16#39 или 16#00, это будет расцениваться как 16#00.
6. Если значение операнда **S** вне пределов диапазона регистра, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003. Инструкция не выполняется.
7. Если операнд **S**, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [3] типа WORD/INT.
8. Если операнд **S**, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [6] типа WORD/INT.

API	Код инструкции			Операнд							Функция						
2104	D	HABIN	P	S, D							Преобразование шестнадцатеричных значений в код ASCII в 16-разрядные двоичные значения						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S		●			●	●		●	●						○	
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**

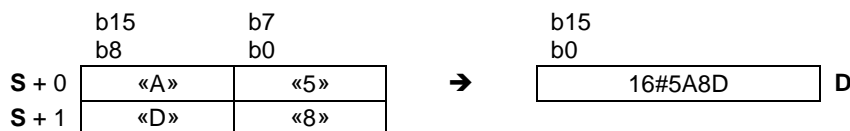


**S** : Исходное значение

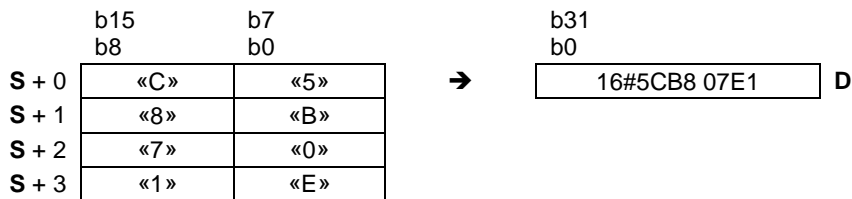
**D** : Регистр, в который записывается результат преобразования

**Описание:**

- Шестнадцатеричное значение в код ASCII из операнда **S** преобразуется в 16-разрядное двоичное значение, и результат записывается в операнд **D**.
- Операнд **S** в 16-битной инструкции занимает два регистра слова. Если значение операнда **S** является строкой без символа конца 16#00, преобразование может быть до 4 разрядов (без знака). Шестнадцатеричное значение в код ASCII операнда **S** может быть в диапазоне от 0000 до FFFF. Если **S** является строкой, она должна быть в диапазоне от «0» до «FFFF».



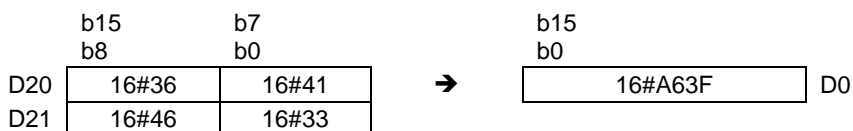
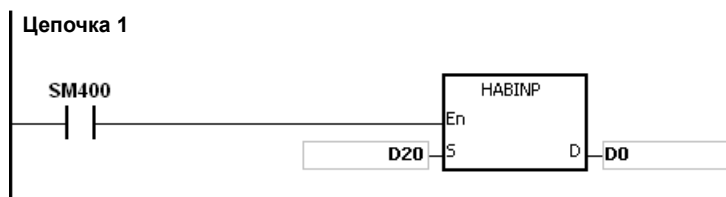
- Операнд **S** в 32-битной инструкции занимает четыре регистра слова. Если значение операнда **S** является строкой без символа конца 16#00, преобразование может быть до 8 разрядов (без знака). Шестнадцатеричное значение в код ASCII операнда **S** может быть в диапазоне от 00000000 до FFFFFFFF. Если **S** является строкой, она должна быть в диапазоне от «0» до «FFFFFFF».



4. Диапазон строки в регистре операнда S у 16-битной инструкции от 1 до 4, а у 32-битной инструкции от 1 до 8.

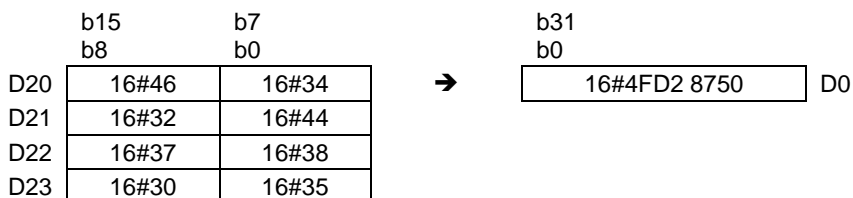
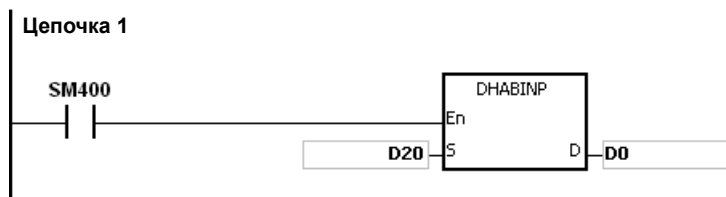
**Пример 1:**

Предположим, что D20=16#3641, D21=16#4633 (ASCII 16#A63F). Во время работы ПЛК D0=-22977.



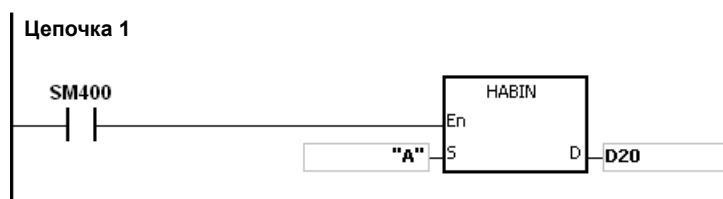
**Пример 2:**

Предположим, что D20=16#4634, D21=16#3244, D22=16#3738, D23=16#3035 (ASCII 16#4FD28750). Во время работы ПЛК значение (D1, D0) = 16#4FD28750.



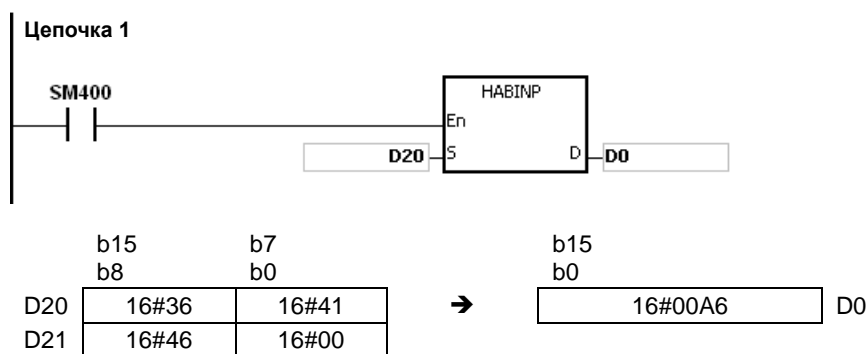
**Пример 3:**

Предположим, что значение строки в операнде **S** равно А. Во время работы ПЛК значение D20=16#A=10.



**Пример 4:**

Предположим, что D20=16#3641, D21=16#4600 (ASCII 16#00A6). Во время работы ПЛК значение D0=166.



**Дополнительные замечания:**

1. Если код ASCII в операнде **S** находится вне пределов от 16#30 до 16#39 («0»-«9») или пределов от 16#41 до 16#46 («A»-«F»), инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.
2. Если операнд S, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.
3. Если операнд S, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа WORD/INT.

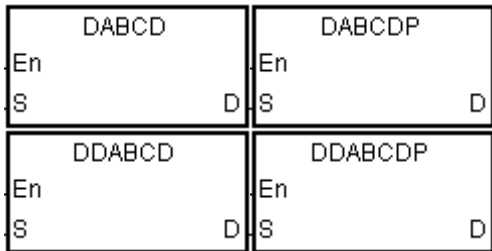
API	Код инструкции			Операнд							Функция					
2105	D	DABCD	P	S, D							Преобразование значений в коде ASCII в значения в двоично-десятичном коде					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S		●			●	●		●	●						○	
D		●			●	●	●	●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
D		●	●		●	●	●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	AS

**Символьное обозначение:**

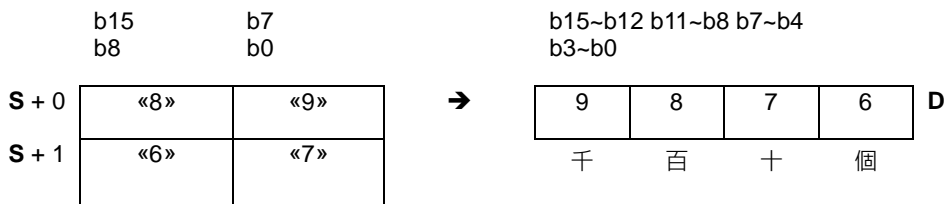


**S** : Исходное значение

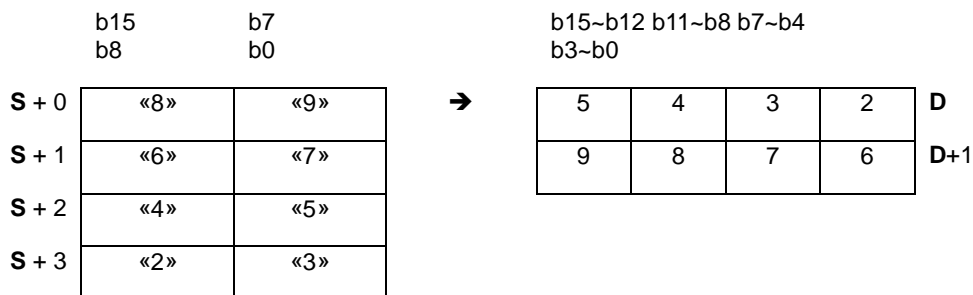
**D** : Регистр, в который записывается результат преобразования

**Описание:**

1. Число в коде ASCII из операнда **S** преобразуется в значение в двоично-десятичном коде, и результат записывается в операнд **D**.
2. Операнд **S**, используемый в 16-битной инструкции, занимает два регистра слова, а значение в коде ASCII операнда **S** должно быть в диапазоне от 0000 до 9999. Если **S** является строкой, она должна быть в диапазоне от «0» до «9999».



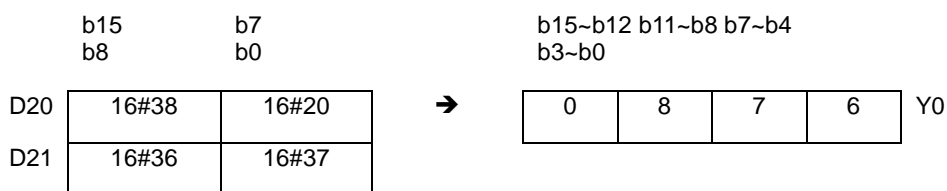
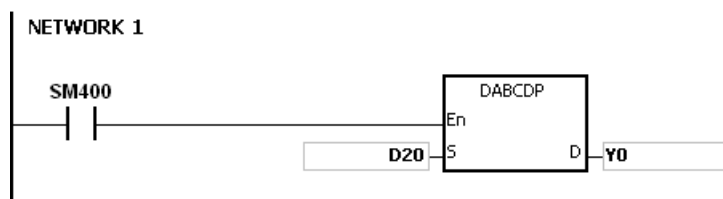
3. Операнд **S**, используемый в 32-битной инструкции, занимает четыре регистра слова, а значение в коде ASCII операнда **S** должно быть в диапазоне от 0000000 до 99999999. Если **S** является строкой, она должна быть в диапазоне от «0» до «99999999».



4. Если значение операнда **S** = 16#20, оно обрабатывается как 16#30. Если значение операнда **S** равно 16#00, оно обрабатывается как конечный символ.
5. Если операнд **S** в 16-битной инструкции является строковой переменной, количество символов в строковой переменной должно быть от 1 до 4. Если операнд **S** в 32-битной инструкции является строковой переменной, количество символов в строковой переменной должно быть от 1 до 8.

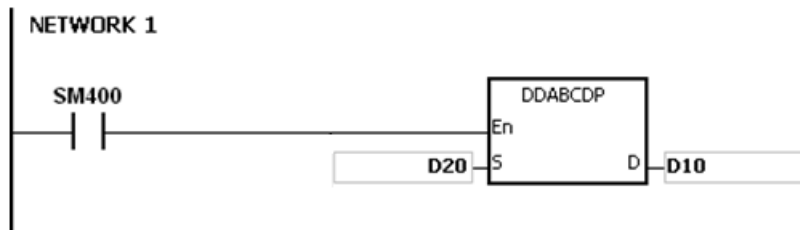
**Пример 1:**

Предположим, что D20=16#3820, D21=16#3637 (ASCII 876). Во время работы ПЛК значение будет преобразовано в Y0=16#876.



**Пример 2:**

Предположим, что D20=16#3738, D21=16#3536, D22=16#3334, D23=16#3132 (ASCII 87654321). Во время работы ПЛК значение (D11, D10) = 16#87654321.

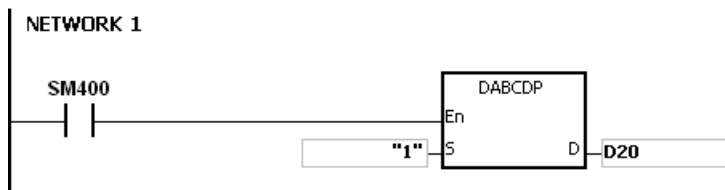


	b15 b8	b7 b0		b31 b0	
D20	16#37	16#38	→	8765 4321	D0
D21	16#35	16#36			
D22	16#33	16#34			
D23	16#31	16#32			

**Пример 3:**

Предположим, что значение строки операнда S равно 1. Во время работы ПЛК значение D20=16#0001.

6



**Пример 4:**

Предположим, что значение строковой переменной операнда S равно 1234. Во время работы ПЛК значение (D21, D20) = 16#00001234.



**Дополнительные замечания:**

1. Если в операнде **S** любой код ASCII кроме 16#30~16#39, 16#20 и 16#00, инструкция не выполняется, SM0 = ON и в регистр SR0 пишется код ошибки 16#2003.
2. Если операнд **S** является строкой и количество символов в строковой переменной превышает допустимый диапазон, инструкция не выполняется, SM0 = ON и в регистре SR0 пишется код ошибки 16#2003.
3. Если операнд S, используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.
4. Если операнд S, используемый во время выполнения 32-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [4] типа WORD/INT.



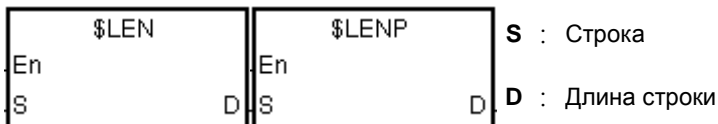
API	Код инструкции			Операнд								Функция				
2106		\$LEN	P	S, D								Вычисление длины строковой переменной				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●		●								
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
D		●			●	●							

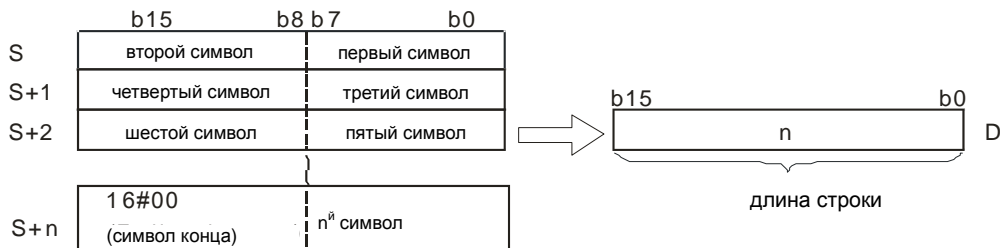
Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

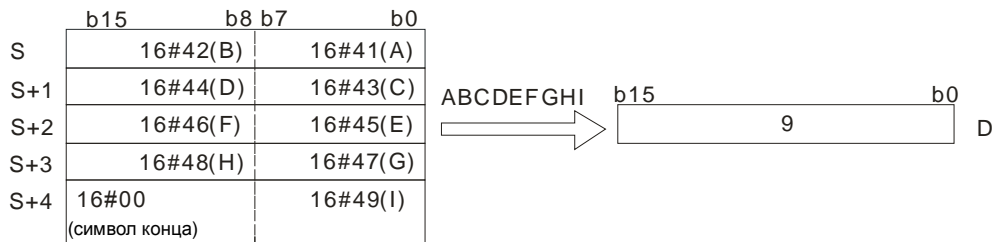


**Описание:**

1. Производится расчет длины строковой переменной, задаваемой операндом **S**, за вычетом 16#00, которым она заканчивается. Длина строковой переменной записывается в операнд **D**.
2. Значение операнда D может быть в диапазоне от 0 до 32767. Если значение операнда D вне пределов данного диапазона, оно берется равным 32767.

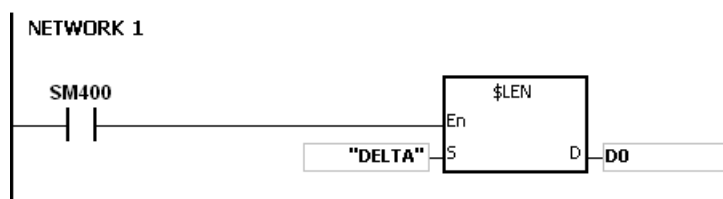


Если данные в **S~S+4** ABCDEFGHI, результат вычисления получается следующим.



**Пример 1:**

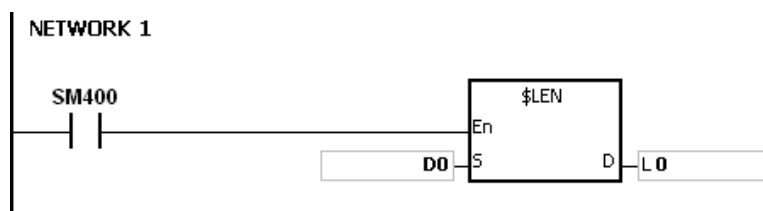
Предположим, что **S** является строкой «DELTA». Во время работы ПЛК значение D0 = 5.



**Пример 2:**

Предположим, что данные в D0~D2 выглядят следующим образом. Во время работы ПЛК значение L0 = 5.

D0	16#45 (E)	16#44 (D)
D1	16#54 (T)	16#4C (L)
D2	16#00 (символ конца)	16#41 (A)



**Дополнительные замечания:**

1. Если строковая переменная не заканчивается 16#00, инструкция выполняется с максимальным значением 32767.
2. Если длина значения выходит за пределы диапазона регистра, последним символ считается конечным символом.

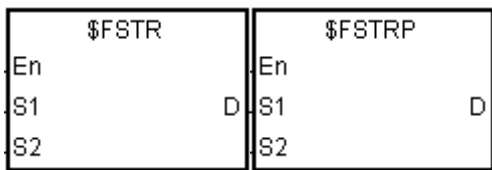
API	Код инструкции			Операнд								Функция				
2109		\$FSTR	P	<b>S<sub>1</sub>, S<sub>2</sub>, D</b>								Преобразование значения с плавающей запятой в строковую переменную				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>	●	●			●	●		●	●							○
<b>S<sub>2</sub></b>	●	●			●	●		●	●							
<b>D</b>		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
<b>S<sub>1</sub></b>									●				
<b>S<sub>2</sub></b>		●			●	●							
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

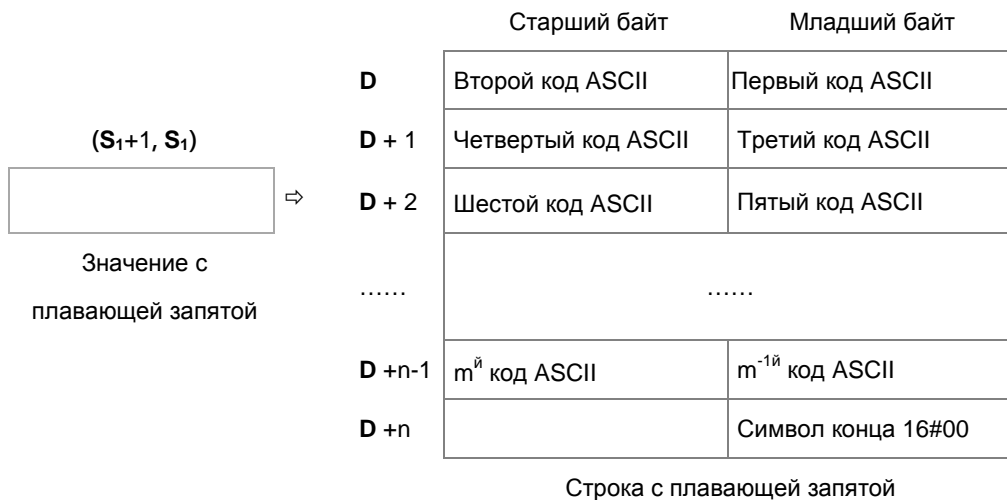
**Символьное обозначение:**



- S<sub>1</sub>** : Исходное значение
- S<sub>2</sub>** : Исходный регистр, в который записывается формат  
Исходный регистр, в который записывается результат преобразования
- D** : записывается результат преобразования

**Описание:**

1. Значение с плавающей запятой, задаваемой операндом **S<sub>1</sub>**, преобразуется в строковую переменную в соответствии со значением операнда **S<sub>2</sub>**, и результат преобразования записывается в операнде **D**.
2. Значение с плавающей запятой в операнде **S<sub>1</sub>** преобразуется в строковую переменную и на конце добавляется символ конца 16#00, а затем записывается в операнде **D**.



3. Результат преобразования зависит от значения операнда **S<sub>2</sub>**.

4. Значение  $S_{2+1}$  должно быть в диапазоне  $2 \leq S_{2+1} \leq 20$ ; знаки (+,-), разряды целой и дробной части десятичного числа могут входить, но десятичная запятая и экспонента не входят.

Операнд	Описание
$S_2$	0: Десятичный формат
	1: Экспоненциальный
$S_{2+1}$	Количество символов

5. Если значение операнда  $S_1$  положительное, первый код ASCII может записываться в D; если значение операнда  $S_1$  отрицательное, знак 16#2D (-) будет записываться первым, а за ним второй код ASCII.

6. Десятичный формат ( $S_2=0$ )

После преобразования строка с плавающей запятой в регистре D.

Знак	Целая часть	.	Дробная часть
------	-------------	---	---------------

- Значение  $S_{2+1}$  должно быть в диапазоне  $2 \leq S_{2+1} \leq 20$ ; знаки (+,-), разряды целой и дробной части десятичного числа могут входить, но десятичная запятая и экспонента не входят.
- Пример 1:

Предположим, количество символов 8. Числа с плавающей запятой -1.2345678 и 123456. Вычисление выглядит следующим образом.

D число	Значение с плавающей запятой -1.2345678		Значение с плавающей запятой 123456	
	Старший байт	Младший байт	Старший байт	Младший байт
D	16#31 (1)	16#2D (-)	16#32 (2)	16#31 (1)
D + 1	16#32 (2)	16#2E (.)	16#34 (4)	16#33 (3)
D + 2	16#34 (4)	16#33 (3)	16#36 (6)	16#35 (5)
D + 3	16#36 (6)	16#35 (5)	16#00 символ конца	
D + 4	16#00 символ конца	16#38 (8)		

Если после преобразования число с плавающей запятой может показаться в  $S_{2+1}$ , но длина превышает значение в  $S_{2+1}$ , дробная часть десятичного числа округляется. Если строка с плавающей запятой не заполняется всеми символами, это необязательно.

- Пример 2:

Если после преобразования число с плавающей запятой может быть показано в  $S_2+1$ , инструкция будет использовать для преобразования экспоненциальный формат. Например, количество символов 5 разрядов и число с плавающей запятой 1234567, поэтому результат преобразования будет 1.2346E+06.

D число	Значение с плавающей запятой 1234567	
D	16#2E (.)	16#31 (1)
D + 1	16#33 (3)	16#32 (2)
D + 2	16#36 (6)	16#34 (4)
D + 3	16#2B (+)	16#45 (E)
D + 4	16#32 (6)	16#30 (0)
D + 5	16#0000 символ конца	

- Пример 3:

Если после преобразования число с плавающей запятой может быть показано в  $S_2+1$ , инструкция будет использовать для преобразования экспоненциальный формат. Например, количество символов 2 разряда и число с плавающей запятой 0.00012345, поэтому результат преобразования будет 1.2E-04.

D число	Значение с плавающей запятой 0.00012345	
D	16#2E (.)	16#31 (1)
D + 1	16#45 (E)	16#32 (2)
D + 2	16#30 (0)	16#2D (-)
D + 3	16#00 символ конца	16#34 (4)

- Пример 4:

Если после преобразования абсолютное значение числа с плавающей запятой  $\leq 10^{-5}$ , инструкция для преобразования использует экспоненциальный формат. Например, количество символов 4 разряда и число с плавающей запятой 0.00001234, поэтому результат преобразования будет 1.234E-05.

D число	Значение с плавающей запятой 0.00001234	
D	16#2E (.)	16#31 (1)
D + 1	16#33 (3)	16#32 (2)
D + 2	16#45 (E)	16#34 (4)
D + 3	16#30 (0)	16#2D (-)
D + 4	16#00 СИМВОЛ КОНЦА	16#35 (5)

7. Экспоненциальный формат ( $S_2=1$ )

После преобразования строка с плавающей запятой в регистре D.

Знак	Целая часть	.	Дробная часть	Экспоненциальный
------	-------------	---	---------------	------------------

- Значение  $S_2+1$  должно быть в диапазоне  $2 \leq S_2+1 \leq 20$ ; знаки (+,-), разряды целой и дробной части десятичного числа могут входить, но десятичная запятая и экспонента не входят. После вычисления к длине добавляется экспоненты (4 разряда) и запятая десятичного числа.
- Количество символов в области целого числа – 1 разряд.
- Количество символов в области экспоненты – 4 слова.  
Если экспонента положительное число, 16#2B (+) добавляется в области экспоненты в операнде D.  
Если экспонента отрицательное число, 16#2D (-) добавляется в области экспоненты в операнде D.  
Количество символов в области экспоненты – 2 разряда. Если в результате преобразования получается только 1 разряд, к первой цифре в области экспоненты добавляется 16#30 (0).
- Пример:

Предположим, количество символов 8. Числа с плавающей запятой -123.456789 и 123456. Вычисление выглядит следующим образом.

D число	Значение с плавающей запятой -123.456789		Значение с плавающей запятой 12345	
	Старший байт	Младший байт	Старший байт	Младший байт
D	16#31 (1)	16#2D (-)	16#2E (.)	16#31 (1)
D + 1	16#32 (2)	16#2E (.)	16#33 (3)	16#32 (2)
D + 2	16#34 (4)	16#33 (3)	16#35 (5)	16#34 (4)

D + 3	16#36 (6)	16#35 (5)	16#2B (+)	16#45 (E)
D + 4	16#45 (E)	16#38 (8)	16#34 (4)	16#30 (0)
D + 5	16#30 (0)	16#2B (+)	16#00 СИМВОЛ КОНЦА	
D + 6	16#00 СИМВОЛ КОНЦА	16#32 (2)		

Если после преобразования число с плавающей запятой может быть показано в  $S_{2+1}$ , избыточные разряды округляются.

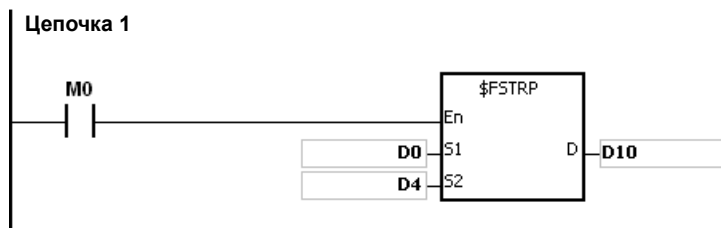
Результат преобразования длины строковой переменной для числа с плавающей запятой -123.456789 в количестве 8 символов будет равен 13 (символ конца исключается).

Результат преобразования длины строковой переменной для числа с плавающей запятой 12345 в количестве 8 символов будет равен 10 (символ конца исключается).

Если строковая переменная с плавающей запятой не заполняется всеми символами, это необязательно.

**Пример 1:**

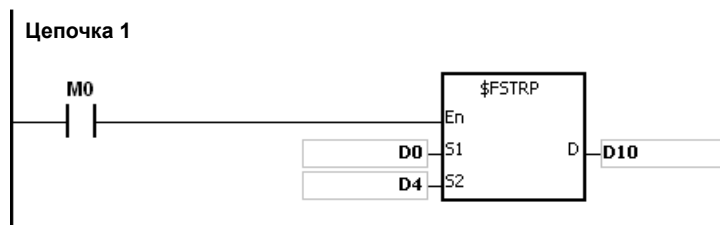
Число с плавающей запятой в (D1, D0= 12.3456) преобразовано в десятичный формат строковой переменной (D4=0, D5=8).



D10	16#32(2)	16#31(1)
D11	16#33(3)	16#2E(.)
D12	16#35(5)	16#34(4)
D13	16#00 СИМВОЛ КОНЦА	16#36(6)

**Пример 2:**

Число с плавающей запятой в (D1, D0= 0.0012345678) преобразовано в экспоненциальный формат строковой переменной (D4=1, D5=8).



D10	16#2E (.)	16#31 (1)
D11	16#33 (3)	16#30 (2)
D12	16#35 (5)	16#34 (4)
D13	16#37 (7)	16#36 (6)
D14	16#45 (E)	16#38 (8)
D15	16#30 (0)	16#2D (-)
D16	16#00 СИМВОЛ КОНЦА	16#33 (3)

**Дополнительные замечания:**

1. Если значение операнда  $S_1$  выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется,  $SM0 = ON$  и в регистр  $SR0$  записывается код ошибки 16#2013.
2. Если значение  $S_2$  не 0 и не 1, инструкция не выполняется,  $SM0 = ON$  и у  $SR0$  код ошибки 16#2003.
3. Если значение  $S_2+1$  вне диапазона  $2 \leq S_2+1 \leq 20$ , инструкция не выполняется,  $SM0 = ON$  и в регистре  $SR0$  записывается код ошибки 16#2003.
4. Если операнд  $S_2$  объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.



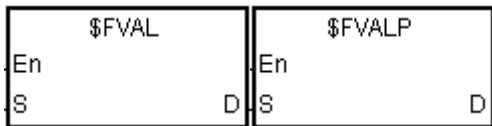
API	Код инструкции			Операнд							Функция					
2110		\$FVAL	P	S, D							Преобразование строковой переменной в значение с плавающей запятой					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●		●	●						○	
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
D									●				

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

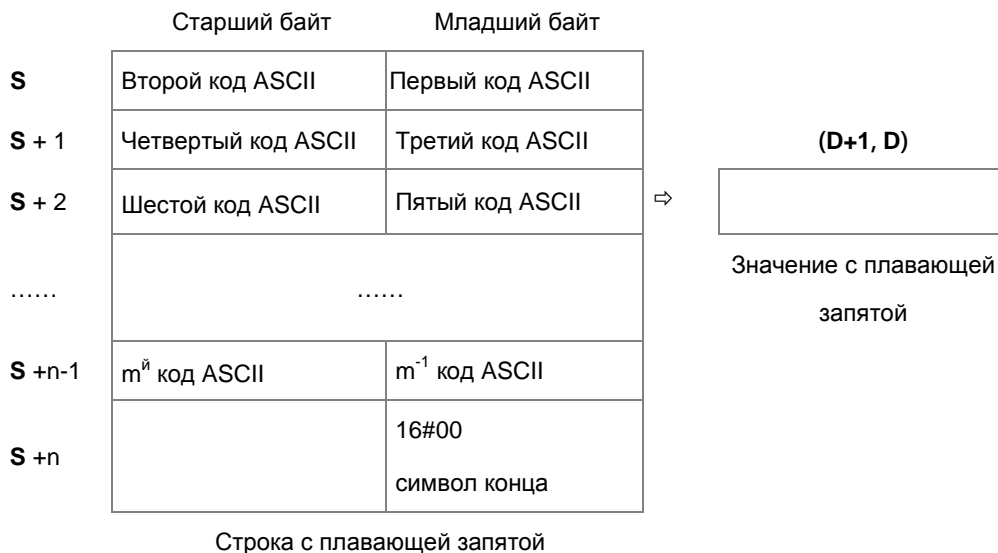
**Символьное обозначение:**



**S** : Исходное значение  
Регистр, в который  
**D** : записывается результат преобразования

**Описание:**

1. Строковая переменная в операнде **S** преобразуется в число с плавающей запятой, и результат преобразования записывается в операнд **D**.



- По кодам ASCII см. следующие разделы.
- Для десятичного и экспоненциального форматов максимальная длина строковой переменной с плавающей запятой (m) составляет 24 слова (без учета символа конца 16#00), а для n максимальная длина составляет 13 слов.

2. Значение операнда **S** может быть в десятичном или экспоненциальном формате. ПЛК автоматически определяет формат в зависимости от значения операнда **S**.

- Десятичный формат: длина строки с плавающей запятой равна 9; с учетом символа конца 16#00.

	Старший байт	Младший байт	
<b>S</b>	16#31 (1)	16#32 (2)	⇒
<b>S + 1</b>	16#32 (2)	16#2E (.)	
<b>S + 2</b>	16#34 (4)	16#33 (3)	
<b>S + 3</b>	16#36 (6)	16#35 (5)	
<b>S + 4</b>	16#00 символ конца	16#37 (7)	

<b>D</b>
21.234567

Или

	Старший байт	Младший байт	
<b>S</b>	16#31 (1)	16#2D (-)	⇒
<b>S + 1</b>	16#32 (2)	16#2E (.)	
<b>S + 2</b>	16#34 (4)	16#33 (3)	
<b>S + 3</b>	16#36 (6)	16#35 (5)	
<b>S + 4</b>	16#00 символ конца	16#38 (8)	

<b>D</b>
-1.234568

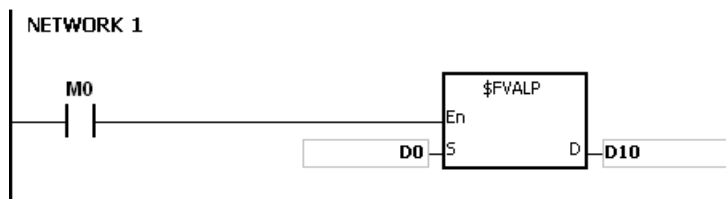
- Экспоненциальный формат: длина строки с плавающей запятой равна 9; символ конца 16#00 включен.

	Старший байт	Младший байт	
<b>S</b>	16#31 (1)	16#2D (-)	⇒
<b>S + 1</b>	16#32 (2)	16#2E (.)	
<b>S + 2</b>	16#45 (E)	16#33 (3)	
<b>S + 3</b>	16#30 (0)	16#2B (+)	
<b>S + 4</b>	16#00 символ конца	16#32 (3)	

<b>D</b>
-1230

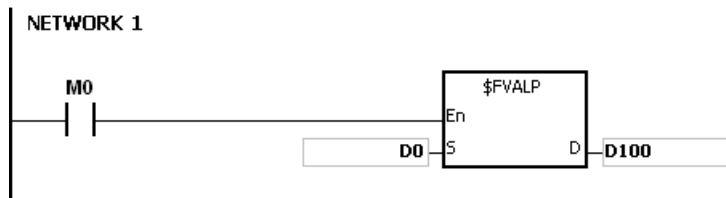
3. Если код знака в операнде **S** равен 16#20, 16#30 или 16#2B, результат сравнения получается положительным значением. Если у операнда **S**<sub>1</sub> код знака 16#2D, результат сравнения получается отрицательным.

Пример 1:



D0	16#31 (1)	16#32 (2)	
D1	16#32 (2)	16#2E (.)	(D11, D10)
D2	16#34 (4)	16#33 (3)	⇒ 21.234567
D3	16#36 (6)	16#35 (5)	
D4	16#00 символ конца	16#37 (7)	

Пример 2:



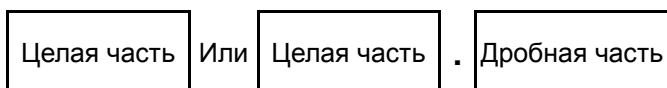
D0	16#31 (1)	16#2D (-)	
D1	16#32 (2)	16#2E (.)	(D101, D100)
D2	16#45 (E)	16#33 (3)	⇒ -1230
D3	16#30 (0)	16#2B (+)	
D4	16#00 символ конца	16#32 (3)	

Дополнительные замечания:

1. Если длина строковой переменной в операнде S более 25 байт и не заканчивается знаком 16#00, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

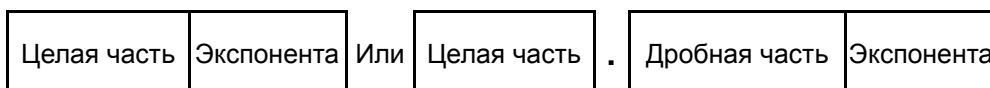
2. Есть несколько правил для значения операнда S. Если значение операнда S не соответствует этим правилам, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003. Первый код ASCII: это могут быть знаки 16#2B(+), 16#2D(-), пробел 16#20 и цифры от 16#30(0) до 16#39(9). Если первый код ASCII является знаком или пробелом, второй код ASCII должен быть цифрой. Второй код ASCII может быть в десятичном или экспоненциальном формате.

- Десятичный формат:



Запятая «.» (16#2E) может ставиться только один раз, и перед ней и после нее должны идти цифры.

- Экспоненциальный формат:



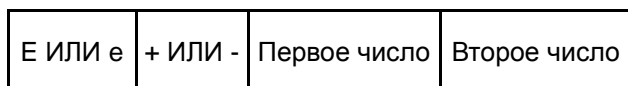
Запятая «.» (16#2E) может ставиться только один раз и перед ней и после нее должны идти цифры.

Перед экспонентой должна идти цифра.

Целая часть: это могут быть только цифры от «0» (16#30) до «9» (16#39).

Дробная часть: это могут быть только цифры от «0» (16#30) до «9» (16#39).

Экспонента: формат для 4-значного кода ASCII показан ниже.



Должно быть «E» (16#45) или «e» (16#65) и только один раз.

Должен быть знак «+» (16#2B) или «-» (16#2D) и только один раз.

Должно быть два разряда 2; цифры от «0» (16#30) до «9» (16#39).

3. Если строка S вне диапазона, инструкция не выполняется, SM = ON и у SR0 код ошибки
- Если первый символ строки цифра «0»~«9» (16#30~16#39), диапазон строки с плавающей запятой будет 1~24. Минимальная длина строки 1.
  - Если первый символ строки пробел (16#20) или знак («+» (16#2B) или «-» (16#2D)), диапазон строки с плавающей запятой будет 2~24. Минимальная длина строки «+1».
4. Если результат преобразования выходит за пределы допустимого диапазона значения с плавающей запятой, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2013.

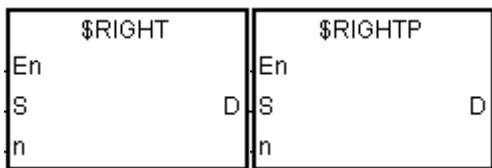
API	Код инструкции			Операнд							Функция					
2111		\$RIGHT	P	S, n, D							Извлечение символов строковой переменной справа					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●		●	●						○	
n	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

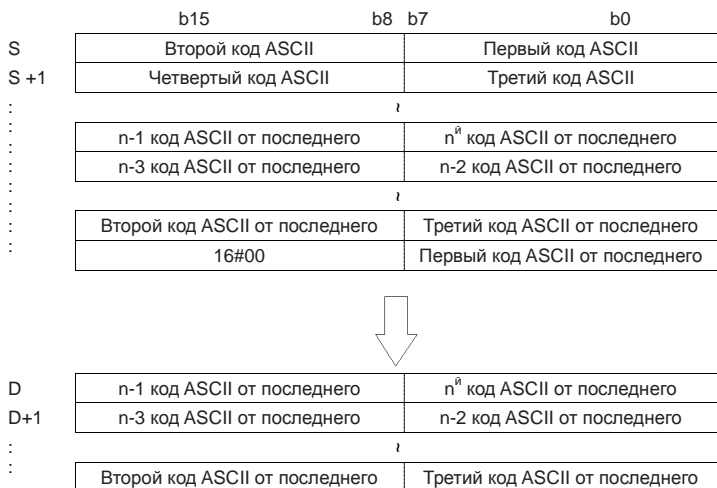
**Символьное обозначение:**



- S** : Строковая переменная
- n** : Количество извлекаемых символов
- Регистр**, в который записываются извлеченные символы
- D** : записываются извлеченные символы

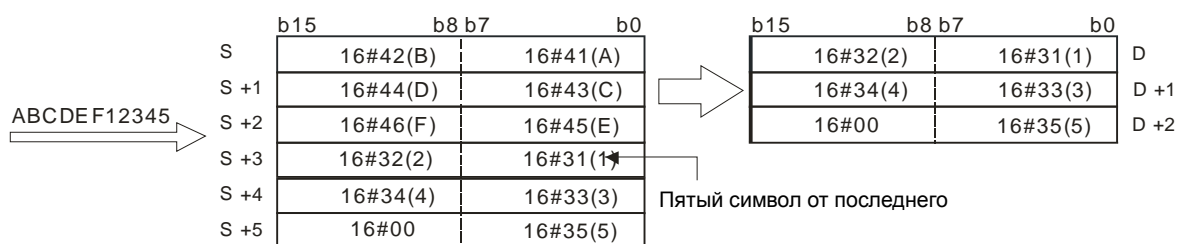
**Описание:**

- Инструкция предназначена для извлечения символов, задаваемых операндом **n**, справа в строковой переменной, задаваемой операндом **S**, и записи извлеченных символов в операнде **D**. Если операндом **S** задан регистр строковой переменной, максимальная длина значения операнда **S** равняется 31 словам; если операндом **S** задан не регистр строковой переменной, максимальная длина значения операнда **S** равняется 255 словам;
- Если **n = 0**, **D = 0**. Максимальная длина **n = 255** слов.

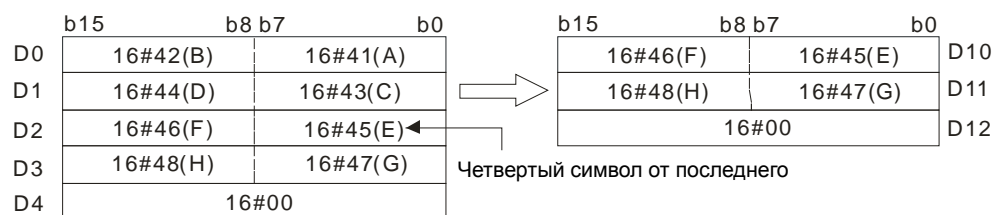
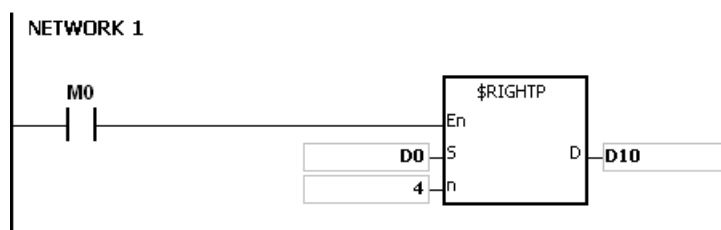


16#00	Первый код ASCII от последнего
-------	--------------------------------

Если значение операнда **S** равно ABCDEF12345, а операнд **n** = 5, из строковой переменной, заданной операндом **S**, будет извлечено пять символов справа. Результат преобразования выглядит следующим образом.



**Пример:**



**Дополнительные замечания:**

1. Если операнд **S** является не строковой переменной (\$), а регистром со строковой переменной, строковая переменная в операнде **S** может насчитывать до 256 слов (включая символ конца 16#00). Если строковая переменная, задаваемая операндом **S**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если **n** меньше 0, он берется как 0. Если **n** больше длины строки, заданной операндом **S**, он берется равным длине строки из операнда **S**.
3. Если **D** недостаточно для записи символов, заданных операндом **n**, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

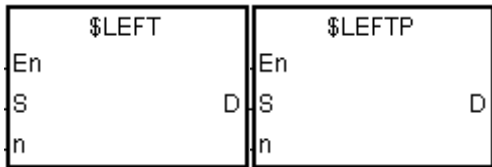
API	Код инструкции			Операнд							Функция					
2112		\$LEFT	P	S, n, D							Извлечение символов строковой переменной слева					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●			●	●		●	●						○	
n	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S		●			●	●							●
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

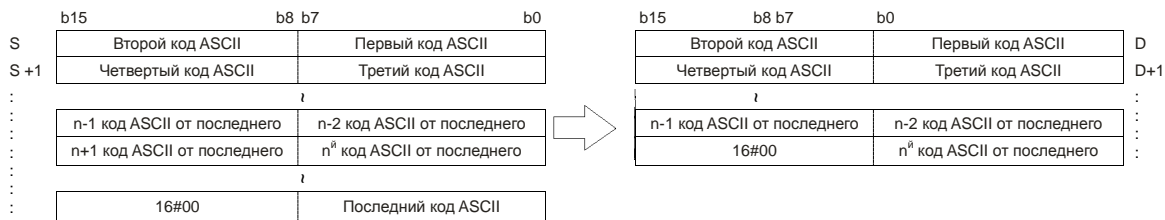
**Символьное обозначение:**



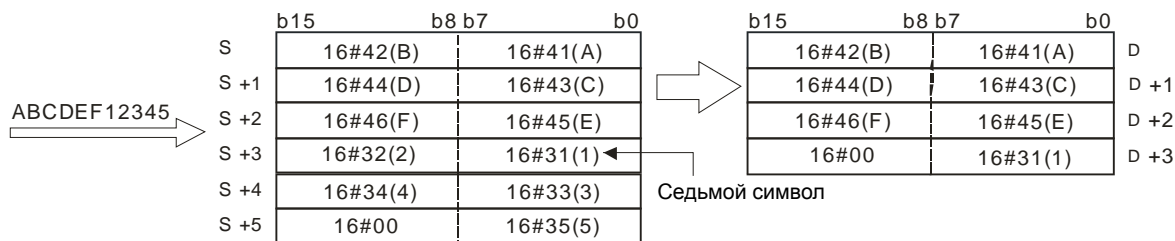
- S** : Строковая переменная
- n** : Количество извлекаемых символов
- Регистр**, в который записываются извлеченные символы
- D** : записываются извлеченные символы

**Описание:**

1. Инструкция предназначена для извлечения символов, задаваемых операндом **n**, слева в строковой переменной, задаваемой операндом **S**, и записи извлеченных символов в операнде **D**. Если операндом **S** задан регистр строковой переменной, максимальная длина значения операнда **S** равняется 31 словам; Если операндом **S** задан не регистр строковой переменной, максимальная длина значения операнда **S** равняется 255 словам;
2. Если **n = 0**, **D = 0**.

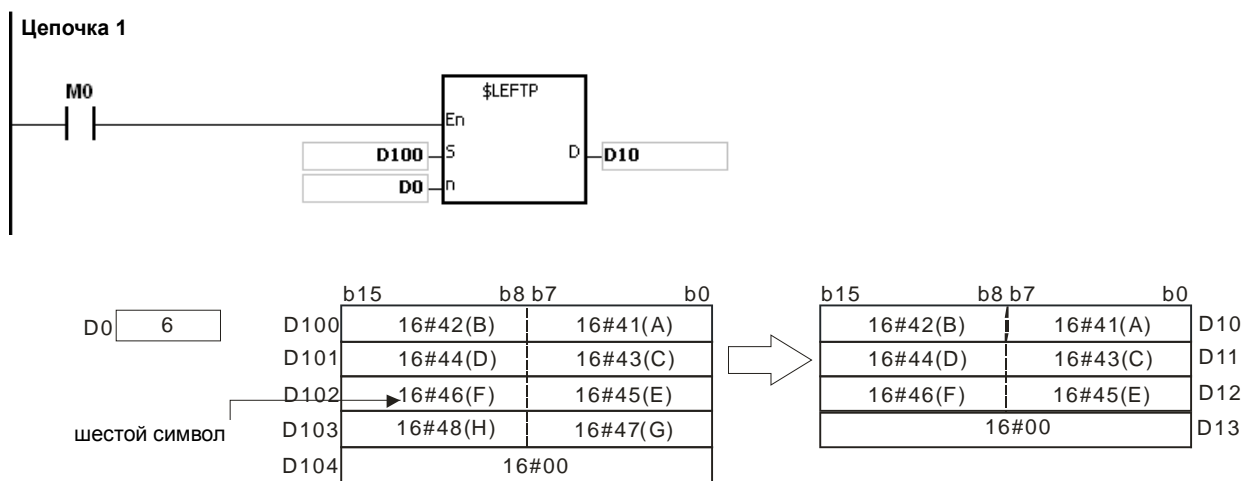


Если значение операнда **S** равно ABCDEF12345, а операнд **n = 7**, из строковой переменной, заданной операндом **S**, будет извлечено семь символов слева. Результат преобразования выглядит следующим образом.



**Пример:**

Когда M0 = ON, инструкция \$LEFT начинает исполняться. Шесть символов, начиная с символа в D100, извлекаются и записываются в D10.



**Дополнительные замечания:**

1. Если операнд S является не строковой переменной (\$), а регистром со строковой переменной, строковая переменная в операнде S может насчитывать до 256 слов (включая символ конца 16#00). Если строковая переменная, задаваемая операндом S, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если n меньше 0, он берется как 0. Если n больше длины строковой переменной, заданной операндом S, он берется равным длине строковой переменной из операнда S.
3. Если D недостаточно для записи символов, заданных операндом n, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.



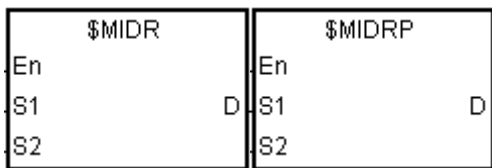
API	Код инструкции			Операнд								Функция				
2113		\$MIDR	P	<b>S<sub>1</sub>, S<sub>2</sub>, D</b>								Извлечение части строковой переменной				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>	●	●			●	●		●	●						○	
<b>S<sub>2</sub></b>	●	●			●	●		●	●		○					
<b>D</b>		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
<b>S<sub>1</sub></b>		●			●	●							●
<b>S<sub>2</sub></b>		●			●	●							
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

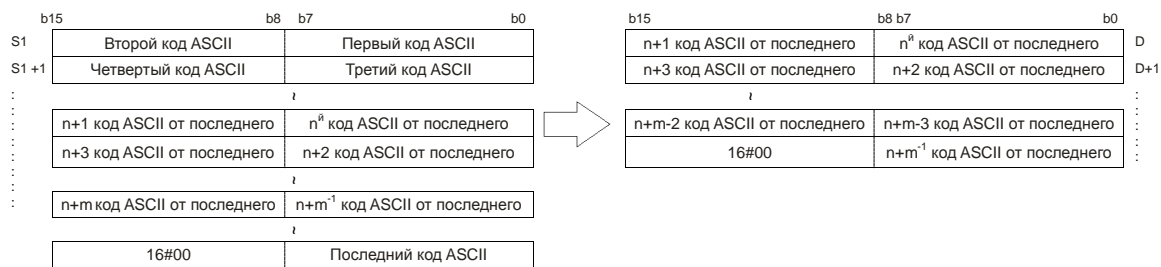
**Символьное обозначение:**



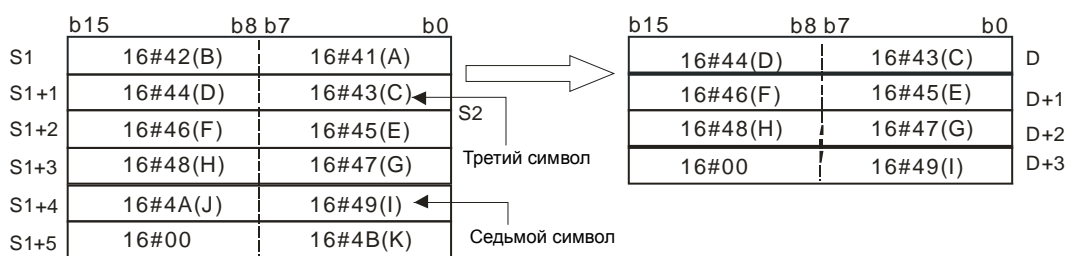
- S<sub>1</sub>** : Строковая переменная
- S<sub>2</sub>** : Часть извлекаемой строковой переменной
- D** : Регистр, в который записываются извлеченные символы

**Описание:**

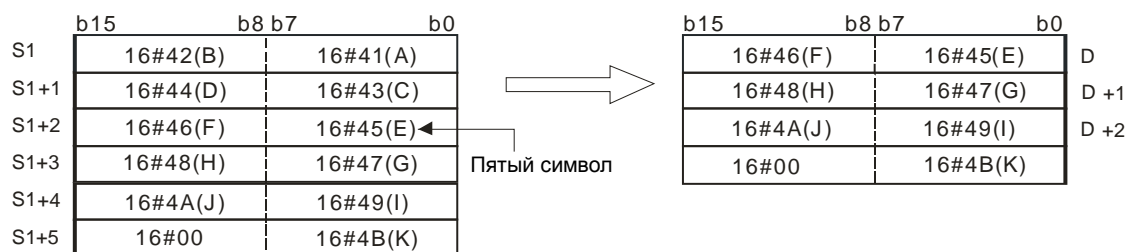
- Предположим, операнды **S<sub>2</sub>** и **S<sub>2</sub>+1** равны *n* и *m* соответственно. Символы *m*, начиная с *n*<sup>го</sup> символа строковой переменной **S<sub>1</sub>**, извлекаются и записываются в **D**.



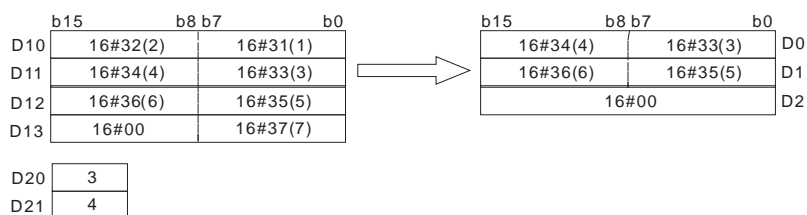
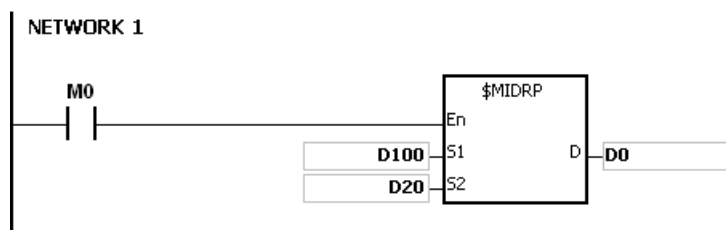
- Если данные **S<sub>1</sub>** = ABCDEFGHIJK, **S<sub>2</sub>** = 3 и **S<sub>2</sub>+1** = 7, семь символов, начиная с третьих, будут извлечены слева из строковой переменной. Результат преобразования выглядит следующим образом.



3. Если  $S_2 \leq 0$ ,  $S_2+1 < -1$  или  $S_2+1=0$ , инструкция не выполняется.
4. Если значение в  $S_2+1$  равно -1, извлекаются символы в строковой переменной  $S_1$ , начиная с символа, заданного значением в операнде  $S_2$ , до последнего символа, заданного операндом  $S_1$ .
5. Если  $S_1 = ABCDEFGHIJK$ ,  $S_2 = 5$  и  $S_2+1 = -1$ , результат преобразования будет следующим.



**Пример:**



**Дополнительные замечания:**

1. Если операнд  $S_1$  является не строковой переменной (\$), а регистром со строковой переменной, строковая переменная в операнде  $S_1$  может насчитывать до 256 слов (включая символ конца 16#00). Если строковая переменная, задаваемая операндом  $S_1$ , не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если  $S_2 \leq 0$  или  $S_2+1 < -1$ , тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

3. Если значение  $S_2$  или  $S_2 + S_{2+1}$  больше длины строковой переменной операнда  $S_1$ , тогда  $SM0 = ON$  и в регистре  $SR0$  записывается код ошибки 16#2003.
4. Если значение  $S_{2+1}$  больше количества символов, которые можно извлечь из строковой переменной в операнде  $S_1$ , тогда  $SM0 = ON$  и в регистре  $SR0$  записывается код ошибки 16#2003.
5. Если  $D$  недостаточно для записи символов, заданных операндом  $S_{2+1}$ , инструкция не выполняется,  $SM0 = ON$  и в регистре  $SR0$  записывается код ошибки 16#2003.
6. Если операнд  $S_2$ , используемый во время выполнения 16-битной инструкции, объявлен в редакторе ISPSOft, типом данных будет МАССИВ [2] типа WORD/INT.

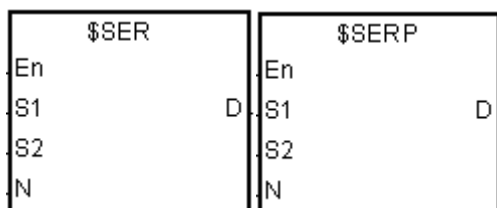
API	Код инструкции			Операнд						Функция					
2115		\$SER	P	<b>S<sub>1</sub>, S<sub>2</sub>, n, D</b>						Поиск строковой переменной					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>	●	●			●	●		●	●						○	
S <sub>2</sub>	●	●			●	●		●	●						○	
n	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●			○	○				

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S <sub>1</sub>		●			●	●							●
S <sub>2</sub>		●			●	●							●
n		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

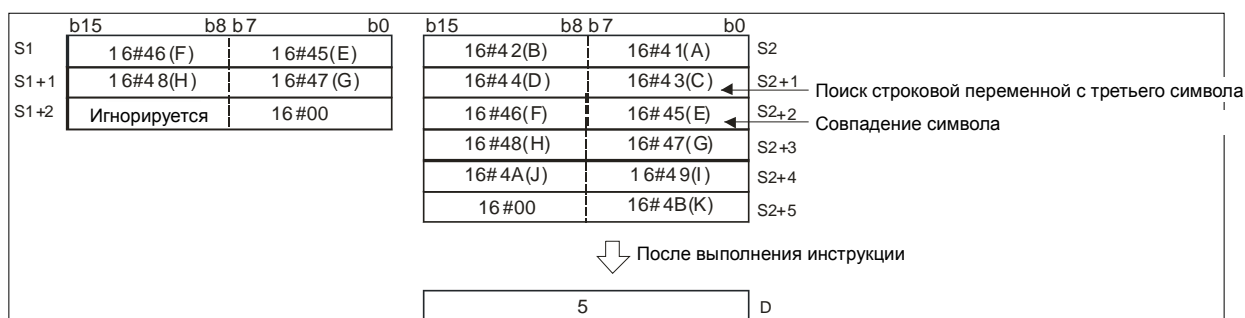
**Символьное обозначение:**



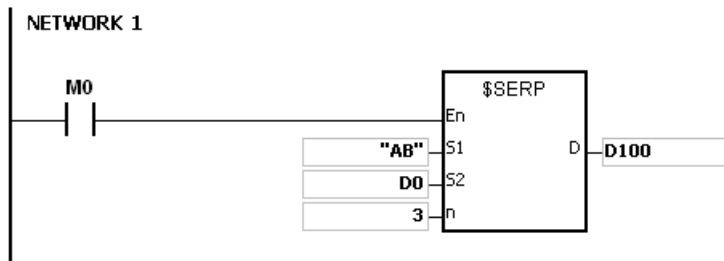
- S<sub>1</sub> : Строковая переменная, которая ищется
- S<sub>2</sub> : Строковая переменная, в которой ведется поиск
- n : n<sup>й</sup> символ в строке операнда S<sub>2</sub>, с которого начинается поиск
- D : Результат поиска

**Описание:**

- Инструкция предназначена для поиска строковой переменной, начиная с n<sup>го</sup> символа строковой переменной, заданной операндом S<sub>2</sub>, в строковой переменной, заданной операндом S<sub>1</sub>, и результат поиска записывается в операнде D.
- Если n<sup>й</sup> символ больше строковой переменной в S<sub>2</sub>, или S<sub>1</sub> > S<sub>2</sub>, тогда D = 0.
- Предположим, что S<sub>2</sub> = «ABCDEFGHIIJK», S<sub>1</sub> = «EFGH» и n = 3. Поиск начинается в строковой переменной операнда S<sub>2</sub>, с третьего символа и D = 5.



Пример:



Дополнительные замечания:

1. Если операнд **S<sub>1</sub>** является не строковой переменной (\$), а регистром со строковой переменной, строковая переменная в операнде **S<sub>1</sub>** может насчитывать до 256 слов (включая символ конца 16#00). Если строковая переменная, задаваемая операндом **S<sub>1</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если операнд **S<sub>1</sub>** является не строковой переменной (\$), а регистром со строковой переменной, строка в операнде **S<sub>2</sub>** может насчитывать до 256 слов (включая символ конца 16#00). Если строковая переменная, задаваемая операндом **S<sub>1</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
3. Если **n** меньше или равен 0, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнд								Функция				
2116		\$RPLC	P	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, D</b>								Замена символов в строковой переменной				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>	●	●			●	●		●	●							
S <sub>2</sub>	●	●			●	●		●	●						○	
S <sub>3</sub>	●	●			●	●		●	●		○	○	○	○		
S <sub>4</sub>	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
S <sub>4</sub>		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

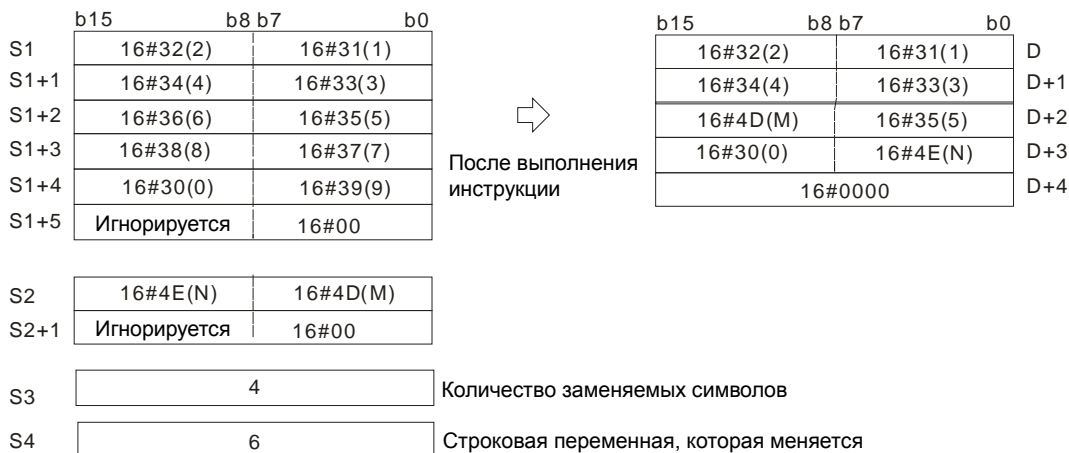
**Символьное обозначение:**

\$RPLC		\$RPLCP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	
S4		S4	

- S<sub>1</sub>** : Строковая переменная, в которой производится замена
- S<sub>2</sub>** : Новая строковая переменная
- S<sub>3</sub>** : Количество символов, заменяемых в строковой переменной **S<sub>1</sub>**
- S<sub>4</sub>** : Символы в строковой переменной **S<sub>1</sub>** будут заменены, начиная с символа, заданного операндом **S<sub>4</sub>**.
- D** : Регистр, в который записывается результат

**Описание:**

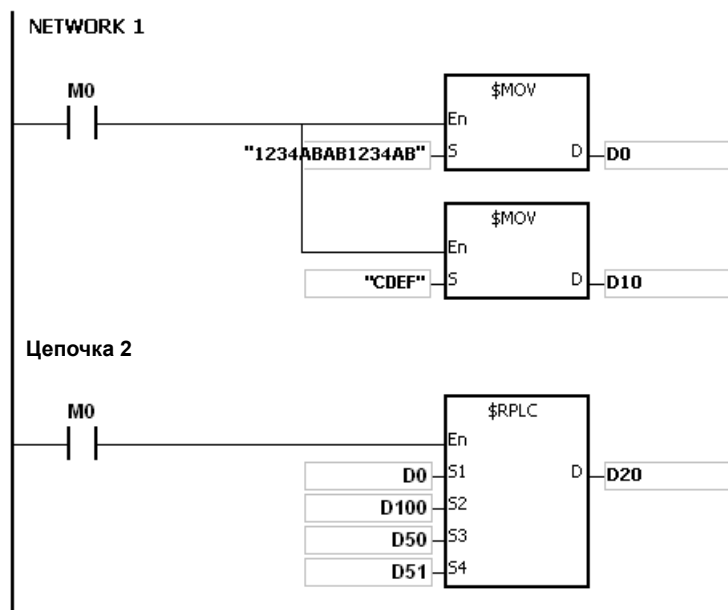
- Символы в строковой переменной операнда **S<sub>1</sub>**, начиная с символа заданного операндом **S<sub>4</sub>**, заменяются символами, заданными операндом **S<sub>2</sub>**, а количество заменяемых символов задано операндом **S<sub>3</sub>**. Результат операции записывается в операнде **D**.
- Четыре символа начиная с шестого в строке «12345**67890**» заменяются на «MN», и в результате получается «12345**MN0**».



3. Если строка в операнде  $S_2 = 16\#00$ , инструкция удаляет символы.
4. Если значение операнда  $S_3$  больше количества символов, которые можно заменить в строковой переменной операнда  $S_1$ , символы в строковой переменной операнда  $S_1$ , начиная с символа, заданного операндом  $S_4$  и до последнего символа строки в операнде  $S_1$  заменяются.
5. Если значение операнда  $S_3$  равно 0, инструкция не выполняется.

**Пример:**

Если  $M0 = ON$ ,  $D0\sim D7 = \text{«1234ABAB1234AB»}$  и  $D10\sim D11 = \text{«CDEF»}$ . Когда выполняется инструкция  $\$RPLC$ , символы в  $D0\sim D7$ , начиная с символа, заданного регистром  $D51$ , заменяются на символы из регистров  $D10\sim D11$ . Количество заменяемых символов определяется значением регистра  $D50$ , а результат записывается в  $D20\sim D27$ .



Если регистры D50 и D51 имеют значения 3 и 4 соответственно, результат операции будет следующим.

	b15	b8 b7	b0		b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		После выполнения инструкции	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)			16#43(C)	16#33(3)		D21
D2	16#42(B)	16#41(A)			16#45(E)	16#44(D)		D22
D3	16#42(B)	16#41(A)			16#41(A)	16#46(F)		D23
D4	16#32(2)	16#31(1)			16#31(1)	16#42(B)		D24
D5	16#34(4)	16#33(3)			16#33(3)	16#32(2)		D25
D6	16#42(B)	16#41(A)			16#41(A)	16#34(4)		D26
D7	Игнорируется	16#00			16#00	16#42(B)		D27
D10	16#44(D)	16#43(C)						
D11	16#45(F)	16#45(E)						
D12	Игнорируется	16#00						
D50	3		Количество заменяемых символов					
D51	4		Строковая переменная, которая меняется					

Если регистры D50 и D51 имеют значения 4 и 4 соответственно, результат операции будет следующим.

	b15	b8 b7	b0		b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		После выполнения инструкции	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)			16#43(C)	16#33(3)		D21
D2	16#42(B)	16#41(A)			16#45(E)	16#44(D)		D22
D3	16#42(B)	16#41(A)			16#42(B)	16#46(F)		D23
D4	16#32(2)	16#31(1)			16#32(2)	16#31(1)		D24
D5	16#34(4)	16#33(3)			16#34(4)	16#33(3)		D25
D6	16#42(B)	16#41(A)			16#42(B)	16#41(A)		D26
D7	Игнорируется	16#00			16#0000			D27
D10	16#44(D)	16#43(C)						
D11	16#45(F)	16#45(E)						
D12	Игнорируется	16#00						
D50	4		Количество заменяемых символов					
D51	4		Строковая переменная, которая меняется					



Если регистры D50 и D51 имеют значения 20 и 4 соответственно, результат операции будет следующим.



Если регистры D50, D51 и D10 имеют значения 3, 4 и 16#00 соответственно, результат операции будет следующим.

Три символа из регистров D0~D7, начиная с четвертого удаляются.



**Дополнительные замечания:**

1. Если строковая переменная, задаваемая операндом **S<sub>1</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если строковая переменная, задаваемая операндом **S<sub>2</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
3. Если **S<sub>3</sub> < 0** или **S<sub>4</sub> ≤ 0** или **S<sub>4</sub>** больше длины строковой переменной, заданной операндом **S<sub>1</sub>**, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.
4. Если замененное значение в строковой переменной (включая символ конца 16#00) в операнде **S<sub>1</sub>** больше 256, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнд							Функция					
2117		\$DEL	P	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D</b>							Удаление символов в строковой переменной					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>	●	●			●	●		●	●							
<b>S<sub>2</sub></b>	●	●			●	●		●	●		○	○	○	○		
<b>S<sub>3</sub></b>	●	●			●	●		●	●		○	○	○	○		
<b>D</b>		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
<b>S<sub>1</sub></b>		●			●	●							
<b>S<sub>2</sub></b>		●			●	●							
<b>S<sub>3</sub></b>		●			●	●							
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

\$DEL		\$DELP	
En		En	
S1	D	S1	D
S2		S2	
S3		S3	

- S<sub>1</sub>** : Строковая переменная
- S<sub>2</sub>** : Количество удаляемых символов
- S<sub>3</sub>** : Символы в строковой переменной **S<sub>1</sub>** будут удалены, начиная с символа, заданного операндом **S<sub>3</sub>**.
- D** : Регистр, в который записывается результат

**Описание:**

- Символы в строковой переменной операнда **S<sub>1</sub>**, начиная с символа заданного операндом **S<sub>3</sub>**, удаляются в количестве, заданном операндом **S<sub>2</sub>**, и результат операции записывается в операнде **D**.
- Четыре символа, начиная с третьего символа в строке «1234567890» операнда **S<sub>1</sub>** удаляются и результат «127890» записывается в операнде **D**.



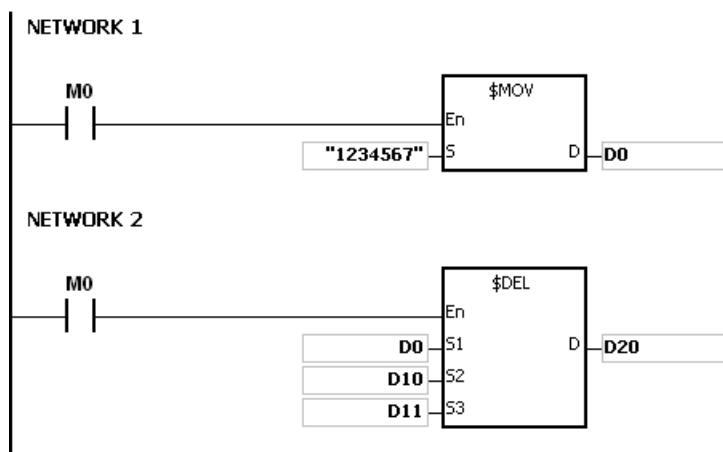
3. Если значение операнда  $S_2$  больше количества символов, которые можно удалить в строке, заданной операндом  $S_1$ , удаляются символы строки операнда  $S_1$ , начиная с символа, заданного операндом  $S_3$  и до последнего символа строки  $S_1$ , а в операнде  $D$  записывается 16#00.



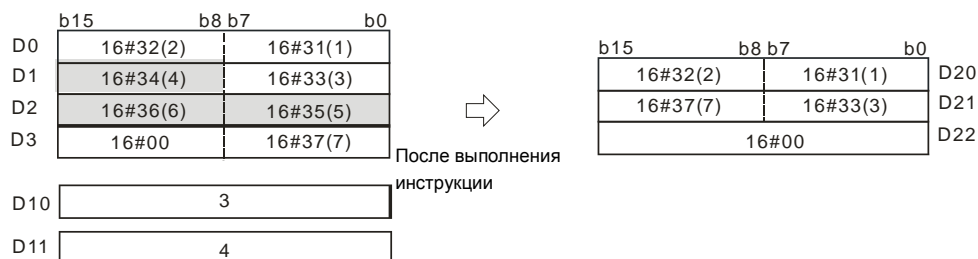
4. Если значение операнда  $S_2$  равно 0, инструкция не выполняется.

**Пример:**

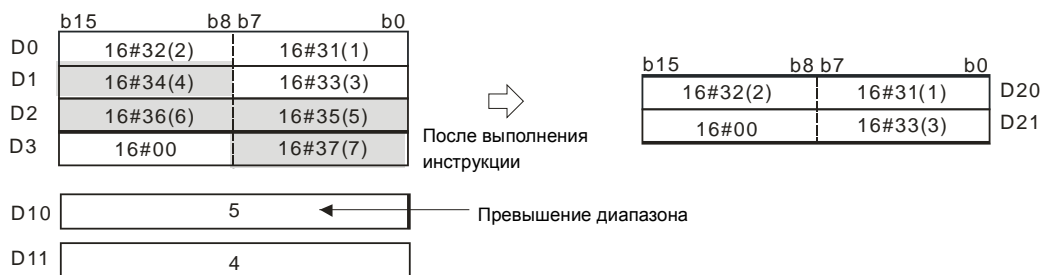
Если  $M0 = ON$ ,  $D0\sim D3 = \text{«1234567»}$ . При выполнении инструкции \$DEL символы в регистрах  $D0\sim D3$ , начиная с символа, заданного регистром  $D11$ , удаляются. Количество удаляемых символов определяется значением регистра  $D10$ , а результат записывается в операнде  $D$ .



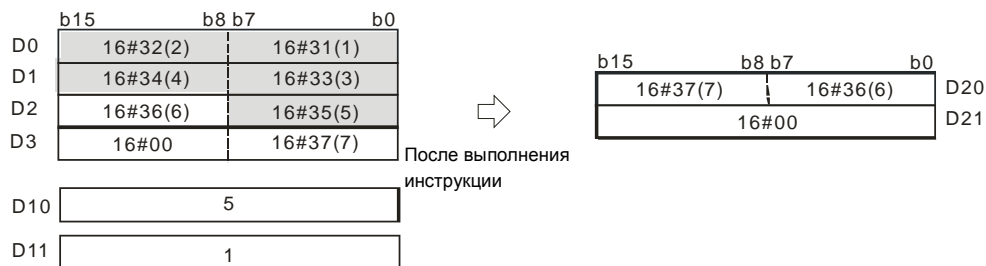
Если регистры  $D10$  и  $D11$  имеют значения 3 и 4 соответственно, результат операции будет следующим.



Если регистры  $D10$  и  $D11$  имеют значения 5 и 4 соответственно, результат операции будет следующим. Учитывая, что количество удаляемых символов превышает допустимый диапазон, символы в регистрах  $D0\sim D3$ , начиная с четвертого символа и до последнего символа, удаляются.



Если регистры D10 и D11 имеют значения 5 и 1 соответственно, результат операции будет следующим.



**Дополнительные замечания:**

1. Если строковая переменная, задаваемая операндом **S<sub>1</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если значение операнда **S<sub>2</sub>** меньше 0, значение операнда **S<sub>3</sub>** меньше или равно 0 или значение операнда **S<sub>3</sub>** больше длины строки, заданной операндом **S<sub>1</sub>**, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнд							Функция						
2118		\$CLR	P	<b>S</b>							Удаление строковой переменной						
Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F	
<b>S</b>		●			●	●		●									
Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String				
<b>S</b>		●			●	●											
Импульсная инструкция								16-битная инструкция				32-битная инструкция					
AS								AS				-					

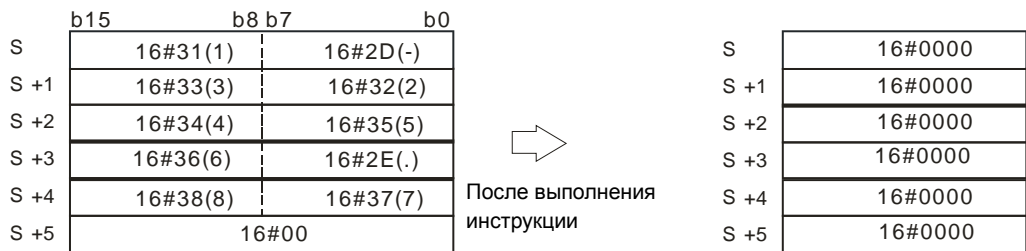
**Символьное обозначение:**

\$CLR	\$CLR P
En	En
S	S

**S** : Строковая переменная, которая удаляется

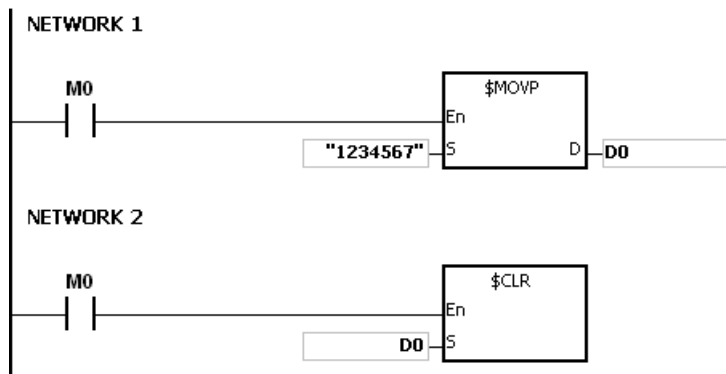
**Описание:**

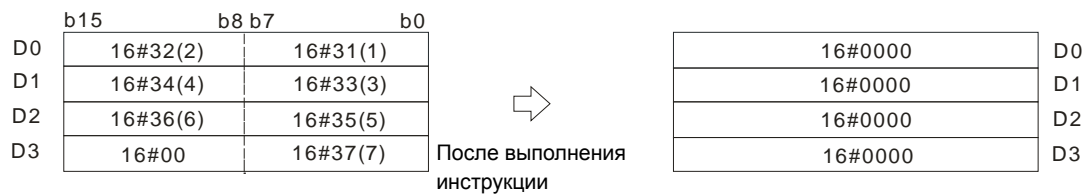
1. Строковая переменная, заданная операндом **S**, удаляется. Если строковая переменная, заданная операндом **S**, не заканчивается символом 16#00, будет удалено до 255 символов.



**Пример:**

Строковая переменная, заданная регистром D0, удаляется, как показано ниже.





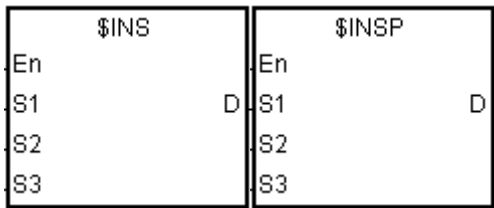
API	Код инструкции			Операнд								Функция				
2119		\$INS	P	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D</b>								Вставка строковой переменной				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>	●	●			●	●		●	●							
S <sub>2</sub>	●	●			●	●		●	●						○	
S <sub>3</sub>	●	●			●	●		●	●		○	○	○	○		
D		●			●	●		●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
D		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S<sub>1</sub> : Строковая переменная
- S<sub>2</sub> : Строковая переменная, которая вставляется
- S<sub>3</sub> : Строковая переменная вставляется в строку, заданную операндом S<sub>1</sub> после символа, заданного операндом S<sub>3</sub>.
- D : Регистр, в который записывается результат

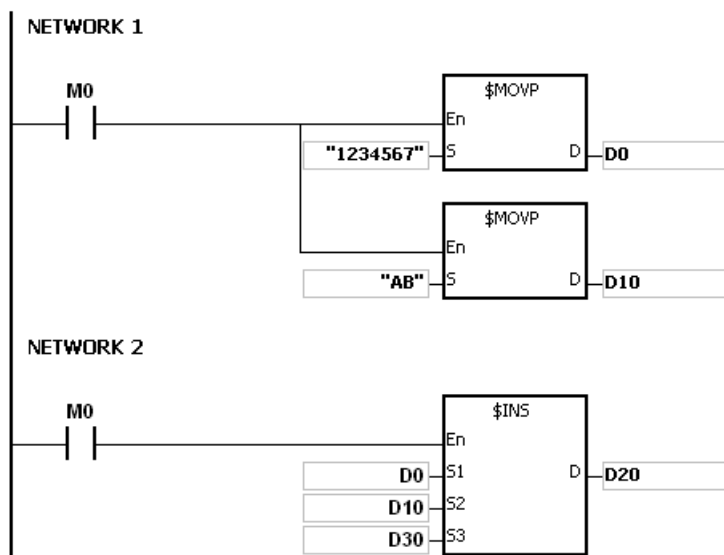
**Описание:**

1. Строковая переменная, задана операндом S<sub>2</sub>, вставляется в строковую переменную, заданную операндом S<sub>1</sub> после символа, заданного операндом S<sub>3</sub> и результат записывается в операнде D.
2. Если строковая переменная, заданная операндом S<sub>1</sub> или S<sub>2</sub> пустая, в регистре D записывается другая строковая переменная, которая не является пустой.
3. Если строковые переменные, заданные операндами S<sub>1</sub> и S<sub>2</sub>, пустые, в регистре D записывается 16#0000.



**Пример:**

Если M0 = ON, D0~D3 = «1234567» и D10 = «AB». При выполнении инструкции \$INS «AB» вставляется в строковую переменную регистров D0~D3 после символа, заданного регистром D30. Результат записывается в регистры D20~D24.



Если D30 = 1, результат операции будет следующим.

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)		D21
D2	16#36(6)	16#35(5)		D22
D3	16#00	16#37(7)		D23
				D24
D10	16#42(B)	16#41(A)		
D11	Игнорируется	16#00		
D30	1			

После выполнения инструкции

Если D30 = 0, результат операции будет следующим.

	b15	b8 b7	b0	
D0	16#32(2)	16#31(1)		D20
D1	16#34(4)	16#33(3)		D21
D2	16#36(6)	16#35(5)		D22
D3	16#00	16#37(7)		D23
				D24
D10	16#42(B)	16#41(A)		
D11	Игнорируется	16#00		
D30	0			

После выполнения



**Дополнительные замечания:**

1. Если строковая переменная, задаваемая операндом **S<sub>1</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
2. Если строковая переменная, задаваемая операндом **S<sub>2</sub>**, не заканчивается 16#00, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#200E.
3. Если **S<sub>3</sub>** меньше нуля или если **S<sub>3</sub>** больше длины строковой переменной, заданной операндом S1, тогда SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.
4. Если вставленное значение в строковой переменной (включая символ конца 16#00) в операнде **S1** больше 256, инструкция не выполняется, SM0 = ON и в регистре SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнд								Функция				
2122		SPLIT	P	S <sub>1</sub> , S <sub>2</sub> , S <sub>3</sub> , S <sub>4</sub> , D <sub>1</sub> , D <sub>2</sub>								Разделение строковой переменной				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>								●	●							
S <sub>2</sub>								●					○	○	○	
S <sub>3</sub>								●					○	○	○	
S <sub>4</sub>								●					○	○		
D <sub>1</sub>								●								
D <sub>2</sub>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							●
S <sub>3</sub>		●			●	●							●
S <sub>4</sub>		●			●	●							
D <sub>1</sub>		●			●	●							
D <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

SPLIT		SPLITP	
En		En	
S1	D1	S1	D1
S2	D2	S2	D2
S3		S3	
S4		S4	

- S<sub>1</sub> : Строковая переменная
- S<sub>2</sub> : Разделитель
- S<sub>3</sub> : Обозначенный конечный символ разделителя
- S<sub>4</sub> : Длина сохраненной строковой переменной после разделения
- D<sub>1</sub> : Операнд сохранения строковой переменной после разделения
- D<sub>2</sub> : Операнд сохранения количества разделенных строковых переменных

**Описание:**

- Эта инструкция доступна только для прошивки версии 1.04 или выше.
- S<sub>1</sub> – источник строковой переменной. Для символов в строковой переменной в качестве допустимых используются только символы перед конечным символом или значением в S<sub>3</sub>.
- S<sub>2</sub> – это разделитель, который разделяет строковые переменные. S<sub>3</sub> – обозначенный конечный символ разделителя. Для S<sub>2</sub> и S<sub>3</sub> действительны только младшие 8 байт значений. Однако, если вы вводите значение, отличное от значения в младших 8 байтах, ПЛК не покажет сообщение об ошибке. Когда S<sub>2</sub> и S<sub>3</sub> используют операнд D для сохранения значений, используйте только младшие 8 байт значений в формате ASCII. Значения в S<sub>2</sub> и S<sub>3</sub> могут быть постоянным числом, кодом ASCII или строковой переменной.
- S<sub>4</sub> – длина сохраненной строковой переменной после разделения; формат словный: WORD. Конечные символы, такие как 16#00 и 16# 000, должны быть включены в длину. Максимальные значения в словном формате WORD, которые может содержать строка, это S<sub>4</sub>, а слово WORD содержит 2 байта.

Конечный символ 16#00 содержит байт. Например, если S<sub>4</sub> равно 2, он может максимально содержать 2 слова или

4 байта. И это означает, что строковая переменная может содержать 3 символа и 1 конечный символ (16#00). Диапазон настройки для  $S_4$  равен 1-100. (Если значение выходит за пределы диапазона, это значение будет отображаться как минимальное или максимальное, при этом ПЛК не показывает сообщений об ошибке).

- 5.  $D_2$  – количество всех разделенных строковых переменных.
- 6. Результат  $D_1 \sim D_1 + S_4 * D_2 - 1$  – это занятые операнды  $D$ .

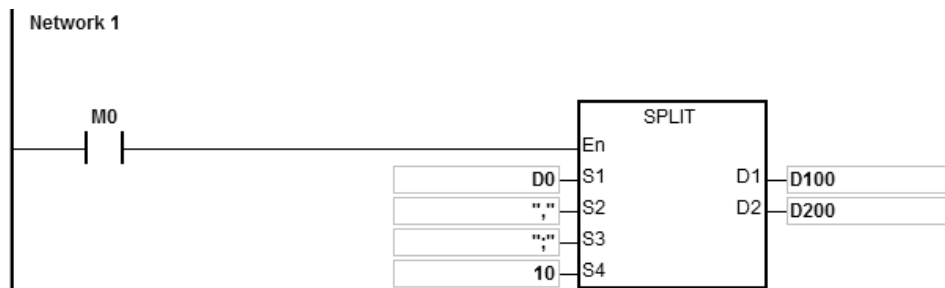
Например, если значение в  $D_2$  равно 3, это означает, что 1-я строковая переменная будет сохранена в  $D_1 + 0$ , а вторая будет сохранена в  $D_1 + S_4$ , а третья будет сохранена в  $D_1 + S_4 * 2$ .

- 7. Команда ищет строковую переменную с длиной, определенной в  $S_4$  (в словах), до тех пор, пока не будет найден конечный символ или разделитель. Добавьте 1 к значению в  $D_2$  после обнаружения каждой строковой переменной.
- 8. В  $S_1$  может храниться до 256 символов, в  $D_2$  может храниться до 20 наборов строковых переменных. Когда достигается какое-либо ограничение даже при отсутствии конечного символа, разделение прекращается.

**Пример:**

Используйте «,» в качестве разделителя для разделения строковой переменной «abcd, 1234,5.67,8910; ef» и используйте «;» в качестве конечного символа, который сохраняется в D0-D11. И сохраните результаты в регистре, начиная с D100, в блоке из 10 слов.

**Примечание:** строковая переменная, которая должна быть разделена, не включает символы «».



**Описание:**

- 1. Результат для этого примера: D0~D11="abcd,1234,5.67,89103;edf".

Операнд	Значение	Строковая переменная
D0	16#6261	"ba"
D1	16#6463	"dc"
D2	16#312C	"1,"
D3	16#3332	32
D4	16#2C34	","
D5	16#2E35	".5"
D6	16#3736	"76"
D7	16#382C	"8,"
D8	16#3139	"19"
D9	16#3330	"30"
D10	16#453B	"e;"
D11	16#4644	"fd"

- 2. Исходным контактом является M0
- 3. Количество разделенных строковых переменных  $D200 = 4$
- 4. Разделенные строковые переменные приведены ниже:

1 набор: Заняты D100~D109; D100-D101="abcd"	D100	D101	D102	D103	D104	D105	D106	D107	D108	D109
Шестнадцатеричные	6261	6463	0000	0000	0000	0000	0000	0000	0000	0000
Строковые	"ba"	"dc"	16#0000	-	-	-	-	-	-	-
2 набор: Заняты D110~D119; D110-D111="1234"	D110	D111	D112	D113	D114	D115	D116	D117	D118	D119
Шестнадцатеричные	3231	3433	0000	0000	0000	0000	0000	0000	0000	0000
Строковые	"21"	"43"	16#0000	-	-	-	-	-	-	-

3 набор: Заняты D120~D129; D120~D121="5.67"

Операнды	D120	D121	D122	D123	D124	D125	D126	D127	D128	D129
Шестнадцатеричные	2E35	3736	0000	0000	0000	0000	0000	0000	0000	0000
Строковые	".5"	".76"	16#0000	-	-	-	-	-	-	-

4 набор: Заняты D130~D139, D130~D131="89103"

Операнды	D130	D131	D132	D133	D134	D135	D136	D137	D138	D139
Шестнадцатеричные	3938	3031	0033	0000	0000	0000	0000	0000	0000	0000
Строковые	"98"	"01"	Старш.: 3" Младш.: 16#00	-	-	-	-	-	-	-

**Примечания:**

1. В **S<sub>1</sub>** могут быть сохранены до 256 символов, если нет конечного символа (16#00) или разделителя, инструкция работает с 256 символами. При этом SM0 включен не будет.
2. Конечный символ 16#00 или 16#0000 будет добавлен в строковую переменную после ее разделения (**D<sub>1</sub>**). См. Пример выше.
3. Конечный символ 16#00 или 16#0000 будет добавлен в строковую переменную после ее разделения (**D<sub>1</sub>**), даже если нет после разделенной строковой переменной другой строковой переменной.

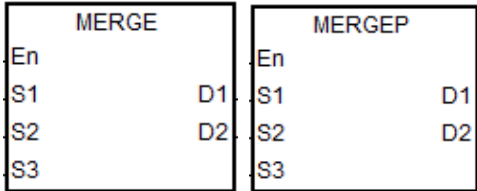
API	Код инструкции			Операнд							Функция					
2123		MERGE	P	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D<sub>1</sub>, D<sub>2</sub></b>							Объединение строковой переменной					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>								●	●						○	
S <sub>2</sub>								●							○	
S <sub>3</sub>								●					○	○	○	
D <sub>1</sub>								●								
D <sub>2</sub>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	String
S <sub>1</sub>		●			●	●							●
S <sub>2</sub>		●			●	●							●
S <sub>3</sub>		●			●	●							●
D <sub>1</sub>		●			●	●							
D <sub>2</sub>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**



- S<sub>1</sub> : Источник строковой переменной 1
- S<sub>2</sub> : Источник строковой переменной 2
- S<sub>3</sub> : Разделитель
- D<sub>1</sub> : Операнд сохранения строковой переменной после объединения
- D<sub>2</sub> : Операнд сохранения количества объединенных строковых переменных

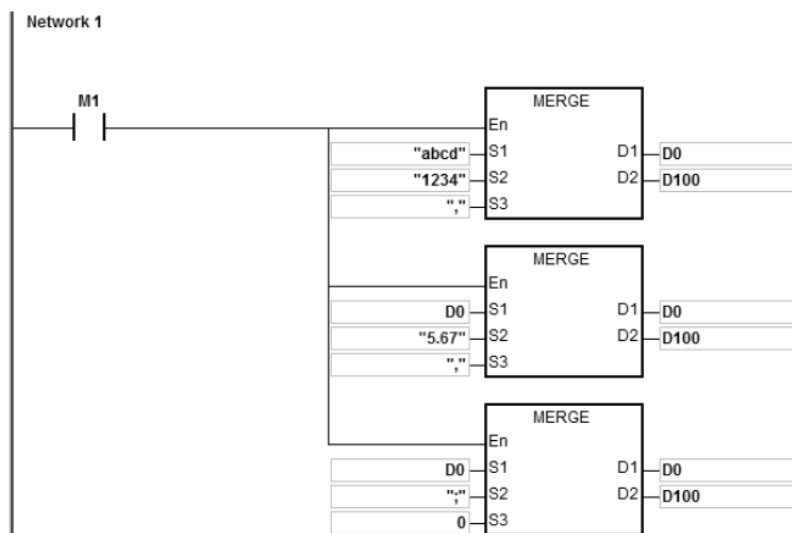
**Описание:**

1. Эта инструкция доступна только для прошивки версии 1.04 или выше.
  2. S<sub>1</sub> и S<sub>2</sub> являются источником 1 и 2 строковых переменных соответственно. S<sub>3</sub> – это их разделитель. Для S<sub>3</sub> действительны только младшие 8 байт значения. Строковые переменные в S<sub>1</sub> и S<sub>2</sub> соединяются напрямую, когда S<sub>3</sub> равно 0 (16#00). Когда S<sub>3</sub> не равен 0, к S<sub>1</sub> добавляется разделитель и затем она объединяется с S<sub>2</sub>.
  3. D<sub>1</sub> – это место сохранения объединенной строковой переменной. Порядок слияния: строковая переменная 1 + разделитель + строковая переменная 2 + конечный символ 0 (16#00).
- Вы можете самостоятельно определить конечный символ. Этот свой конечный символ будет добавлен до стандартного конечного символа 0 (16#00).
4. Если значения в S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub> равны 0, значение в D<sub>1</sub> равно 16#0000.
  5. D<sub>2</sub> – количество всех объединенных строковых переменных (исключая символ 16#00). суммарная длина для строковой переменной – это длина строковой переменной 1 или длина строковой переменной 1 + разделитель + длина строковой переменной 2.
  6. Если значения S<sub>1</sub> и S<sub>2</sub> равны 0 или когда строковые переменные объединены, длина объединенной строковой переменной составляет до 255 слов (исключая символ 16#00). Например, есть 250 слов в строкой переменной 1 и 100 слов в строкой переменной 2. После их объединения длина для объединенной строкой переменной

составляет 250 слов из строкой переменной 1 + разделитель \* 1 + первые 4 слова из строкой переменной 2.

**Пример:**

Используем «,» в качестве разделителя, чтобы разбить строковую переменную, затем используем «;» в качестве конечного символа, а затем объединим 3 строковых переменных «abcd», «1234» и «5.67».



Результат «abcd, 1234,5.67;» сохраняется в D100 = 15, в регистрах, начиная с D0-D7.

**Описание:**

1. Стартовым контактом является M0.
2. Первая программа: объединить 1 строковую переменную «abcd» и 2 строковую переменную «1234» вместе и поместить разделитель «,» между ними, а затем сохранить результат в регистрах, начиная с D0.

Результат объединения: D0 ~ D4 = "abcd, 1234"; количество объединенных строковых переменных: D100 = 9.

Операнды	D0	D1	D2	D3	D4
Шестнадцатеричные	6261	6463	312C	3332	0034
Строковые	"ba"	"dc"	"1,"	"32"	Старш. 8 байт: 16#00 ; младш. 8 байт: 4

3. Вторая программа: объединить 1 строковую переменную D0 = «abcd, 1234» и 2 строковую переменную «5.67» вместе и поместить разделитель «,» между ними и затем сохранить результат в регистрах, начиная с D0.

Результат объединения: D0 ~ D7 = "abcd, 1234,5.67"; количество объединенных строковых переменных: D100 = 14.

Операнды	D0	D1	D2	D3	D4	D5	D6	D7
Шестнадцатеричные	6261	6463	312C	3332	2C34	2E35	3736	0000
Строковые	"ba"	"dc"	"1,"	"32"	"4"	".5"	"76"	16#0000

4. Третья программа: объединить 1 строковую переменную D0 = "abcd, 1234,5.67" и 2 строковую переменную ", " вместе и поместить разделитель "0" между ними (объединить строковые переменные 1 и 2 напрямую), а затем сохранить результат в регистрах, начиная с D0.

Результат объединения: D0 ~ D7 = "abcd, 1234,5.67;"; количество объединенных строковых переменных: D100 = 15.

Операнды	D0	D1	D2	D3	D4	D5	D6	D7
Шестнадцатеричные	6261	6463	312C	3332	2C34	2E35	3736	003B
Строковые	"ba"	"dc"	"1,"	"32"	"4"	".5"	"76"	Старш. 8 байт: 16#00 ; младш. 8 байт: ","

**Примечания:**

1. Если строковая переменная в **S<sub>1</sub>** не заканчивается символом 16#00, включается флаг SM0, а в SR0 записывается код ошибки 16#200E.
2. Если строковая переменная в **S<sub>2</sub>** не заканчивается символом 16#00, включается флаг SM0, а в SR0 записывается код ошибки 16#200E.
3. Если значение  $D_1+(D_2+1)/2-1$  выходит за допустимые пределы, инструкция не выполняется, включается флаг SM0, а в SR0 записывается код ошибки 16#2003.
4. За объединенной строковой переменной будет добавлен конечный символ 16#00 или 16#0000. См. примеры выше.

## 6.22 Инструкции для Ethernet

### 6.22.1 Описание инструкций для Ethernet

API	Код инструкции			Операнды								Функция				
2200		SOPEN	P	$S_1, S_2, S_3$								Открытие сокета				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●			○	○	○		
$S_2$	●	●			●	●		●	●			○	○	○		
$S_3$	●	●			●	●		●	●			○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

	SOPEN	SOPENP
En		En
S1		S1
S2		S2
S3		S3

$S_1$  : Режим сокета

$S_2$  : Номер сокета

$S_3$  : Режим старта

Описание:

- $S_1$  равен 1 если пользователь открывает TCP сокет и  $S_1$  равен 0 если пользователь открывает UDP сокет.  $S_2$  это номер сокета. Если  $S_3$  равен 1, ПЛК серии AS в качестве клиента (Client) отправляет запрос TCP-соединения на сервер (Server), а если  $S_3$  равен 0, ПЛК серии AS в качестве сервера ожидает запроса TCP-соединения от клиента. Если необходимо запустить соединение UDP,  $S_3$  будет равен соответственно 0 и 1.
- Операнд  $S_1$  может иметь значение 0 или 1; значение операнда  $S_2$  находится в диапазоне от 1 до 4; операнд  $S_3$  может иметь значение только 0 или 1.
- Перед использованием инструкции необходимо выполнить следующие настройки в HWCONFIG ПО ISPSOft.
  - PLC Parameter Setting→Ethernet-Basic→Setting the IP address (Задание IP адреса) и адреса маски подсети



- PLC Parameter Setting→Ethernet-Advance→Socket→Enable Socket Function (Разрешение функции сокета)
  - PLC Parameter Setting→Ethernet-Advance→Socket→TCP/UDP Socket Connection (Установка режима сокета) и установка используемых сокетов.
4. См. описание инструкции API2209 SCONF для задания параметров TCP и UDP сокета.
5. Когда открыт TCP сокет, настройки IP сокета и порта связи показаны в таблице ниже.

Режим старта	Удаленный IP	Локальный порт связи	Удаленный порт связи	Описание
1	Конкретный IP адрес	0	0	Недействителен
1	Конкретный IP адрес	0	Не равен 0	Client режим, Конкретный IP адрес; локальный порт связи не указан
1	Конкретный IP адрес	Не равен 0	0	Недействителен
1	Конкретный IP адрес	Не равен 0	Не равен 0	Client режим, Конкретный IP адрес, указаны локальный и удаленный порты связи
1	0.0.0.0	Ограничений на значение нет	Ограничений на значение нет	Недействителен
0	Конкретный IP адрес	0	Ограничений на значение нет	Недействителен
0	Конкретный IP адрес	Не равен 0	0	Server режим, Конкретный IP адрес; удаленный порт связи не указан.
0	Конкретный IP адрес	Не равен 0	Не равен 0	Slave режим, Конкретный IP адрес, указан удаленный порт связи.
0	0.0.0.0	0	Ограничений на значение нет	Недействителен
0	0.0.0.0	Не равен 0	0	Server режим, Не указаны конкретный IP адрес и

Режим старта	Удаленный IP	Локальный порт связи	Удаленный порт связи	Описание
				удаленный порт связи.
0	0.0.0.0	Не равен 0	Не равен 0	Server режим, Не указан конкретный IP адрес, указан удаленный порт связи.

6. Если данные передаются через сокет TCP, и после выполнения не возникает ошибок, сокет начинает устанавливать соединение с удаленным устройством, а флаг запуска соединения включается. Если соединение выполнено успешно, флаг успешного соединения включается, а флаг запуска соединения отключается. Если в процессе возникает ошибка, включается флаг ошибки.

Номер сокета TCP	Успешное соединение	Данные получены	Данные переданы	Запуск соединения	Закрытие соединения	Отправка данных	Флаг ошибки
1	SM1270	SM1271	SM1272	SM1273	SM1274	SM1275	SM1277
2	SM1278	SM1279	SM1280	SM1281	SM1282	SM1283	SM1285
3	SM1286	SM1287	SM1288	SM1289	SM1290	SM1291	SM1293
4	SM1294	SM1295	SM1296	SM1297	SM1298	SM1299	SM1301

7. Если данные передаются через сокет UDP, и после выполнения не возникает ошибок, включается флаг запуска соединения. Если в процессе возникает ошибка, включается флаг ошибки.

Номер сокета UDP	Запуск соединения	Данные получены	Данные переданы	Закрытие соединения	Флаг ошибки
1	SM1334	SM1335	SM1336	SM1337	SM1338
2	SM1339	SM1340	SM1341	SM1342	SM1343
3	SM1344	SM1345	SM1346	SM1347	SM1348
4	SM1349	SM1350	SM1351	SM1352	SM1353

8. Если данные передаются через TCP Socket в Server режиме или через UDP Socket, не разрешается использовать следующие номера портов связи для ПЛК серии AS.

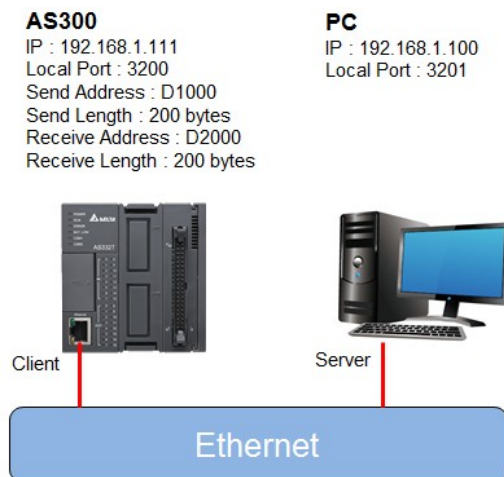
Тип сокета	Номер порта связи	Протокол связи
TCP	80	HTTP
TCP	502	MODBUS TCP
TCP	44818	EtherNet/IP
UDP	68	DHCP/BOOTP
UDP	2222	EtherNet/IP
UDP	44818	EtherNet/IP
UDP	20006	Самоопределяемый протокол для

		серии AS
--	--	----------

9. В основном, используется импульсная инструкция SOPENP.

**Пример 1:**

1. Системная структура ниже иллюстрирует, как установить TCP-соединение между компьютером как сервером (Server) и ПЛК серии AS в качестве клиента (Client).



2. ISPSOft -> HWCONFIG (Ethernet – Basic)

Конфигурация Ethernet	
Режим адресации IP	Статический
IP адрес	192.168.1.111
Адрес маски подсети	255.255.255.0
Адрес шлюза	192.168.1.1

3. а ) ISPSOft -> HWCONFIG ( Ethernet – Advance > Socket > TCP Socket Connection )

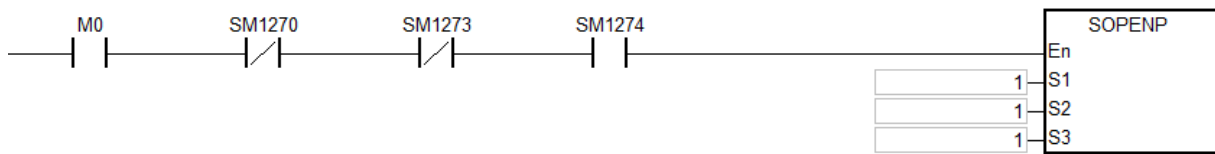
б ) Инструкция SCONF

Подключение сокета TCP	
Удаленный IP	192.168.1.110
Удаленный порт	3201
Локальный порт	3200
Адрес отправляемых данных	D1000
Длина отправляемых данных	200
Адрес принимаемых данных	D2000
Длина принимаемых данных	200

Подключение сокета TCP	
Таймер удержания	10

4. Параметры, относящиеся к сокету TCP, сначала задаются в HWCONFIG ПО ISPSOFT или инструкцией SCONF.
5. Когда M0 включен, независимо от того, был ли сокет закрыт, проверяется подключение. Если сокет не закрыт и не подключен, выполняется подключение. После того, как сокет подключен, M0 отключается.
6. Когда M1 включен, проверяется подключение сокета и отправка данных. Если сокет подключен и данные не отправлены, данные будут отправлены. Если сокет не подключен, инструкция не будет выполнена. По завершении отправки данных M1 отключается.
7. Когда M2 включен, проверяется подключение сокета и состояние флага получения данных. Если сокет подключен и флаг приема данных включен, это означает, что данные получены.
8. Когда M3 включен, проверяется подключение сокета. Если сокет подключен, соединение будет закрыто. Если сокет не подключен, инструкция не будет выполнена. После того, как соединение будет закрыто, M2 и M3 отключаются.

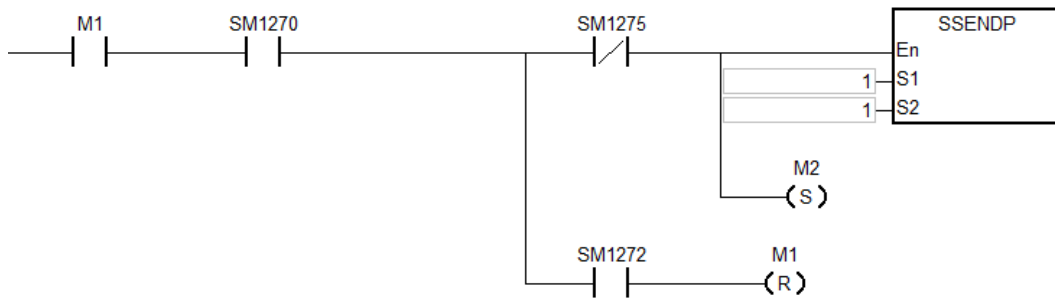
Network 1



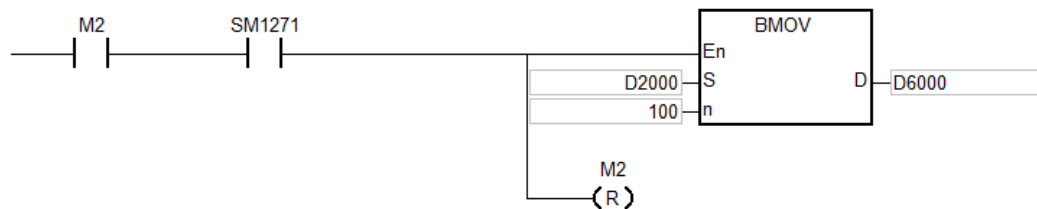
Network 2



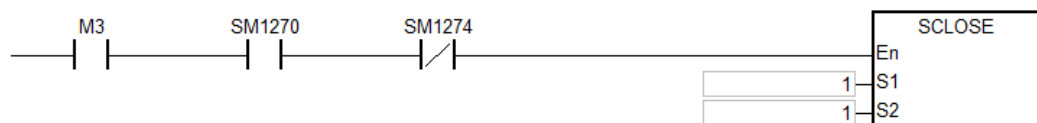
Network 3



Network 4

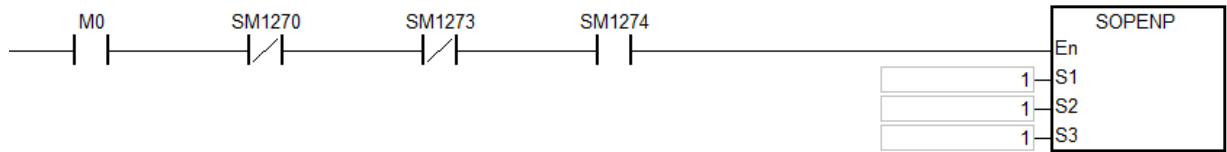


Network 5



6

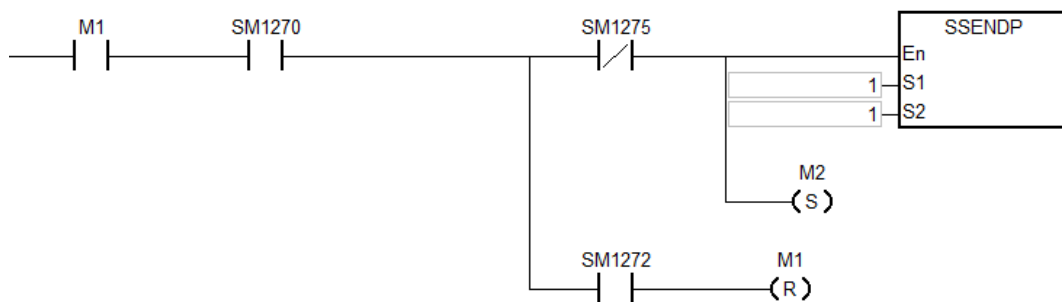
Network 1



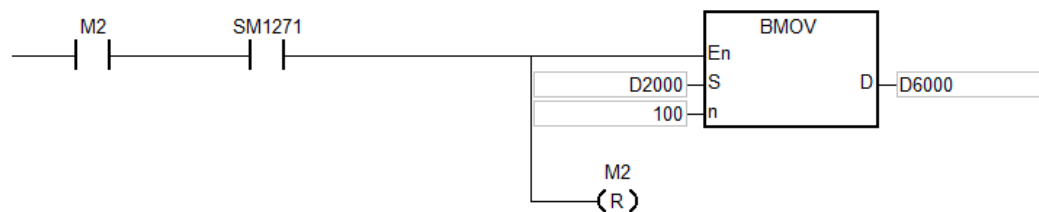
Network 2



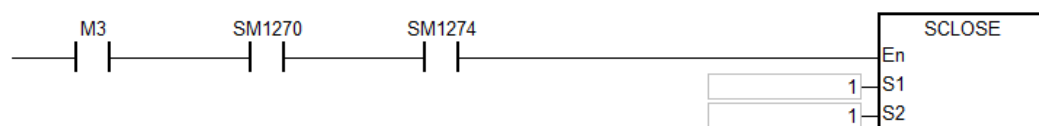
Network 3



Network 4



Network 5



**Пример 2:**

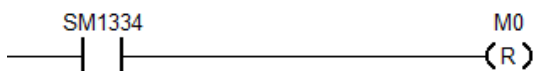
1. Пример иллюстрирует, как установить соединение UDP между компьютером и ПЛК серии AS.
2. Когда M0 включен, проверяется подключение сокета. Если сокет не подключен, выполняется процедура подключения. После того, как сокет подключен, M0 отключается.
3. Когда M1 включен, данные отправляются. По завершении отправки данных M1 отключается.
4. Когда M2 включен, проверяется подключение сокета и состояние флага получения данных. Если сокет подключен и флаг приема данных включен, это означает, что данные получены.

5. Когда M3 включен, проверяется подключение сокета. Если сокет подключен, соединение будет закрыто. Если сокет не подключен, инструкция не будет выполнена. После того, как соединение будет закрыто, M2 и M3 отключаются.

Network 1



Network 2



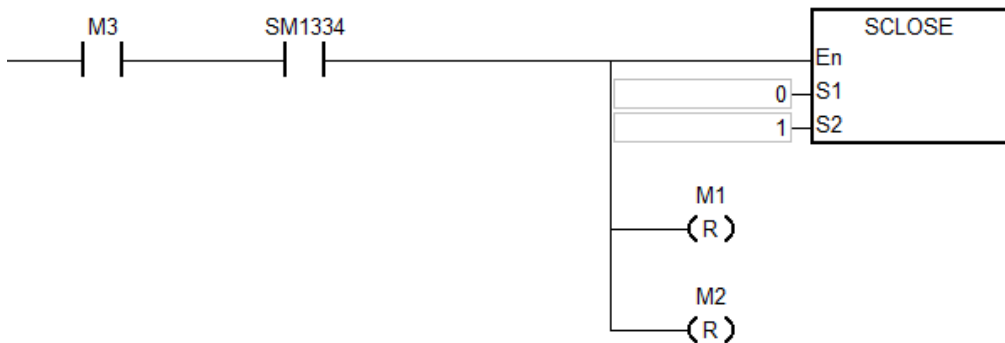
Network 3



Network 4



Network 5



**Примечание:**

1. Ошибки соединения TCP показаны ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	Значения <b>S<sub>1</sub></b> , <b>S<sub>2</sub></b> или <b>S<sub>3</sub></b> выходят за допустимые пределы
16#600C	SM1109	Используется локальный сокет
16#600D	SM1100	Сеть Ethernet не подключена
16#6200	Флаг ошибки сокета TCP	Недопустимый IP-адрес
16#6201	Флаг ошибки сокета TCP	Недействительный режим связи сокета TCP
16#6202	Флаг ошибки сокета TCP	Неверная настройка режима сокета TCP
16#6203	Флаг ошибки сокета TCP	Недопустимый адрес для отправки данных
16#6204	Флаг ошибки сокета TCP	Длина отправленных данных превышает допустимый диапазон
16#6205	Флаг ошибки сокета TCP	Устройство, куда отправляются данные, находится вне допустимого диапазона
16#6206	Флаг ошибки сокета TCP	Недопустимый адрес для получения данных
16#6207	Флаг ошибки сокета TCP	Длина полученных данных превышает допустимый диапазон
16#6208	Флаг ошибки сокета TCP	Устройство для приема данных находится вне допустимого диапазона
16#6212	Флаг ошибки сокета TCP	Превышение ожидания при связи через сокет TCP
16#6213	Флаг ошибки сокета TCP	Размер фактически полученных данных больше заданных данных
16#6214	Флаг ошибки сокета TCP	Соединение с сокетом TCP отклонено удаленным оборудованием
16#6215	Флаг ошибки сокета TCP	Сокет TCP не подключен
16#6217	Флаг ошибки сокета TCP	Соединение сокета TCP переключено
16#6218	Флаг ошибки сокета TCP	Сбой передачи данных через сокет TCP
16#621A	Флаг ошибки сокета TCP	Подключение сокета TCP отключено

2. Ошибки соединения UDP показаны ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	Значения <b>S<sub>1</sub></b> , <b>S<sub>2</sub></b> или <b>S<sub>3</sub></b> выходят за допустимые пределы



Код ошибки	Флаг ошибки	Описание
16#600C	SM1109	Используется локальный сокет
16#600D	SM1100	Сеть Ethernet не подключена
16#6209	Флаг ошибки сокета UDP	Недопустимый IP-адрес
16#620A	Флаг ошибки сокета UDP	Недопустимый режим связи
16#620C	Флаг ошибки сокета UDP	Недопустимый адрес для отправки данных
16#620D	Флаг ошибки сокета UDP	Длина отправленных данных превышает допустимый диапазон
16#620E	Флаг ошибки сокета UDP	Устройство, куда отправляются данные, находится вне допустимого диапазона
16#620F	Флаг ошибки сокета UDP	Недопустимый адрес для получения данных
16#6210	Флаг ошибки сокета UDP	Длина фактически полученных данных превышает допустимый диапазон
16#6211	Флаг ошибки сокета UDP	Устройство для приема данных находится вне допустимого диапазона
16#6213	Флаг ошибки сокета UDP	Размер фактически полученных данных больше заданных данных
16#6215	Флаг ошибки сокета UDP	Сокет UDP не подключен
16#6217	Флаг ошибки сокета UDP	Соединение с сокетом UDP переключено
16#6218	Флаг ошибки сокета UDP	Сработала отправка данных через сокет UDP
16#621A	Флаг ошибки сокета UDP	Соединение с сокетом UDP отключено

- 6
3. Когда клиентом (Client) и сервером (Server) являются ПЛК серии AS, а настройки ожидания одинаковы, сервер автоматически отключит соединение, если превышение ожидания произойдет сначала на сервере. Таким образом, флаг ошибки в клиенте включен не будет. Если превышение ожидания произойдет сначала в клиенте, флаг ошибки в клиенте будет включен и соединение будет сброшено.

API	Код инструкции			Операнды							Функция						
2201		SSEND	P	$S_1, S_2$							Отправка данных через сокет						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●			○	○	○		
$S_2$	●	●			●	●		●	●			○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

	SSEND	SSENDP
En		En
S1		S1
S2		S2

$S_1$  : Режим сокета

$S_2$  : Номер сокета

Описание:

- $S_1$  равен 1 если пользователь открывает TCP сокет и  $S_1$  равен 0 если пользователь открывает UDP сокет.  $S_2$  это номер сокета.
- Операнд  $S_1$  может иметь значение 0 или 1; значение операнда  $S_2$  находится в диапазоне от 1 до 4.
- Операнд  $S_1$  должен быть равен 0 или 1 (по умолчанию: 0); операнд  $S_2$  должен быть между 1-4 (по умолчанию: 1). Если значения превышают диапазон ПЛК отображает код ошибки N'2003.
- Перед применением данной инструкции необходимо выполнить инструкцию SOPEN для открытия сокета. Если включен флаг успешного соединения (сокета TCP) или включен флаг запуска соединения (сокета UDP), эта инструкция может быть выполнена.
- Если данные передаются через сокет TCP, и после выполнения не возникает ошибок, сокет начинает устанавливать соединение с удаленным устройством, а флаг запуска соединения включается. Если соединение выполнено успешно, флаг успешного соединения включается, а флаг запуска соединения отключается. Если в процессе возникает ошибка, включается флаг ошибки.

Номер сокета TCP	Отправка данных	Данные отправлены	Флаг ошибки
1	SM1275	SM1272	SM1277
2	SM1283	SM1280	SM1285

3	SM1291	SM1288	SM1293
4	SM1299	SM1296	SM1301

6. Если данные передаются через сокет UDP, и после выполнения не возникает ошибок, включается флаг запуска соединения. Если в процессе возникает ошибка, включается флаг ошибки.

Номер сокета UDP	Данные переданы	Флаг ошибки
1	SM1336	SM1338
2	SM1341	SM1343
3	SM1346	SM1348
4	SM1351	SM1353

7. В основном, используется импульсная инструкция SSENDP.

#### Пример:

См. пример применения инструкции SOPEN.

#### Примечание:

1. Ошибки соединения TCP показаны ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	Значения $S_1$ или $S_2$ выходят за допустимые пределы
16#600D	SM1100	Сеть Ethernet не подключена
16#6202	Флаг ошибки сокета TCP	Неверная настройка режима сокета TCP
16#6203	Флаг ошибки сокета TCP	Недопустимый адрес для отправки данных
16#6204	Флаг ошибки сокета TCP	Длина отправленных данных превышает допустимый диапазон
16#6205	Флаг ошибки сокета TCP	Устройство, куда отправляются данные, находится вне допустимого диапазона
16#6212	Флаг ошибки сокета TCP	Превышение ожидания при связи через сокет TCP
16#6214	Флаг ошибки сокета TCP	Соединение с сокетом TCP отклонено удаленным оборудованием
16#6215	Флаг ошибки сокета TCP	Сокет TCP не подключен
16#6218	Флаг ошибки сокета TCP	Отправка данных была переключена

2. Ошибки соединения UDP показаны ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	Значения <b>S<sub>1</sub></b> или <b>S<sub>2</sub></b> выходят за допустимые пределы
16#600D	SM1100	Сеть Ethernet не подключена
16#620A	Флаг ошибки сокета UDP	Неверная настройка режима сокета UDP
16#620C	Флаг ошибки сокета UDP	Недопустимый адрес для отправки данных
16#620D	Флаг ошибки сокета UDP	Длина отправленных данных превышает допустимый диапазон
16#620E	Флаг ошибки сокета UDP	Устройство, куда отправляются данные, находится вне допустимого диапазона
16#6218	Флаг ошибки сокета UDP	Отправка данных была переключена

API	Код инструкции			Операнды					Функция				
2203		SCLOSE	P	<b>S<sub>1</sub>, S<sub>2</sub></b>					Закрытие сокета				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S<sub>1</sub></b>	●	●			●	●		●	●			○	○	○		
<b>S<sub>2</sub></b>	●	●			●	●		●	●			○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>		●			●	●							
<b>S<sub>2</sub></b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

	SCLOSE	SCLOSEP
En		En
S1		S1
S2		S2

**S<sub>1</sub>** : Режим сокета

**S<sub>2</sub>** : Номер сокета

Описание:

- S<sub>1</sub>** равен 1 если пользователь открывает TCP сокет и **S<sub>1</sub>** равен 0 если пользователь открывает UDP сокет. **S<sub>2</sub>** это номер сокета.
- Операнд **S<sub>1</sub>** может иметь значение 0 или 1; значение операнда **S<sub>2</sub>** находится в диапазоне от 1 до 4.
- Перед закрытием сокета пользователям необходимо убедиться, что он подключен. В противном случае инструкция не выполняется.
- Если сокет TCP закрыт клиентом (Client), сервер (Server) продолжает подключаться к локальному порту связи (включен флаг успешного соединения). Если сокет TCP закрыт сервером, после завершения выполнения инструкции сервер не будет подключен к локальному коммуникационному порту. После закрытия сокета TCP соответствующие флаги будут отключены.
- После выполнения инструкции и закрытия сокета UDP соответствующий флаг отключается.
- Если сокет TCP закрыт и после выполнения команды не возникает ошибок, соединение с удаленным устройством начинает закрываться, и включается флаг закрытия соединения. Если соединение успешно закрыто, флаг закрытия соединения отключается. Если в процессе возникает ошибка, включается флаг ошибки.

Номер сокета TCP	Флаг закрытия соединения	Флаг ошибки
1	SM1274	SM1277

Номер сокета TCP	Флаг закрытия соединения	Флаг ошибки
2	SM1282	SM1285
3	SM1290	SM1293
4	SM1298	SM1301

7. Если сокет UDP закрыт и после выполнения команды не возникает ошибок, флаг запуска подключения отключается. Если в процессе возникает ошибка, включается флаг ошибки.

Номер сокета UDP	Флаг ошибки
1	SM1338
2	SM1343
3	SM1348
4	SM1353

8. В основном, используется импульсная инструкция SCLOSEP.

**Пример:**

См. пример применения инструкции SOPEN.

**Примечание:**

1. Ошибки соединения TCP представлены ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	Значения $S_1$ или $S_2$ выходят за допустимые пределы
16#600D	SM1100	Сеть Ethernet не подключена
16#6212	Флаг ошибки сокета TCP	Превышение ожидания при связи через сокет TCP
16#6214	Флаг ошибки сокета TCP	Соединение с сокетом TCP отклонено удаленным оборудованием
16#621A	Флаг ошибки сокета TCP	Соединение сокета TCP переключено

2. Ошибки соединения UDP представлены ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	Значения $S_1$ или $S_2$ выходят за допустимые пределы
16#600D	SM1100	Сеть Ethernet не подключена
16#621A	Флаг ошибки сокета UDP	Соединение с сокетом UDP переключено

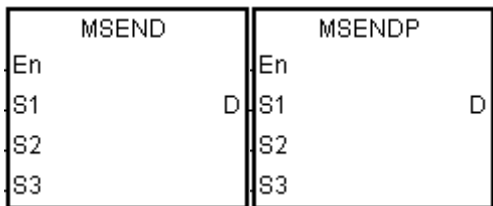
API	Код инструкции			Операнды							Функция					
2204		MSEND	P	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D</b>							Отправка email					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S<sub>1</sub></b>	●	●			●	●		●	●			○	○	○		
<b>S<sub>2</sub></b>	●	●			●	●		●	●							
<b>S<sub>3</sub></b>	●	●			●	●		●	●							
<b>D</b>	●	●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>		●			●	●							
<b>S<sub>2</sub></b>		●			●	●							
<b>S<sub>3</sub></b>		●			●	●							
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**



- S<sub>1</sub>** : Номер удаленной электронной почты
- S<sub>2</sub>** : Тема письма
- S<sub>3</sub>** : Содержание письма
- D** : Индикатор выполнения инструкции

**Описание:**

1. Перед отправкой письма необходимо задать значения **S<sub>1</sub>**, **S<sub>2</sub>** и **S<sub>3</sub>**.
2. Перед выполнением инструкции необходимо выполнить следующие шаги в ПО ISPSOft.
  - PLC Parameter Setting→Ethernet-Basic→Setting the IP address and the netmask address (Настройка IP-адреса и адреса маски подсети).
  - PLC Parameter Setting→Ethernet-Advance→Email→Setting the SMTP server, the port, the local email address, and the SMTP subject (Настройка SMTP сервера, порта, локального адреса email и темы SMTP)
  - PLC Parameter Setting→Ethernet-Advance→Email and Trigger Configuration→Setting the email address
  - Если требуется идентификация аккаунта, PLC Parameter Setting→Ethernet-Advance→Email→Setting the user name and the password (Настройка имени пользователя и пароля).

3. Адрес email задается следующим образом.

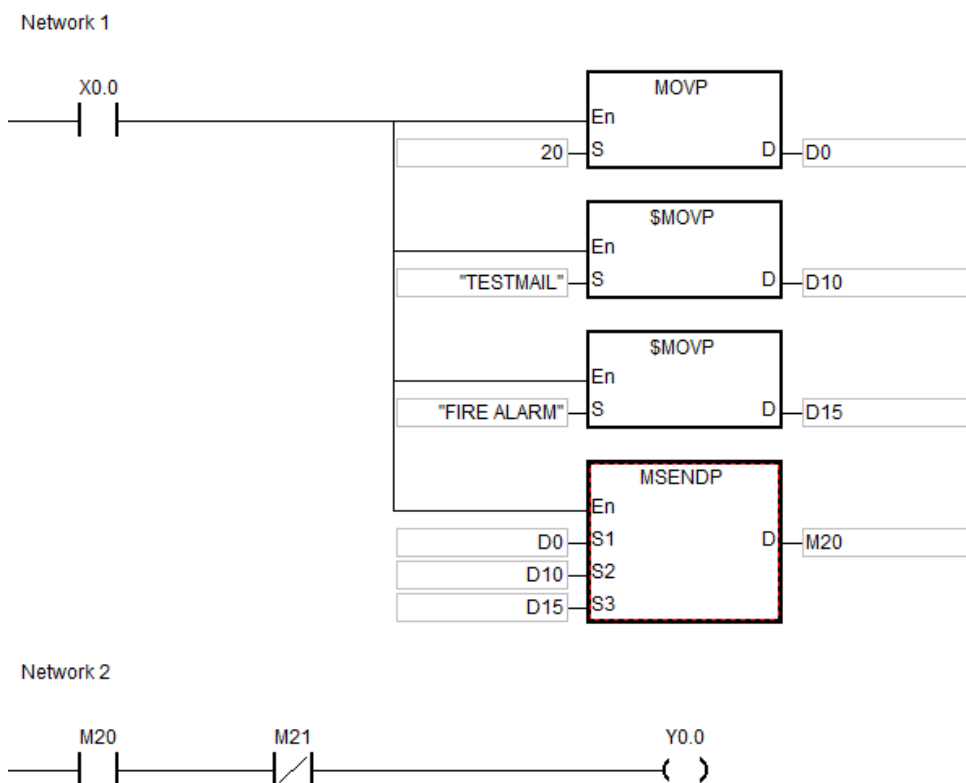
Операнд	Описание	Диапазон настройки
<b>S<sub>1</sub></b>	Номер удаленной электронной почты	Значение <b>S<sub>1</sub></b> должно находиться между 1 и 15. Бит0 ~ бит3, заданные в ISPSoft, указывают удаленные адреса электронной почты (в ISPSoft можно задать 4 адреса электронной почты). Bit0 соответствует удаленной электронной почте 1; бит1 соответствует удаленной электронной почте 2 и так далее. Для отправки email необходимо задать соответствующие биты в ПО ISPSoft
<b>S<sub>2</sub></b>	Тема email	Размер темы письма может составлять до 16 слов.
<b>S<sub>3</sub></b>	Содержимое email	Размер содержимого электронной почты может составлять до 64 слов.
<b>D</b>	Индикатор выполнения инструкции	По завершении выполнения инструкции включается соответствующий бит. Если выполнение инструкции произошло с ошибкой, включается следующий бит.

4. В основном, выполняется импульсная инструкция MSENDP.

**Пример:**

Предположим, что значение в D0 равно 00010100. Когда X0.0 включен, email отправляется на номер удаленной электронной почты 3 и номер удаленной электронной почты 5. После завершения связи с SMTP-сервером включается M20. Если во время связи ошибок не возникло, M21 выключается и включается Y0.0.





**Примечание:**

1. Для строковых переменных в **S<sub>2</sub>** или **S<sub>3</sub>** система будет захватывать данные с конечным символом 16#00. Если размер строковой переменной больше максимального значения (включая символ 16#00 в конце), размер строковой переменной в **S<sub>2</sub>** или **S<sub>3</sub>** будет считаться равный максимальному значению.
2. Если пользователи объявляют операнд **D** в ISPSOft, тип данных будет ARRAY [2] для WORD/INT.
3. Обязательно зарезервируйте одно слово между **S<sub>2</sub>** и **S<sub>3</sub>** для символа прерывания.
4. Ошибки при выполнении инструкции показаны ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0	1. <b>D+1</b> превышает допустимый предел 2. <b>S<sub>1</sub>&lt;1</b> или <b>S<sub>1</sub>&gt;15</b>
16#600D	SM1100	Сеть Ethernet не подключена
16#6100	D+1	Конфликт оборудования при коммуникации
16#6107	D+1	Превышение времени ожидания связи по электронной почте
16#6108	D+1	Ошибка аутентификации пароля аккаунта SMTP-сервера

Код ошибки	Флаг ошибки	Описание
16#6111	D+1	Недействительный удаленный адрес электронной почты

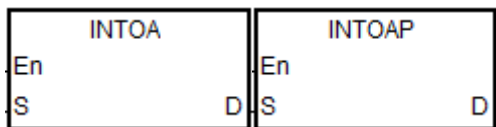
API	Код инструкции			Операнды				Функция			
2206		INTOA	P	<b>S, D</b>				Преобразование IP-адреса целочисленного типа в IP-адрес строкового типа			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S</b>								●	●							
<b>D</b>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●			●	●							
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

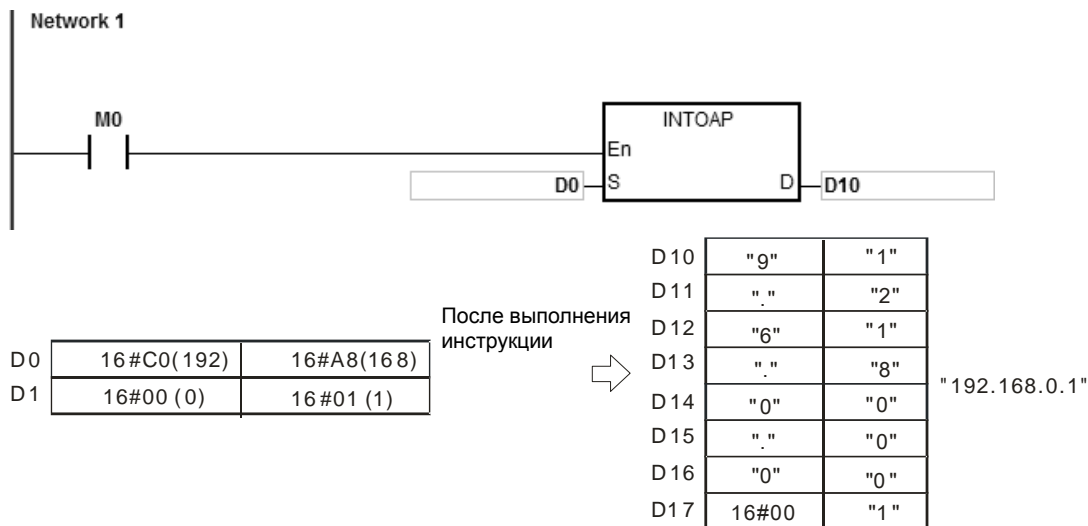


**S** : Источник данных  
**D** : Результат преобразования

Описание:

1. IP адрес целого типа в **S** и **S+1** преобразуется в IP адрес строкового типа, результат сохраняется в **D**.
2. Операнд **D** занимает 8 последовательных регистров.

Пример:



Примечание:

1. Если операнд **S** задается в ISPSOft, тип данных будет ARRAY [2] для WORD/INT.
2. Если операнд **D** в ISPSOft, задается в ISPSOft, тип данных будет ARRAY [8] для WORD/INT.

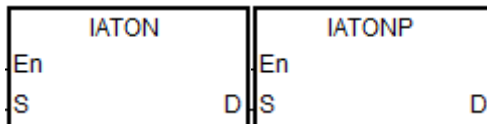
API	Код инструкции			Операнды				Функция					
2207		IATON	P	<b>S, D</b>				Преобразование IP-адреса строкового типа в IP-адрес целочисленного типа					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S</b>								●	●						○	
<b>D</b>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●			●	●							●
<b>D</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

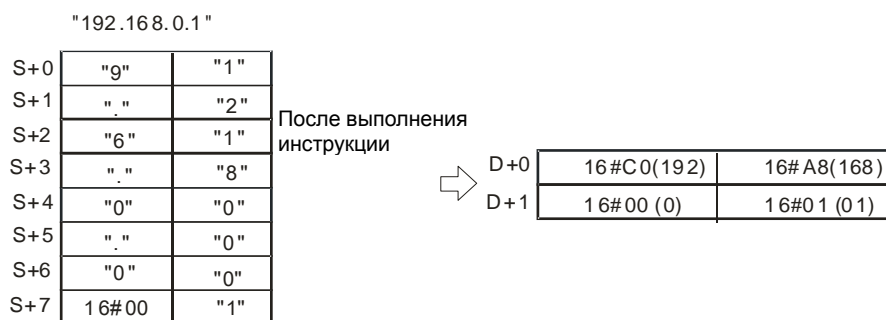


**S** : Источник данных

**D** : Результат преобразования

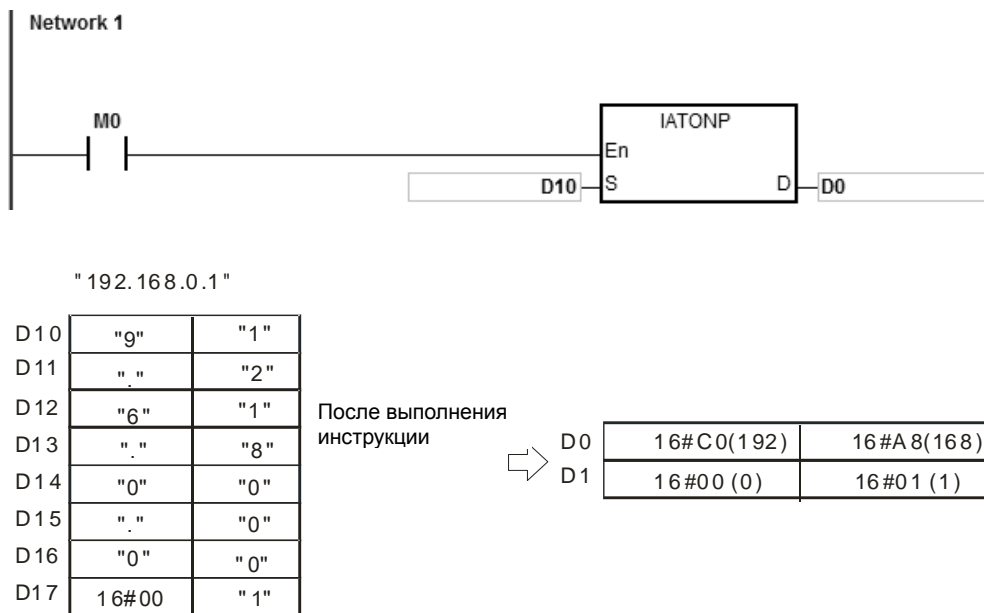
Описание:

- IP адрес строкового типа в **S** преобразуется в IP адрес целочисленного типа, результат сохраняется в **D** и **D+1**.
- Операнд **S** занимает 8 последовательных регистров.

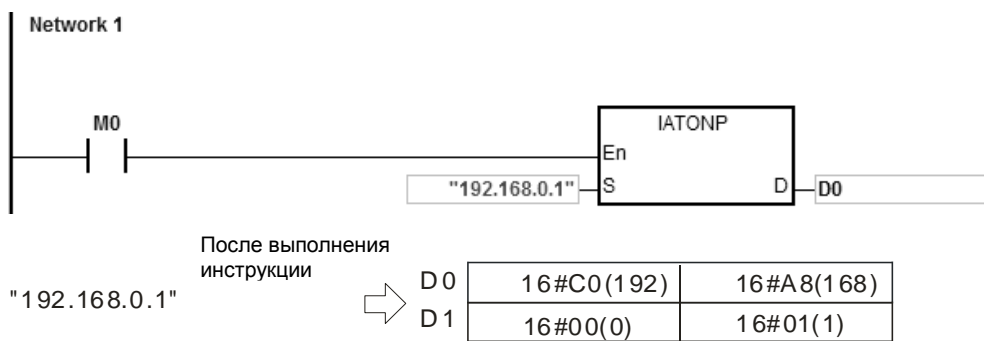


- В каждом разделе IP-адреса строкового типа в **S** присутствуют 1~3 символа. Эти разделы разделяются символом «.» (16#2E). Например, можно вводить «192.168.0.1» вместо «192.168.000.001»,
- Значение, преобразованное из символов в каждом разделе IP-адреса строкового типа в **S**, должно находиться в диапазоне от 0 до 255.

**Пример 1:**



**Пример 2:**



**Примечание:**

1. Если строковое значение в **S** не заканчивается символом 16#00, включен SM0 и в SR0 записывается код ошибки 16#200E.
2. В значении **S**, исключая код десятичной точки, двоичные коды должны находиться в диапазоне между 16#30 и 16#39. Если двоичные коды не находятся в этом диапазоне, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
3. Если число десятичных точек "." в **S** не равно трем, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
4. Если преобразованное значение в любом из секторов **S** не находится в пределах диапазона 0~255, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.

5. Число знаков в любом секторе IP адреса в **S** должно находиться в диапазоне 1~3. Если число знаков в секторе превышает 3, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
6. Если операнд **S** задается в ПО ISPSOft, тип данных должен быть ARRAY [8] для WORD/INT.
7. Если операнд **D** задается в ПО ISPSOft, тип данных должен быть ARRAY [2] для WORD/INT.

API	Код инструкции			Операнды							Функция					
2208		EIPRW		$S_1 \sim S_7, n, S, D_1, D_2$							Чтение и запись данных Ethernet/IP					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$								●	●							
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		
$S_4$								●	●				○	○		
$S_5$								●	●				○	○		
$S_6$	●	●	●	●												
$S_7$								●	●				○	○		
n								●	●				○	○		
S								●								
$D_1$								●								
$D_2$								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1 \sim D_2$	См. Описание												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символ:

EIPRW	
En	
S1	D1
S2	D2
S3	
S4	
S5	
S6	
S7	
n	
S	

$S_1 \sim S_7, n, S$  : См. Описание

$D_1 \sim D_2$  : См. Описание

**Описание:**

1. Наименование и описание операндов  $S_1 \sim S_7$ ,  $n$ ,  $S$  и  $D_1 \sim D_2$  представлены в таблице ниже.

Операнд	Наименование	Описание	Тип данных	Примечание
$S_1$	IP адрес	Первые два раздела IP-адреса занимают первое слово, а остальные два раздела IP-адреса занимают второе слово. Например: если IP адрес – 192.168.1.5, $S_1=16\#C0A8$ и $S_1+1=16\#0105$ .	WORD[2]	Занимает 2 последовательных слова
$S_2$	Режим подключения	0: UCMM	WORD	
$S_3$	Функциональный (Служебный) код	Диапазон: $16\#0000 \sim 00FF$ . Если функциональный код превышает диапазон, инструкция не выполняется	WORD	
$S_4$	Class ID	См. протокол Ethernet/IP	WORD	
$S_5$	Instance ID	См. протокол Ethernet/IP	WORD	
$S_6$	Переключатель атрибута ID	ВКЛ: Разрешено; ВЫКЛ: Запрещено	BOOL	
$S_7$	Атрибут (Attribute) ID	См. протокол Ethernet/IP	WORD	
$n$	Длина считываемых и записываемых данных	Размер данных для чтения и записи. Ед. изм.: байт; Макс. длина: 200 байтов.	WORD	
$S$	Регистр, связанный с данными чтения / записи	Регистр, из которого поступают отправленные данные, или регистр, в котором хранятся полученные данные	WORD[n]	
$D_1$	Состояние коммуникации	0: Коммуникация не запускается 1: Выполняется 2: Коммуникация завершена без ошибок 3: Произошла ошибка коммуникации.	WORD	



Операнд	Наименование	Описание	Тип данных	Примечание
		4: Ошибка в настройке параметра		
<b>D<sub>2</sub></b>	Код ошибки	Основной и расширенный коды ошибки	WORD[2]	Занимает 2 последовательных слова

2. Когда инструкция запускается первый раз, она показывает, что отправлена команда коммуникации. Если количество подключенных ведомых устройств достигает верхнего предела, значение состояния коммуникации в **D<sub>1</sub>** равно 0, и это означает, что выполнение коммуникации не происходит.
3. Если значения параметров выходят за допустимые пределы, инструкция не выполняется, а значение в **D<sub>1</sub>** равно 4.

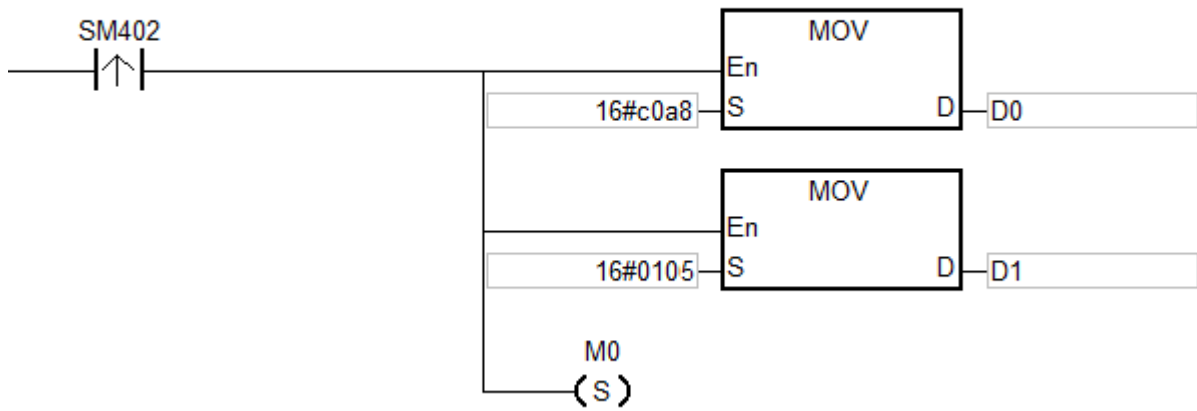
**Пример 1:**

Считывается код удаленного устройства (по адресу 192.168.1.5), данные считывания сохраняются в D100.

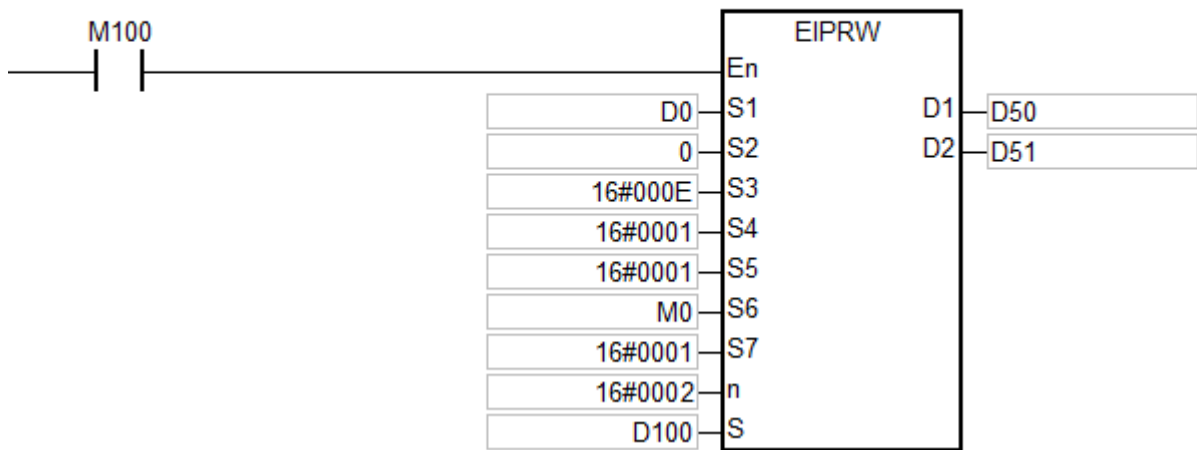
Параметры объекта EtherNet / IP установлены ниже.

- (1) Class ID = 1
- (2) Instance ID = 1
- (3) Attribute ID = 1

Network 1



Network 2

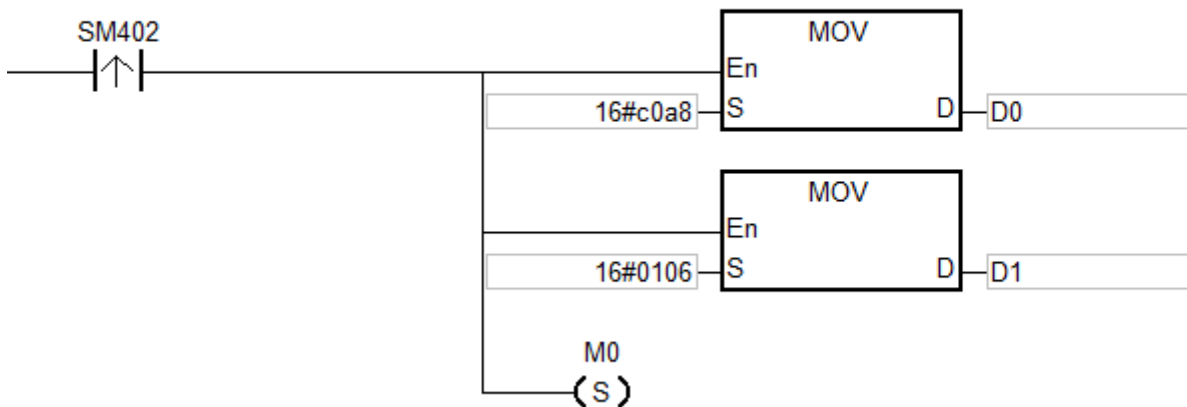


**Пример 2:**

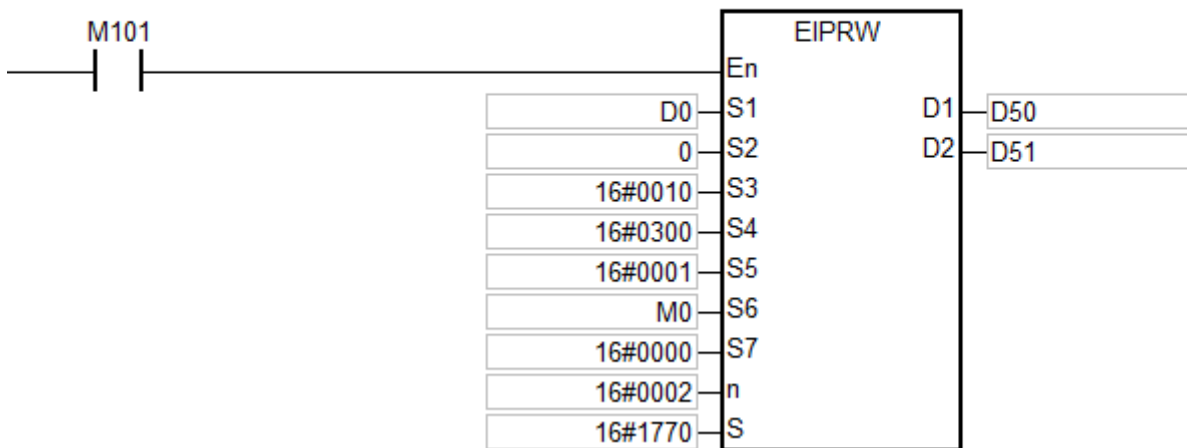
Максимальная частота (01-00) ПЧ (по адресу 192.168.1.6) равна 60,00 Гц. Параметры объекта EtherNet/IP представлены ниже.

- (1) Class ID = 16#0300
- (2) Instance ID = 16#0001
- (3) Attribute ID = 16#0000

**Network 1**



**Network 2**



**Примечание:**

- 1. Если операнд **S**<sub>1</sub> задается в ПО ISPSOft, тип данных должен быть ARRAY [2] для WORD.
- 2. Если операнд **S** задается в ПО ISPSOft, тип данных должен быть ARRAY [n] для WORD, где **n** – размер считанных / записанных данных.

3. Если операнд  $D_2$  задается в ПО ISPSOft, тип данных должен быть ARRAY [2] для WORD.
4. Ошибки, возникающие при выполнении инструкции, показаны ниже.

Код ошибки	Флаг ошибки	Описание
16#2003	SM0 / $D_1$	1. Значения $S_1$ , $S_2$ , $S_3$ или $S_6$ выходят за допустимые пределы 2. $S + n$ находится за допустимыми пределами для объектов
16#200B	SM0	Значение $n$ выходит за допустимые пределы
16#600D	SM1100	Сеть Ethernet не подключена
16#6301	$D_1$	Соединение с удаленным устройством нарушено
16#6302	$D_1$	Превышение времени ожидания ответа удаленного устройства
16#6303	$D_1$	Недопустимый IP-адрес
16#6304	$D_1$	Ошибка в сервисном коде команды ответа
16#6305	$D_1$	Ошибка в длине команды ответа
16#6306	$D_1$	Конфликт оборудования связи

5. Коды состояний в  $D_2$  [0] представлены ниже.

Код состояния	Описание	Методы устранения ошибок
16#00	Соединение выполнено	
16#01	Ошибка соединения	Проверьте корректность файла EDS Slave-устройства
16#02	Устройства, подключенные к сети, недоступны	1. Проверьте, не превышает ли количество устройств, подключенных к Master, допустимое значение. 2. Проверьте, не превышает ли количество устройств, подключенных к Slave допустимое.
16#03	Ошибка настройки параметра	Проверьте корректность данных для чтения/записи в $S$
16#04	Ошибка пути	Проверьте настройки Class ID ( $S_4$ ), Instance ID ( $S_5$ ) и Attribute ID ( $S_7$ )
16#05	Путь не существует	Проверьте настройки Class ID ( $S_4$ ), Instance ID ( $S_5$ ) и Attribute ID ( $S_7$ )
16#07	Обрыв соединения	1. Проверьте, правильно ли подключен порт Ethernet Slave устройства 2. Проверьте правильность настройки поддерживающего

Код состояния	Описание	Методы устранения ошибок
		таймера для Slave устройства
16#08	Сервисный код не поддерживается	Проверьте функциональный код ( <b>S<sub>3</sub></b> )
16#09	Неверное значение атрибута	Проверьте соответствие настроек регистров данных для чтения/записи и содержимого
16#0A	Ошибка в списке атрибутов	Проверьте, позволяет ли атрибут объекта Slave устройства выполнять функции Get_Attribute_List и Set_Attribute_List
16#0B	Конфликт передачи	Проверьте повторяемость настроек
16#0C	Конфликт состояний объектов	Проверьте соединение IO
16#0D	Объект не существует	Проверьте поддержку Slave устройством заданного объекта
16#0E	Атрибут не доступен для записи	Проверьте, поддерживает ли атрибут объекта функцию записи
16#0F	Нет прав на выполнение служебного кода	Проверьте, разрешено ли Slave устройству выполнять данный служебный код
16#10	Устройство не может выполнить действие в настоящий момент	Проверьте соединение IO
16#11	Размер данных в ответе слишком велик	Проверьте длину данных на соответствие атрибуту объекта (100 слов)
16#12	При доступе к тегу возникает ошибка последовательности доступа к данным	Проверьте правильность длины и типа данных
16#13	Переданные данные слишком короткие	Проверьте длину ( <b>n</b> ) данных для чтения/записи
16#14	Значение атрибута не поддерживается	Проверьте правильность установленного переключателя атрибута ID ( <b>S<sub>6</sub></b> ) и значения атрибута ID ( <b>S<sub>7</sub></b> )
16#15	Слишком большой объем переданных данных	Проверьте длину ( <b>n</b> ) данных для чтения/записи

Код состояния	Описание	Методы устранения ошибок
16#16	Объект не существует	Проверьте установленный Class ID ( <b>S<sub>4</sub></b> )
16#17	После доступа к тегам при доступе к данным возникает ошибка	<ol style="list-style-type: none"> <li>1. Проверьте соединение сети Ethernet</li> <li>2. Проверьте потери пакетов данных при коммуникации Ethernet</li> </ol>
16#18	Значение атрибута не сохраняется	Проверьте, не возникает ли ошибка состояния Slave устройства.
16#19	Ошибка сохранения значения атрибута	Проверьте, произошла ли аппаратная ошибка Slave устройства
16#1A	Ошибка маршрутизатора. Длина пакета запроса превышает допустимый предел	Проверьте, превышает ли длина данных ( <b>n</b> ) для чтения/записи предел значения для маршрутизатора
16#1B	Ошибка маршрутизатора. Длина пакета ответа превышает допустимый предел	Проверьте, превышает ли длина данных ( <b>n</b> ) для чтения/записи предел значения для маршрутизатора
16#1F	Пользовательская ошибка доступа к объекту	См. описание ошибок Slave устройства
16#20	Недопустимое значение параметра	Проверьте правильность прочитанного/записанного значения в <b>S</b> .

API	Код инструкции			Операнды							Функция						
2209		SCONF	P	$S_1 \sim S_{10}$							Задание параметров TCP/UDP сокета						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$								●	●				○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		
$S_4$								●	●				○	○		
$S_5$								●	●				○	○		
$S_6$								●	●				○	○		
$S_7$								●	●				○	○		
$S_8$								●	●				○	○		
$S_9$								●	●				○	○		
$S_{10}$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1 \sim S_{10}$	См. Описание												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

SCONF	SCONFP
En	En
S1	S1
S2	S2
S3	S3
S4	S4
S5	S5
S6	S6
S7	S7
S8	S8
S9	S9
S10	S10

$S_1 \sim S_{10}$  : См. Описание

Описание:

1. Наименование и описание операндов  $S_1 \sim S_{10}$  представлены в таблице ниже.

Операнд	Наименование	Описание	Тип данных	Примечание
<b>S<sub>1</sub></b>	Выбор TCP/UDP	0: TCP, 1: UDP	WORD	
<b>S<sub>2</sub></b>	Номер сокета	Диапазон: 1–4	WORD	
<b>S<sub>3</sub></b>	Удаленный IP-адрес	Первые два раздела IP-адреса занимают первое слово, а остальные два раздела IP-адреса занимают второе слово. Например: если IP-адрес 192.168.1.5, то <b>S<sub>3</sub></b> =16#C0A8 и <b>S<sub>3</sub>+1</b> =16#0105.	WORD[2]	Занимает 2 последовательных слова
<b>S<sub>4</sub></b>	Удаленный порт	Диапазон: 0–65535; 0 указывает на любой порт	WORD	
<b>S<sub>5</sub></b>	Локальный порт	Диапазон: 0–65535; 0 указывает на любой порт	WORD	
<b>S<sub>6</sub></b>	Регистр, откуда поступают данные	Указывает номер операнда <b>D</b> . Диапазон: 0–29999. Например: заданное значение 100 означает, что байты данных из <b>S<sub>7</sub></b> отправляются из регистров, начиная с D100 (от младших к старшим байтам)	WORD	
<b>S<sub>7</sub></b>	Размер переданных данных	Максимум: 200 байтов.	WORD	
<b>S<sub>8</sub></b>	Регистр, в котором сохраняются полученные данные	Указывает номер операнда <b>D</b> . Диапазон: 0–29999. Например: заданное значение 200 означает, что байты принятых данных в <b>S<sub>9</sub></b> сохраняются в регистрах, начиная с D200 (от младших к старшим байтам)	WORD	
<b>S<sub>9</sub></b>	Размер принятых данных	Максимум: 200 байтов.	WORD	
<b>S<sub>10</sub></b>	Время соединения	Диапазон: 1–30000, ед. изм.: секунда	WORD	Применимо только для режима TCP

2. Рекомендуется применять импульсный вариант данной инструкции.



3. Параметры сокета по умолчанию задаются в HWCONFIG. Инструкция потребует для редактирования только когда необходимо изменять случайным образом значения параметров во время коммуникации,
4. Если параметры заданы, когда номер сокета находится в процессе связи, значения настройки не будут действовать до завершения текущего сеанса связи. Настройка должна быть выполнена после проверки того, что сокет не используется.
5. Если значение хотя бы одного параметра выходит за допустимые пределы, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнды								Функция					
2210		MCONF	P	<b>S<sub>1</sub>~S<sub>11</sub></b>								Чтение/запись данных Modbus TCP					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●							
S <sub>3</sub>								●	●				○	○		
S <sub>4</sub>								●	●				○	○		
S <sub>5</sub>								●	●				○	○		
S <sub>6</sub>								●	●				○	○		
S <sub>7</sub>			●					●								
S <sub>8</sub>								●	●				○	○		
S <sub>9</sub>								●	●				○	○		
S <sub>10</sub>								●	●				○	○		
S <sub>11</sub>			●					●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub> ~S <sub>11</sub>	См. Описание												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

MCONF	MCONF
En	En
S1	S1
S2	S2
S3	S3
S4	S4
S5	S5
S6	S6
S7	S7
S8	S8
S9	S9
S10	S10
S11	S11

S<sub>1</sub>~S<sub>11</sub> : См. Описание

Описание:

1. Наименование и описание операндов S<sub>1</sub>~S<sub>11</sub> представлены в таблице ниже.

Операнд	Наименование	Описание	Тип данных	Примечание
<b>S<sub>1</sub></b>	Номер в таблице обмена данными	Таблица обмена данными MODBUS TCP, диапазон: 1~32	WORD	
<b>S<sub>2</sub></b>	Удаленный IP- адрес	Первые два раздела IP-адреса занимают первое слово, а остальные два раздела IP-адреса занимают второе слово. Например: если IP-адрес 192.168.1.5, то <b>S<sub>3</sub></b> =16#C0A8 и <b>S<sub>3</sub>+1</b> =16#0105.	WORD[2]	Занимает 2 последовательных слова
<b>S<sub>3</sub></b>	Адрес станции удаленного Slave устройства	Диапазон: 0 ~ 255. Если значение настройки превышает диапазон, значение в младшем байте будет задаваться автоматически	WORD	
<b>S<sub>4</sub></b>	Функциональный код чтения	Диапазон: 16#01, 02, 03, 04 и 17. Если значение функционального кода выходит за пределы диапазона, модификация полученных параметров, включая адрес и длину данных, не выполняется	WORD	
<b>S<sub>5</sub></b>	Чтение удаленного адреса при коммуникации	16#0000~16#FFFF	WORD	
<b>S<sub>6</sub></b>	Длина считываемых данных	Функциональный код чтения битовых объектов поддерживает максимально 256 битов; Функциональный код чтения словных объектов поддерживает максимально 100 слов	WORD	
<b>S<sub>7</sub></b>	Локальный регистр, в котором хранятся полученные данные	Функциональный код чтения битовых объектов поддерживает только объекты M; Функциональный код чтения	BOOL WORD	

Операнд	Наименование	Описание	Тип данных	Примечание
		словных объектов поддерживает только объекты D		
<b>S<sub>8</sub></b>	Функциональный код записи	Диапазон: 16#05, 06, 0F и 10. Если значение функционального кода выходит за пределы диапазона, модификация полученных параметров, включая адрес и длину данных, не выполняется	WORD	
<b>S<sub>9</sub></b>	Запись удаленного адреса при коммуникации	Диапазон: 16#0000~16#FFFF	WORD	
<b>S<sub>10</sub></b>	Длина записанных данных	Функциональный код записи битовых объектов поддерживает максимально 256 битов; Функциональный код записи словных объектов поддерживает максимально 100 слов	WORD	
<b>S<sub>11</sub></b>	Локальный регистр, в котором записаны данные от удаленного устройства	Функциональный код записи битовых объектов поддерживает только объекты M; Функциональный код записи словных объектов поддерживает только объекты D	BOOL WORD	

2. Рекомендуется применять импульсный вариант данной инструкции.
3. Функциональные коды коммуникации по Modbus в **S<sub>4</sub>** и **S<sub>8</sub>** представлены ниже.

Когда ПЛК серии AS считывает данные из нескольких битовых устройств (кроме дискретных входов), переданный функциональный код равен 1 (16#01).

Когда ПЛК серии AS считывает данные из нескольких битовых устройств (только дискретные входы), переданный функциональный код равен 2 (16#02).

Когда ПЛК серии AS считывает данные из нескольких словных устройств (кроме входных регистров), переданный функциональный код равен 3 (16#03).

Когда ПЛК серии AS считывает данные из нескольких словных устройств (только входные регистры), переданный функциональный код равен 4 (16#04).

Когда ПЛК серии AS записывает данные в одно битовое устройство, переданный функциональный код равен 5 (16#05).

Когда ПЛК серии AS записывает данные в одно словное устройство, переданный функциональный код равен 6 (16#06).

Когда ПЛК серии AS записывает состояние в несколько битовых устройств, переданный функциональный код равен 15 (16#0F).

Когда ПЛК серии AS записывает данные в несколько битовых устройств, переданный функциональный код равен 16 (16#10).

Когда ПЛК серии AS записывает данные в несколько словных устройств, переданный функциональный код равен 23 (16#17).

В настоящее время ПЛК серии AS поддерживает только функциональные коды, представленные выше.

4. Значения параметров, заданные в инструкции, действительны только во время работы ПЛК. Когда ПЛК перегружается после выключения питания, значения настроек в таблице обмена данными будут приниматься по умолчанию. Если во время обмена данными необходимо редактировать некоторые значения параметров, данная инструкция используется для изменения значений параметров.
5. Если в сообщении находится указанный номер, вновь установленные значения параметров будут активированы при следующем сеансе коммуникации по завершении текущего сеанса.
6. Если значение хотя бы одного параметра выходит за допустимые пределы, инструкция не выполняется; включается SM0 и в SR0 записывается код ошибки 16#200B.
7. Если программа ПЛК используется для управления, сначала включите режим «Program control» в таблице обмена данными и после этого можно использовать SM из следующей таблицы.

Номер SM	Атрибут	Описание параметра обмена данными Ethernet
SM1167	R/W	Флаг общего включения обмена данными
SM1168 ~ SM1199	R/W	Флаг включения обмена данными 1~32
SM1200 ~ SM1231	R	Флаг успешного соединения для обмена данными 1~32
SM1232 ~ SM1263	R	Флаг ошибки подключения для обмена данными 1~32

Если при получении данных ошибки не возникло, включается флаг успешного соединения. При возникновении ошибки, включается флаг ошибки. Для одного соединения оба флага одновременно быть включены не могут.

8. Специальные регистры (только для чтения) SR используются следующим образом.

Номер SR	Описание
SR1120 ~ SR1151	Указывают соответственно фактическое время связи соединений 1~32.
SR1152 ~ SR1183	Указывают соответственно коды ошибок связи соединений 1~32.

Коды ошибок в SR1152 ~ SR1183 представлены ниже.

Код ошибки	Описание
16#00XX	Ошибка ответа удаленного устройства
16#F000	Сеть Ethernet не подключена
16#F001	Превышение времени ожидания ответа удаленного устройства
16#F003	Превышение времени ожидания соединения TCP
16#F007	Ошибка команды ответа
16#F009	Соединение с удаленным устройством отключено

API	Код инструкции			Операнды							Функция						
2211		EMCONF1	P	<b>Server ~ Passw</b>							Задание параметров сервера Email						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
Server								●	●							
Port								●	●				○	○		
LMail								●	●						○	
Sub								●	●						○	
Verify		●	●	●				●								
User								●	●						○	
Passw								●	●						○	

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Server ~ Passw	См. Описание												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

EMCONF1	EMCONF1P
En	En
Server	Server
Port	Port
LMail	LMail
Sub	Sub
Verify	Verify
User	User
Passw	Passw

**Server ~ Passw** : См. Описание

Описание:

1. Операнды, работающие в данной инструкции представлены в таблице ниже:

Операнд	Наименование	Описание	Тип данных	Примечание
<b>Server</b>	Сервер Email	<b>Server</b> +0: Длина данных, ед.: байты (Диапазон: 0~63, 0: нет изменений) <b>Server</b> +1~ <b>Server</b> +32: Содержимое данных (данные занимают младшие 8 бит)	WORD[N]	Обратитесь к <b>Server</b> +0 для присвоения адреса. Можно использовать 33 последовательных слова
<b>Port</b>	Порт связи	Диапазон: 1~65535, 0: нет изменений.	WORD	
<b>LMail</b>	Локальный адрес Email	<b>LMail</b> +0: Длина данных, ед.: байты (Диапазон: 0~31, 0: нет изменений) <b>LMail</b> +1~ <b>LMail</b> +16: Содержимое данных (данные занимают младшие 8 бит)	WORD[N]	Обратитесь к <b>LMail</b> +0 для присвоения адреса. Можно использовать 17 последовательных слов

<b>Sub</b>	Субъект Email	<b>Sub +0:</b> Длина данных, ед.: байты (Диапазон: 0~31, 0: нет изменений) <b>Sub +1~ Sub +16:</b> Содержимое данных (данные занимают младшие 8 бит)	WORD[N]	Обратитесь к <b>Sub +0</b> для присвоения адреса. Можно использовать 17 последовательных слов
<b>Verify</b>	Включение / отключение проверки	0: Проверка отключена, 1: Проверка включена.	BOOL	
<b>User</b>	Имя пользователя	<b>User +0:</b> Длина данных, ед.: байты (Диапазон: 0~31, 0: нет изменений) <b>User +1~ User +16:</b> Содержимое данных (данные занимают младшие 8 бит)	WORD[N]	Обратитесь к <b>User +0</b> для присвоения адреса. Можно использовать 17 последовательных слов
<b>Passw</b>	Пароль	<b>Passw +0:</b> Длина данных, ед.: байты (Диапазон: 0~20, 0: нет изменений) <b>Passw +1~ Passw +10:</b> Содержимое данных (данные занимают младшие 8 бит)	WORD[N]	Обратитесь к <b>Passw +0</b> для присвоения адреса. Можно использовать 11 последовательных слов



API	Код инструкции			Операнды							Функция					
2212		EMCONF2	P	Index, Mail							Задание адреса Email					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Index								●	●							
Mail								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Index, Mail	См. Описание												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:

EMCONF2	EMCONF2P
En	En
Index	Index
Mail	Mail

Index, Mail : См. Описание

Описание:

1. Операнды, работающие в данной инструкции представлены в таблице ниже:

Операнд	Наименование	Описание	Тип данных	Примечание
Index	Номер адреса Email	<b>Index</b> : Номер изменяемого адреса Email (Диапазон: 0~8, 0: нет изменяемых адресов Email)	WORD	
Mail	Адрес Email	<b>Mail</b> +0: Длина данных, ед.: байты (Диапазон: 0~63, 0: нет изменений) <b>Mail</b> +1~ <b>Mail</b> +32: Содержимое данных (данные занимают младшие 8 бит)	WORD[N]	Обратитесь к <b>Mail</b> +0 для присвоения адреса. Можно использовать 33 последовательных слова

## 6.23 Инструкции для карты памяти

### 6.23.1 Описание инструкций для карты памяти

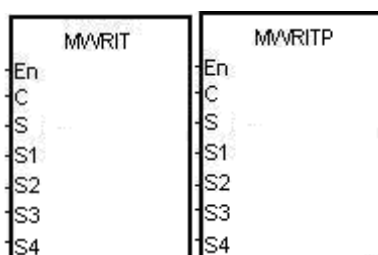
API	Код инструкции			Операнды	Функция
2300		MWRIT	P	C · S · S <sub>1</sub> · S <sub>2</sub> · S <sub>3</sub> · S <sub>4</sub>	Запись данных с ПЛК на карту памяти

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
C								●	●				○	○		
S								●	●							
S <sub>1</sub>								●	●				○	○		
S <sub>2</sub>								●	●				○	○		
S <sub>3</sub>								●	●							
S <sub>4</sub>								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
C		●											
S		●											
S <sub>1</sub>			●										
S <sub>2</sub>		●											
S <sub>3</sub>													●
S <sub>4</sub>			●										

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символ:



- C** : Параметр управления
- S** : Источник данных
- S<sub>1</sub>** : Длина данных
- S<sub>2</sub>** : Линейное продвижение
- S<sub>3</sub>** : Имя файла
- S<sub>4</sub>** : Адрес данных в файле

Описание:

1. Описание операндов:

- **C**: параметр управления



Пункт	Код	Описание
Формат файла	0	Двоичное значение
		Значение по умолчанию
		Файл с расширением .dmd
		Единицей значения является слово
	1	Значения разделяются запятой
		Единицей значения является слово
		Файл с расширением .cvs
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	2	Значения разделяются запятой
		Единицей значения является двойное слово
		Файл с расширением .cvs
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	3	Значения разделяются вкладкой
		Единицей значения является слово
		Файл с расширением .txt
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	4	Значения разделяются вкладкой
		Единицей значения является двойное слово
		Файл с расширением .txt
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
5	Значения не разделяются никаким знаком	
	Единицей значения является слово	
	Файл с расширением .txt	
	Принимаются коды ASCII	
	Сохраняемое значение является шестнадцатеричным	
6	Значения не разделяются никаким знаком	

6

Пункт	Код	Описание
		Единицей значения является двойное слово
		Файл с расширением .txt
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
Функции	0	Добавление
		Данные, записанные на карту памяти, добавляются после последнего значения в файле.
		Значение по умолчанию
		Если файл не существует, он создается автоматически
1	Перезапись	
	Данные, записанные на карту памяти, заменяются значением в файле, начиная с указанного в <b>S<sub>4</sub></b> .	
	Если файл не существует, он создается автоматически	
Зарезервирован	-	Значение бит8-бит15 равно 0

- **S**: Источник данных
- **S<sub>1</sub>**: Длина данных, которые записываются в файл

Если значение в **S<sub>1</sub>** равно 0, данные в файл не записываются.

Пункт	Описание
Единицы значения	Если формат файла 0, 1, 3 или 5, единицей значения является слово. Если формат файла 2, 4 или 6, единицей значения является двойное слово
Единицы параметров	Двойное слово
Длина данных	Объекты, в которых хранятся данные, не могут выходить за пределы допустимого диапазона, а размер данных, записанных в файл, не может превышать четырех гигабайт (См. Глава 2)

- **S<sub>2</sub>**: Линейное продвижение  
Значение в **S<sub>2</sub>** должно находиться в диапазоне от 0 до 256.
- ПЛК серии AS с версией прошивки 1.04.00 или более поздней версии поддерживает использование строковых переменных (не более 9 символов) в имени файла **S<sub>3</sub>**.
- **S<sub>3</sub>-S<sub>3+4</sub>**: **S<sub>3</sub>** занимают пять объектов. Чаще всего имя файла составляют девять символов, включая 16#00. Если строковое значение не заканчивается символом 16#00, возникает ошибка. Если конечный

символ прочитан, чтение символов прекращается и проверяется корректность имени файла. Символами, которые могут использоваться для создания имени файла, являются A ~ Z, a ~ z и 0 ~ 9. Расширение имени файла зависит от формата файла. Созданный файл находится в папке по умолчанию (создается автоматически). Если имя файла «Test1», символы записываются в объекты следующим образом.



- Путь к папке по умолчанию (на примере ЦПУ ПЛК AS300):

Модуль ЦПУ AS300	Путь к папке по умолчанию
AS332T-A	PLC CARD\AS300\UserProg
AS332P-A	
AS324MT-A	

- **S<sub>4</sub>**: Значение в файле, который перезаписывается, указывается в **S<sub>4</sub>**.

Пункт	Описание
Единицы значения	Если формат файла 0, 1, 3 или 5, единицей значения является слово. Если формат файла 2, 4 или 6, единицей значения является двойное слово
Единицы параметров	Двойное слово
Применение	Если значение функции равно 0, <b>S<sub>4</sub></b> не используется
	Если значение функции равно 1, данные, записанные на карту памяти, заменяют значения в файле, начиная со значения, указанного в <b>S<sub>4</sub></b>
	Значение в <b>S<sub>4</sub></b> должно указывать значение в файле;
	Если значение в <b>S<sub>4</sub></b> равно 0, перезаписывается первое значение в файле

2. Соответствующие флаги:

Флаг	Описание
SM450	Если карта памяти вставлена в модуль ЦПУ, флаг включается
SM451	Переключатель защиты от записи на карте памяти:
	Флаг включен: Карта памяти защищена от записи; Флаг выключен: Карта памяти не защищена от записи

Флаг	Описание
SM452	Данные записываются из ПЛК на карту памяти или данные считываются с карты памяти в ПЛК: флаг включен
SM453	Если во время работы карты памяти произошла ошибка, флаг будет включен. Если флаг включен, пользователь должен вручную его отключить. Код ошибки хранится в SR453.

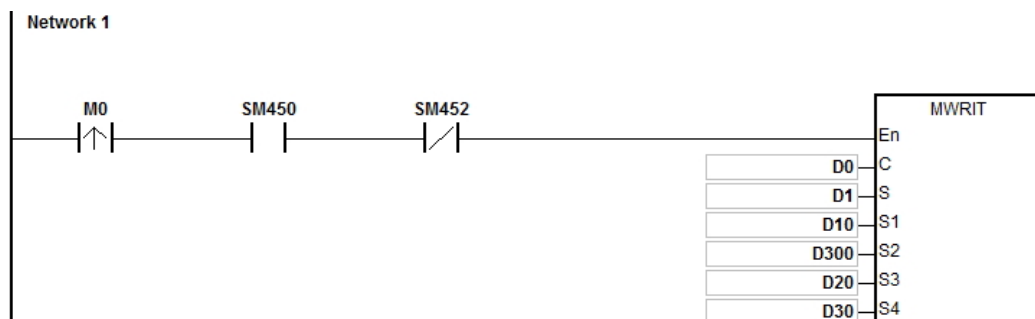
3. Коды ошибки (SR453):

Код ошибки	Описание
16#005E	Ошибка инициализации карты памяти
16#005F	Некорректный путь или файл не существует
16#0060	Папка по умолчанию не может быть создана
16#0061	Недостаточный объем памяти
16#0062	Карта памяти защищена от записи
16#0063	Произошла ошибка при записи данных в файл
16#0064	Данные не могут быть считаны с карты памяти
16#0065	Файл является доступным только для чтения

4. Если формат файла, в который записаны данные, равен 0, формат файла, из которого считываются данные, также равен 0. В противном случае данные не могут быть прочитаны и включается SM453. Аналогично по другим форматам файлов.

**Пример:**

SM450 включается при помещении карты памяти в модуль ЦПУ; SM452 включается при выполнении инструкции MWRIT и выключается после выполнения данной инструкции. Импульсная инструкция MWRITP не может использоваться непрерывно. Если выполнять импульсную инструкцию для непрерывной записи данных на карту памяти, она может допустить превышение ограничения на запись и повредить карту памяти.



Операнд	Заданное значение	Описание
D0	16#0011	Файл, содержащий записанные данные

Операнд	Заданное значение	Описание
		Формат файла: Значения разделяются запятой Единицей значения является слово Файл с расширением .cvs. Принимаются коды ASCII
D1	-	Данные, записанные в файл
D10, D11	16#00000030	Размер данных, записанных в файл, составляет 48 слов
D300	16#000A	В каждую строку записываются десять значений
D20	D20=16#6554 D21=16#7473 D22=16#0031	Имя файла - «Test1»
D30 · D31	16#00000000	Данные, записанные на карту памяти, заменяют значения в файле, начиная с первого значения

**Примечание:**

1. Если значение в **C** выходит за допустимые пределы возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение в **S<sub>1</sub>** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
3. Если значение в **S<sub>2</sub>** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
4. Если значение в **S<sub>3</sub>** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнды								Функция					
2301		MREAD	P	<b>C · S · S<sub>1</sub> · S<sub>2</sub> · S<sub>3</sub> · D</b>								Считывание данных с карты памяти в ПЛК					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>C</b>								●	●				○	○		
<b>S</b>								●	●							
<b>S<sub>1</sub></b>								●	●				○	○		
<b>S<sub>2</sub></b>								●	●				○	○		
<b>S<sub>3</sub></b>								●	●				○	○		
<b>D</b>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>C</b>		●											
<b>S</b>													●
<b>S<sub>1</sub></b>			●										
<b>S<sub>2</sub></b>		●											
<b>S<sub>3</sub></b>			●										
<b>D</b>		●											

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**

**C** : Параметр управления

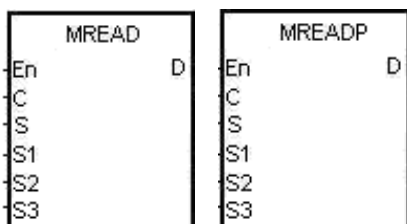
**S** : Имя файла

**S<sub>1</sub>** : Адрес данных в файле

**S<sub>2</sub>** : Зарезервирован

**S<sub>3</sub>** : Длина данных

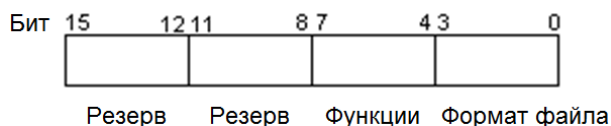
**D** : Операнд сохранения данных



**Описание:**

1. Описание операндов:

- **C**: Параметр управления





Пункт	Код	Описание
Формат файла	0	Двоичное значение
		Значение по умолчанию
		Файл с расширением .dmd
		Единицей значения является слово
	1	Значения разделяются запятой
		Единицей значения является слово
		Файл с расширением .cvs
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	2	Значения разделяются запятой
		Единицей значения является двойное слово
		Файл с расширением .cvs
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	3	Значения разделяются вкладкой
		Единицей значения является слово
		Файл с расширением .txt
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	4	Значения разделяются вкладкой
		Единицей значения является двойное слово
		Файл с расширением .txt
		Принимаются коды ASCII
		Сохраняемое значение является шестнадцатеричным
	5	Значения не разделяются никаким знаком
		Единицей значения является слово
		Файл с расширением .txt
		Принимаются коды ASCII
Сохраняемое значение является шестнадцатеричным		
6	Значения не разделяются никаким знаком	
	Единицей значения является двойное слово	
	Файл с расширением .txt	
	Принимаются коды ASCII	
	Сохраняемое значение является шестнадцатеричным	

Пункт	Код	Описание
Функция	0	В файле считываются значения, начиная со значения в <b>S<sub>1</sub></b> Значение по умолчанию равно 0
	1	Число значений сохраняется в <b>D</b> и <b>D+1</b> . Если формат файла 0, 1, 3 или 5, единицей значения является слово. Если формат файла 2, 4 или 6, единицей значения является двойное слово
Зарезервирован	-	Значение бит8~бит15 равно 0

- ПЛК серии AS с версией прошивки 1.04.00 или более поздней версии поддерживает использование строковых переменных (не более 9 символов) в имени файла **S**.
- **S~S+4**: **S** занимают пять объектов. Чаще всего имя файла составляют девять символов, включая 16#00. Если строковое значение не заканчивается символом 16#00, возникает ошибка. Если конечный символ прочитан, чтение символов прекращается и проверяется корректность имени файла. Символами, которые могут использоваться для создания имени файла, являются A ~ Z, a ~ z и 0 ~ 9. Расширение имени файла зависит от формата файла. Созданный файл находится в папке по умолчанию (создается автоматически). Если имя файла «Test1», символы записываются в объекты следующим образом.



- Путь к папке по умолчанию (на примере ЦПУ ПЛК AS300):

Модуль ЦПУ AS300	Путь к папке по умолчанию
AS332T-A	PLC CARD\AS300\UserProg
AS332P-A	
AS324MT-A	

- **S<sub>1</sub>**: Значение в файле, который считывается, указывается в **S<sub>1</sub>**.

Пункт	Описание
Единицы значения	Если формат файла 0, 1, 3 или 5, единицей значения является слово. Если формат файла 2, 4 или 6, единицей значения является двойное слово
Единицы параметров	Двойное слово
Применение	Значение в <b>S<sub>1</sub></b> должно указывать значение в файле; Если значение в <b>S<sub>1</sub></b> равно 0, считывается первое значение в файле

- **S<sub>3</sub>**: Длина данных, которые считываются из файла

Объекты, в которых сохраняются данные, не могут выходить за допустимые пределы. Если значение в **S<sub>3</sub>** больше, чем количество значений в файле, длина данных, считанных из файла – это количество значений в файле. Единица значения **S<sub>3</sub>** – двойное слово.

- **D**: Заданный операнд для сохранения считанных данных.

## 2. Соответствующие флаги:

Флаг	Описание
SM450	Если карта памяти вставлена в модуль ЦПУ, флаг включается
SM451	Переключатель защиты от записи на карте памяти: Флаг включен: Карта памяти защищена от записи; Флаг выключен: Карта памяти не защищена от записи
SM452	Данные записываются из ПЛК на карту памяти или данные считываются с карты памяти в ПЛК: флаг включен
SM453	Если во время работы карты памяти произошла ошибка, флаг будет включен. Если флаг включен, пользователь должен вручную его отключить. Код ошибки хранится в SR453.

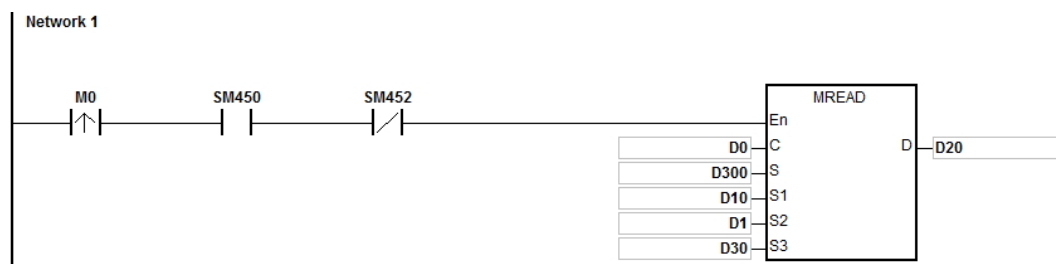
## 3. Коды ошибки (SR453):

Код ошибки	Описание
16#005E	Ошибка инициализации карты памяти
16#005F	Некорректный путь или файл не существует
16#0060	Папка по умолчанию не может быть создана
16#0061	Недостаточный объем памяти
16#0062	Карта памяти защищена от записи
16#0063	Произошла ошибка при записи данных в файл
16#0064	Данные не могут быть считаны с карты памяти
16#0065	Файл является доступным только для чтения

4. Если формат файла, в который записаны данные, равен 0, формат файла, из которого считываются данные, также равен 0. В противном случае данные не могут быть прочитаны и включается SM453. Аналогично по другим форматам файлов.

### Пример:

SM450 включается при помещении карты памяти в модуль ЦПУ; SM452 включается при выполнении инструкции MREAD и выключается после выполнения данной инструкции.



Операнд	Заданное значение	Описание
D0	16#0011	Файл, содержащий записанные данные Формат файла: Значения разделяются запятой Единицей значения является слово Файл с расширением .cvs. Принимаются коды ASCII
D300	D300=16#6554 D301=16#7473 D302=16#0031	Имя файла - «Test1»
D10, D11	16#00000000	Отображают значения в файле, начиная с первого значения
D1	16#000A	В каждую строку записываются десять значений
D30 · D31	16#00000020	Размер данных, считанных из файла, составляет 32 слова
D20	-	Считанные данные сохраняются в D20.

**Примечание:**

1. Если значение в **C** выходит за допустимые пределы возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение в **S<sub>2</sub>** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
3. Если значение в **S<sub>3</sub>** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
4. Если значение в **D** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции	Операнды	Функция
2302	MTWRIT P	C · S · S <sub>1</sub> · S <sub>2</sub> · S <sub>3</sub>	Запись строковых данных на карту памяти

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>C</b>								●	●				○	○		
<b>S</b>								●	●							
<b>S<sub>1</sub></b>								●	●				○	○		
<b>S<sub>2</sub></b>								●	●				○	○		
<b>S<sub>3</sub></b>								●	●							

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>C</b>		●											
<b>S</b>		●											
<b>S<sub>1</sub></b>		●											
<b>S<sub>2</sub></b>		●											
<b>S<sub>3</sub></b>													●

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**

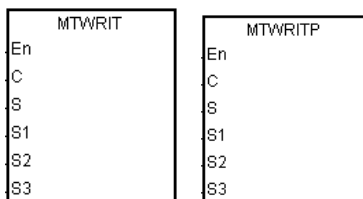
**C** : Параметр управления

**S** : Источник данных

**S<sub>1</sub>** : Длина данных

**S<sub>2</sub>** : Разделительная метка

**S<sub>3</sub>** : Имя файла



**Описание:**

1. Описание операндов:

- **C**: Параметр управления

Значение параметра	Описание
0 (Добавление)	Данные, записанные на карту памяти, добавляются после последнего значения в файле Если файл не существует, он создается автоматически
1	Если файл существует, новые данные, записанные на карту памяти, заменяют старые

(Overwriting)	данные в файле. Размер файла – это размер новых данных
	Если файл не существует, он создается автоматически

- **S**: Источник данных

Если строковое значение, записанное в файл, составляет «12345», символы сохраняются следующим образом. Так как байт принимается за базовую единицу, первый символ сохраняется в младшем байте в D300, второй символ сохраняется в старшем байте в D300. То же самое относится к другим символам. Символ «16#00» сохраняется в старшем байте в D300+2 и указывает конец строкового значения.

S300	S300+1	S300+2
byte 2	byte 1	byte 4
byte 1	byte 4	byte 3
byte 6	byte 5	

16#32	16#31	16#34	16#33	16#00	16#35
-------	-------	-------	-------	-------	-------

- **S<sub>1</sub>**: Длина данных, записываемых на карту памяти

Базовой единицей является байт. Объекты, в которых хранятся данные, не могут превышать допустимый диапазон для объектов, а длина данных, записанных на карту памяти, не может превышать 255 байт.

- **S<sub>2</sub>**: Разделительная метка

Если значение в **S<sub>1</sub>** равно N, значение в **S<sub>2</sub>** записывается на карту памяти следующим образом.

Операнд S <sub>2</sub>		Описание
Старший байт	Младший байт	
16#00	16#00 или отсутствие 16#00	В файл записывается N байт данных
Отсутствие 16#00	16#00	В файл записывается N+1 байт данных. Значение в старшем байте в <b>S<sub>2</sub></b> это значение в N+1 байте
Отсутствие 16#00	Отсутствие 16#00	В файл записывается N+2 байт данных. Значение в старшем байте в <b>S<sub>2</sub></b> это значение в седьмом байте, значение в младшем байте в <b>S<sub>2</sub></b> это значение в N+2 байте

- ПЛК серии AS с версией прошивки 1.04.00 или более поздней версии поддерживает использование строковых переменных (не более 9 символов) в имени файла **S<sub>3</sub>**.
- **S<sub>3</sub>~S<sub>3</sub>+4**: **S<sub>3</sub>** занимают пять объектов. Чаще всего имя файла составляют девять символов, включая 16#00. Если строковое значение не заканчивается символом 16#00, возникает ошибка. Если

конечный символ прочитан, чтение символов прекращается и проверяется корректность имени файла. Символами, которые могут использоваться для создания имени файла, являются A ~ Z, a ~ z и 0 ~ 9. Расширение имени файла зависит от формата файла. Созданный файл находится в папке по умолчанию (создается автоматически). Если имя файла «Test1», символы записываются в объекты следующим образом.



- Путь к папке по умолчанию (на примере ЦПУ ПЛК AS300):

Модуль ЦПУ AS300	Путь к папке по умолчанию
AS332T-A	PLC CARD\AS300\UserProg
AS332P-A	
AS324MT-A	

2. Соответствующие флаги:

Флаг	Описание
SM450	Если карта памяти вставлена в модуль ЦПУ, флаг включается
SM451	Переключатель защиты от записи на карте памяти: Флаг включен: Карта памяти защищена от записи; Флаг выключен: Карта памяти не защищена от записи
SM452	Данные записываются из ПЛК на карту памяти или данные считываются с карты памяти в ПЛК: флаг включен
SM453	Если во время работы карты памяти произошла ошибка, флаг будет включен. Если флаг включен, пользователь должен вручную его отключить. Код ошибки хранится в SR453.

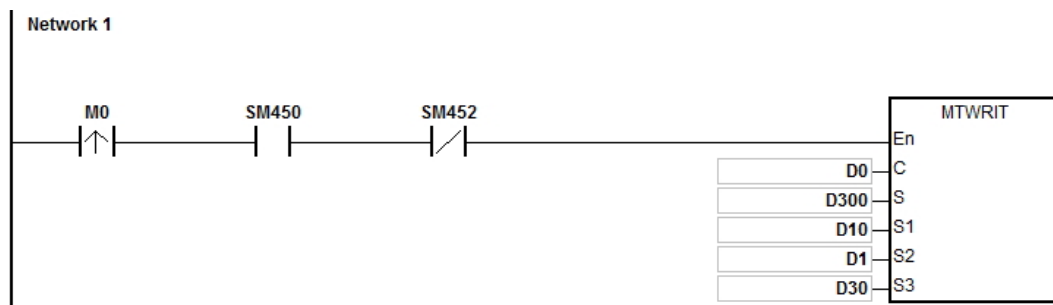
3. Коды ошибки (SR453):

Код ошибки	Описание
16#005E	Ошибка инициализации карты памяти
16#005F	Некорректный путь или файл не существует
16#0060	Папка по умолчанию не может быть создана
16#0061	Недостаточный объем памяти
16#0062	Карта памяти защищена от записи
16#0063	Произошла ошибка при записи данных в файл

Код ошибки	Описание
16#0064	Данные не могут быть считаны с карты памяти
16#0065	Файл является доступным только для чтения

**Пример:**

SM450 включается при помещении карты памяти в модуль ЦПУ; SM452 включается при выполнении инструкции MTWRIT и выключается после выполнения данной инструкции.



Операнд	Заданное значение	Описание
D0	16#0001	Файл, содержащий записанные данные Формат файла: Единицей значения является байт Расширение имени файла является .txt Принимаются коды ASCII Данные в D300 записываются в файл
D300	-	Данные, записанные в файл
D10	16#000A	Размер строкового значения, записанного в файл, составляет 10 байтов
D1	16#0A00	После того, как данные записаны в файл, разделительная метка добавляется после последнего байта в файле
D30	D30=16#6554 D31=16#7473 D32=16#0031	Имя файла "Test1".

**Примечание:**

1. Если значение в **C** выходит за допустимые пределы возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
2. Если значение в **S1** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
3. Если значение в **S3** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется,



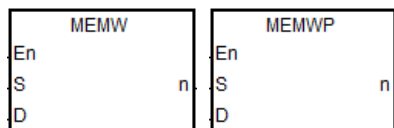
включается SM0 и в SR0 записывается код ошибки 16#2003.

API	Код инструкции			Операнды							Функция						
2303		MEMW	P	S · D · n							Запись данных в файловый регистр						

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S	●	●			●	●		●								
D									●							
n	●	●			●	●		●				●	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●											
D		●											
n		●											

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символ:**

**S** : Начальный адрес источника данных

**D** : Начальный адрес, в котором сохраняются данные

**n** : Длина данных

**Описание:**

- S**: Начальный адрес источника данных; объявляется тип массива переменных.
  - D**: Начальный адрес, в котором сохраняются данные; назначается адрес для файлового регистра и объявляется тип массива переменных.
  - n**: Длина данных, которые записываются в файловый регистр, диапазон от 1 до 2048. При превышении инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#200B.
  - Если значение в **S** или **D** выходит за допустимые пределы, возникает ошибка, инструкция не выполняется, включается SM0 и в SR0 записывается код ошибки 16#2003.
  - Поскольку для выполнения инструкции записи требуется 60 ~ 120 мс, рекомендуется использовать эту инструкцию, когда ПЛК находится в режиме ожидания. Например, нет задачи внешнего прерывания, не выполняется обработка высокоскоростного выхода или каких-либо других процедур обработки для ПЛК.
  - Инструкция выполняется только один раз и только тогда, когда включается соответствующий контакт.
- ПРИМЕЧАНИЕ: Файловый регистр может быть записан не более 100 000 раз.

## 6.24 Инструкции управления задачами

### 6.24.1 Описание инструкций управления задачами

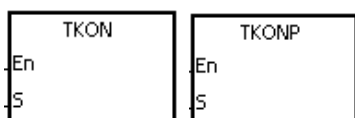
API	Код инструкции			Операнд								Функция			
2400		TKON	P	S								Запуск циклической задачи			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S	●	●						●	●		○	○	○			

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

Символьное обозначение:



S : Номер задачи

Описание:

1. Запуск циклической задачи, заданной операндом S.
2. Во время работы ПЛК выполнение циклической задачи зависит от настройки циклических задач в программе ISPSoft.
3. Описание операндов:
  - Операнд S может иметь значение в диапазоне от 0 до 31.
  - Подробнее см. порядок создания и запуска задач в руководстве пользователя программы ISPSoft.

Пример:

Во время работы ПЛК производится запуск циклической задачи (0). При выполнении инструкции TKON в циклической задаче (0), циклическая задача (1) выполняется и выход Y0.0 переходит в состояние ON.

В программе ISPSoft создано две циклических задачи. Циклическая задача (0) запускается во время работы ПЛК, а циклическая задача (1) не запускается во время работы ПЛК.

Циклическая задача (1) запускается инструкцией TKON в циклической задаче (0).



Циклическая задача (1) выполняется.



**Дополнительные замечания:**

Подробнее см. руководство пользователя программы ISPSOft.

API	Код инструкции			Операнд							Функция				
2401		TKOFF	P	<b>S</b>							Выключение циклической задачи				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S</b>	●	●						●	●		○	○	○			

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	AS	-

**Символьное обозначение:**

TKOFF	TKOFFP
En	En
S	S

**S** : Номер задачи

**Описание:**

1. Выключение циклической задачи, заданной операндом **S**.
2. Во время работы ПЛК выполнение циклической задачи зависит от настройки циклических задач в программе ISPSoft.
3. Описание операндов:
  - Операнд **S** может иметь значение в диапазоне от 0 до 31.
  - Подробнее см. порядок создания и запуска задач в руководстве пользователя программы ISPSoft.

**Пример:**

Когда ПЛК работает, циклическая задача (0) и циклическая задача (1) запущены. При выполнении инструкции TKOFP в циклической задаче (0), циклическая задача (1) выключается и выход Y0.0 переходит в состояние OFF.

В программе ISPSoft создано две циклических задачи. Циклическая задача (0) и циклическая задача (1) запускаются во время работы ПЛК, и циклическая задача (1) выключается после запуска инструкции TKOFF в циклической задаче (0).

Циклическая задача (1) выключается инструкцией TKOFF в циклической задаче (0).



Циклическая задача (1) не выполняется.



**Дополнительные замечания:**

Подробнее см. руководство пользователя программы ISPSoft.

## 6.25 Инструкции SFC

### 6.25.1 Описание инструкций SFC

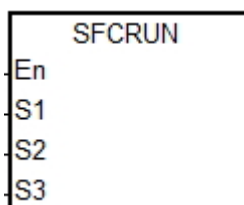
API	Код инструкции		Операнд				Функция				
2500		SFCRUN	$S_1, S_2, S_3$				Запуск программного модуля SFC				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$																
$S_2$								●					○	○		
$S_3$																

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$													
$S_2$		●			●	●							
$S_3$													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:



$S_1$  : Название программного модуля SFC

$S_2$  : Функциональный код

$S_3$  : Адрес регистра

Описание:

- Программный модуль SFC, заданный операндом  $S_1$ , запускается по значению операнда  $S_2$ .
- При выполнении инструкции программный модуль SFC, заданный операндом  $S_1$ , запускается только при сканировании программного модуля SFC.
- Операнд:
  - Операнд  $S_1$  задает имя программного модуля SFC.
  - При выполнении программного модуля SFC, заданного операндом  $S_1$ , такие параметры программы SFC, как SFC/STEP/ACTION/TRANSITION, стираются, если  $S_2=0$  или 1, и выполнение начинается в соответствии со значением операнда  $S_2$ .
  - Если  $S_2=0$ , ПЛК выполняет программный модуль SFC с первого шага.

- Если  $S_2=1$ , ПЛК выполняется программный модуль SFC с шага, заданного операндом  $S_3$ .
  - Если  $S_2=2$ , состояние и такие параметры, как SFC/STEP/ACTION/TRANSITION программы SFC, НЕ стираются и ПЛК начинает выполнение с места, где программа была остановлена.
  - Операнд  $S_3$  задает шаг, с которого будет начинаться выполнение программы SFC, заданной операндом  $S_1$ .
4. Операнд  $S_2$  имеет диапазон значений от 0 до 2. Если значение вне данного диапазона, оно берется равным 0.
  5. Если программный модуль SFC имеет состояние RUN, выполнение данной инструкции недействительно.

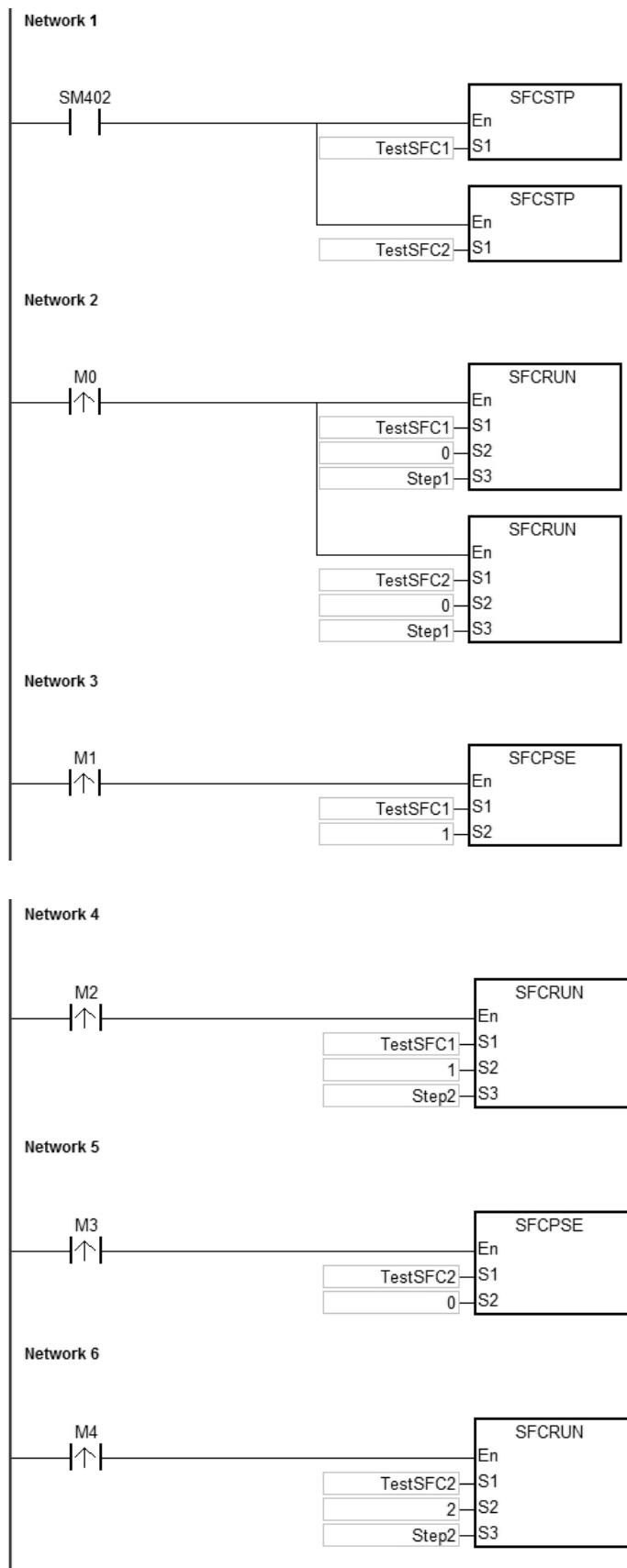
**Пример:**

Создайте один программный модуль на языке LD (ladder) и назовите его Main, потом еще два программных модуля SFC и дайте им имена TestSFC1 и TestSFC2.

1. При выполнении программы (RUN) программные модули TestSFC1 и TestSFC2 выполняют SFCSTP и два программных модуля SFC прекратят работу.
2. Когда состояние M0 меняется с OFF на ON, программный модуль TestSFC1/ TestSFC2 выполняет инструкцию SFCRUN\*. Подробнее см. содержимое программных модулей TestSFC1 и TestSFC2. Если  $S_2 = 0$ , состояние и параметры программы SFC стираются и начинается выполнение с ШАГА 1. Если  $S_2 = 1$ , состояние и параметры стираются и начинается выполнение с шага, заданного операндом  $S_3$ .
3. Если состояние M1 меняется с OFF на ON, выполнение программного модуля TestSFC1 приостанавливается. Если  $S_2 = 1$ , все действия и выходы программного модуля SFC стираются и ПЛК начинает заключительный цикл сканирования.
4. Если состояние M2 меняется с OFF на ON, программный модуль TestSFC1 выполняется. Если  $S_2 = 1$ , состояние и параметры программы SFC стираются и начинается выполнение с ШАГА 2.
5. Если состояние M3 меняется с OFF на ON, выполнение программного модуля TestSFC2 приостанавливается. Если  $S_2 = 0$ , все действия и выходы программного модуля SFC остаются без изменений и ПЛК не начинает заключительный цикл сканирования.
6. Если состояние M4 меняется с OFF на ON, программный модуль TestSFC1 выполняется. Если  $S_2 = 2$ , состояние и параметры программы SFC не стираются и начинается выполнение с места, где программа была остановлена.
6. \*SFCRUN запустит программный модуль SPC при следующем сканировании.

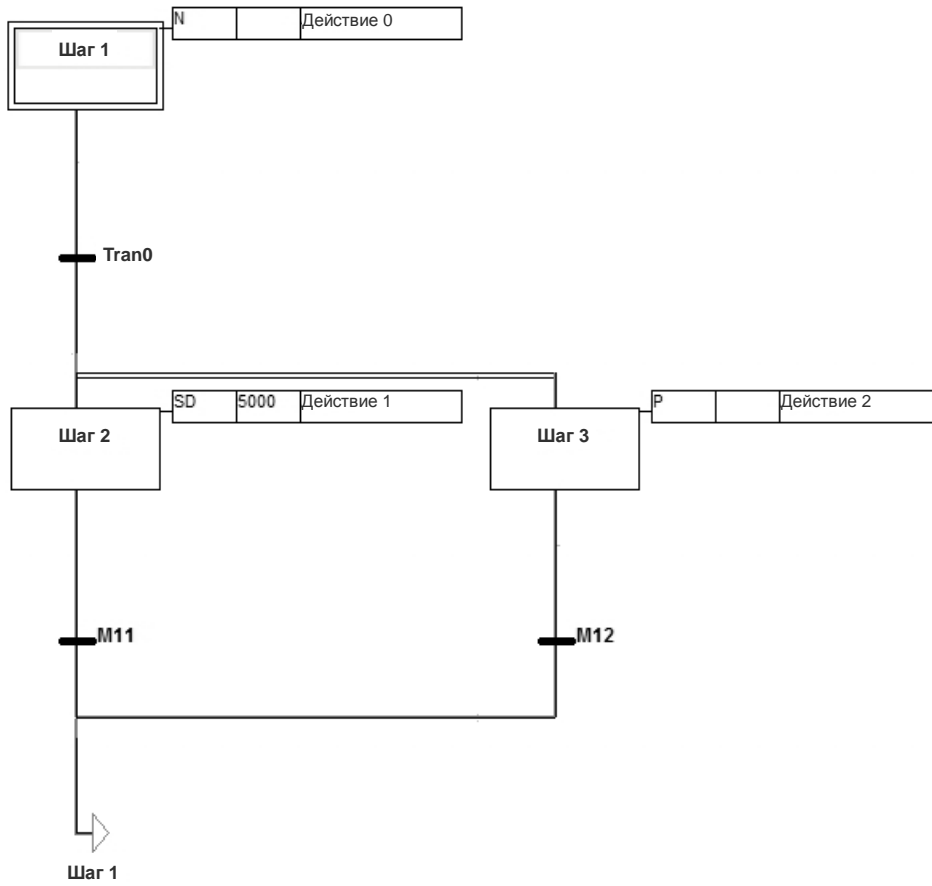


Программный модуль Main

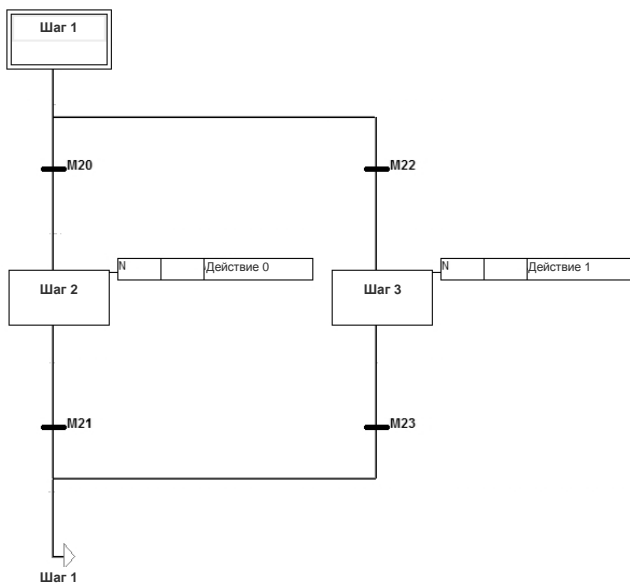


6

Программный модуль TestSFC1



Программный модуль TestSFC2



**Дополнительные замечания:**

Подробнее по SFC см. руководство пользователя программы ISPSOft.

API	Код инструкции			Операнд								Функция				
2501		SFCPSE		<b>S<sub>1</sub>, S<sub>2</sub></b>								Приостановка выполнения программного модуля SFC				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>																
<b>S<sub>2</sub></b>								●					○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>													
<b>S<sub>2</sub></b>		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

**Символьное обозначение:**

SFCPSE
En
S1
S2

- S<sub>1</sub>** : Название программного модуля SFC
- S<sub>2</sub>** : Функциональный код

**Описание:**

- Программный модуль SFC, заданный операндом **S<sub>1</sub>**, приостанавливается по значению операнда **S<sub>2</sub>**.
- При выполнении инструкции программный модуль SFC, заданный операндом **S<sub>1</sub>**, приостанавливается только при сканировании программного модуля SFC.
- Состояние и такие параметры приостановленного программного модуля SFC, как SFC/STEP/ACTION/TRANSITION, сохраняются.
- Операнд:
  - Операнд **S<sub>1</sub>** задает имя программного модуля SFC.
  - Если **S<sub>2</sub> = 0**, все действия и выходы программного модуля SFC остаются без изменений и ПЛК не начинает заключительный цикл сканирования.
  - Если **S<sub>2</sub> = 1**, все действия и выходы программного модуля SFC стираются и ПЛК начинает заключительный цикл сканирования.
- Операнд **S<sub>2</sub>** имеет диапазон значений от 0 до 1. Если значение вне данного диапазона, оно берется равным 0.
- Если программный модуль SFC имеет состояние PAUSE/STOP, выполнение данной инструкции недействительно.

**Пример:**

Подробнее см. инструкцию SFCRUN.

**Дополнительные замечания:**

Подробнее по SFC см. руководство пользователя программы ISPSOft.

API	Код инструкции		Операнд								Функция				
2502		SFCSTP	S								Прекращение выполнения программного модуля SFC				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S																

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>													

Импульсная инструкция	16-битная инструкция	32-битная инструкция
-	AS	-

Символьное обозначение:

SFCSTP
En
S

**S** : Название программного модуля SFC

Описание:

1. Программный модуль SFC, заданный операндом **S**, останавливается.
2. При выполнении инструкции программный модуль SFC, заданный операндом **S<sub>1</sub>**, останавливается только при сканировании программного модуля SFC.
3. После остановки состояние и параметры программного модуля SFC стираются, и ПЛК запускает заключительный цикл сканирования.
4. Если программный модуль SFC имеет состояние STOP, выполнение данной инструкции недействительно.

Пример:

Подробнее см. инструкцию SFCRUN.

Дополнительные замечания:

Подробнее по SFC см. руководство пользователя программы ISPSOft.

## 6.26 Инструкции высокоскоростных выходов

### 6.26.1 Описание инструкций высокоскоростных выходов

API	Инструкция			Операнд							Описание						
2700	D	PLSY		<b>S<sub>1</sub>, S<sub>2</sub>, D</b>							Высокоскоростной импульсный выход (без линейного повышения/понижения)						

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>							●	●	●		○		○	○		
<b>S<sub>2</sub></b>							●	●	●		○		○	○		
<b>D</b>		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>			●				●						
<b>S<sub>2</sub></b>			●				●						
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

Символьное обозначение:

DPLSY	
En	
S1	D
S2	

**S<sub>1</sub>** : Частота выходного импульса

**S<sub>2</sub>** : Количество выходных импульсов

**D** : Импульсный выход

Описание:

- Операнд **S<sub>1</sub>** задает частоту выходных импульсов равной 4 МГц для моделей с дифференциальным выходом и значение в диапазоне 0 Гц ~ 200 кГц для моделей с выходом типа открытый коллектор. Единица измерения выходной частоты 1 Гц. Допустимая погрешность частоты для 200 кГц составляет примерно 0,02 %, а для частоты 100 кГц примерно 0,01 %. Погрешность снижается по мере понижения частоты. Например, если частота выходных импульсов равна 199990 Гц, реальная частота выходных импульсов будет 199960 Гц. Если частота выходных импульсов равна 99999 Гц, реальная частота выходных импульсов будет 99990 Гц. Если частота выходных импульсов вне пределов допустимого диапазона, ПЛК автоматически выставляет на выходе минимальную или максимальную частоту.
- Частота выходных импульсов, заданная операндом **S<sub>1</sub>**, может меняться во время выполнения инструкции без линейного понижения или повышения. Момент, когда происходит смена частоты, наступает при сканировании выполняемой инструкции по завершении выдачи всех импульсов данной инструкции.

3. Операнд **S<sub>2</sub>** определяет количество выходных импульсов. Диапазон: 0~2,147,483,647. Если количество выходных импульсов равно 0, значит количество не ограничено и импульсы могут выдаваться постоянно, пока инструкция не будет выключена. Если количество выходных импульсов меньше 0, они не выдаются.
4. Если выдача выходных импульсов начата, их количество, задаваемое операндом **S<sub>2</sub>**, изменить нельзя.
5. Операнд **D** только присваивает дискретному выходу Y0.0~Y0.11 функцию импульсного выхода. Когда инструкция запущена, выход становится высокоскоростным выходом. Стандартный механизм управления выходом становится недействительным. Предполагается, что высокоскоростной выход не будет использоваться в программе для других функций.
6. Отношение времени включенного состояния (Duty-OFF Time) и выключенного (Duty-ON Time) импульсного выхода составляет 1:1.
7. Ограничений по количеству применений инструкции нет, равно как нет ограничений по количеству повторного применения выходов. Но за один цикл сканирования каждый дискретный выход может становиться активным и выдавать импульсы только под управлением одной инструкции высокоскоростного выхода, и та инструкция, которая была запущена первой, первой и берет на себя управление этим дискретным выходом.
8. После запуска инструкции высокоскоростного выхода для любого дискретного выхода другая инструкция, которая попытается использовать этот же выход, запущена быть не может, пока выполняемая в данный момент инструкция высокоскоростного выхода не будет остановлена.
9. Если инструкция высокоскоростного выхода запущена в программе прерывания или не отредактирована в главной программе, предполагается, что эта инструкция будет использоваться вместе с функцией автоматического сброса по завершении выдачи выходных импульсов и ПЛК обновит состояние выхода в инструкции END.
10. После постановки флага СТОП ПЛК прекращает выдавать импульсы и снимает флаг ЗАНЯТО (Busy) только после выполнения инструкции СТАРТ во второй раз и после выдачи всех импульсов. ПЛК продолжает выдавать импульсы, когда флаг СТОП сброшен, и оставшиеся выходные импульсы еще не выданы до конца. Флаг СТОП устанавливается и сбрасывается инженерами по проектированию.
11. Высокоскоростные дискретные выходы и соответствующие SM/SR приведены ниже в таблице.

Номер выхода	Атрибут <sup>#2</sup>	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Флаг ЗАНЯТО	R	SM460	SM472	SM480	SM492	SM500	SM512
Флаг ЗАВЕРШЕНО <sup>#3</sup>	R/W	SM461	SM473	SM481	SM493	SM501	SM513
Флаг СТОП	R/W	SM463	SM474	SM483	SM494	SM503	SM514
Автоматический сброс завершения выдачи импульсов <sup>#4</sup>	R/W	SM470	SM475	SM490	SM495	SM510	SM515
Текущее состояние выхода <sup>#1</sup> (32 бита)	R/W	SR460 SR461	SR474 SR475	SR480 SR481	SR494 SR495	SR500 SR501	SR514 SR515

Номер выхода	Атрибут <sup>#2</sup>	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Номер выхода	Атрибут <sup>#2</sup>	Y0.6	Y0.7	Y0.8	Y0.9	Y0.10	Y0.11
Флаг ЗАНЯТО	R	SM520	SM532	SM540	SM552	SM560	SM572
Флаг ЗАВЕРШЕНО <sup>#3</sup>	R/W	SM521	SM533	SM541	SM553	SM561	SM573
Флаг СТОП	R/W	SM523	SM534	SM543	SM554	SM563	SM574
Автоматический сброс завершения выдачи импульсов <sup>#4</sup>	R/W	SM530	SM535	SM550	SM555	SM570	SM575
Текущее состояние выхода <sup>#1</sup> (32 бита)	R/W	SR520 SR521	SR534 SR535	SR540 SR541	SR554 SR555	SR560 SR561	SR574 SR575

Примечание:

#1: Все текущие состояния выходов блокируются после выключения питания.

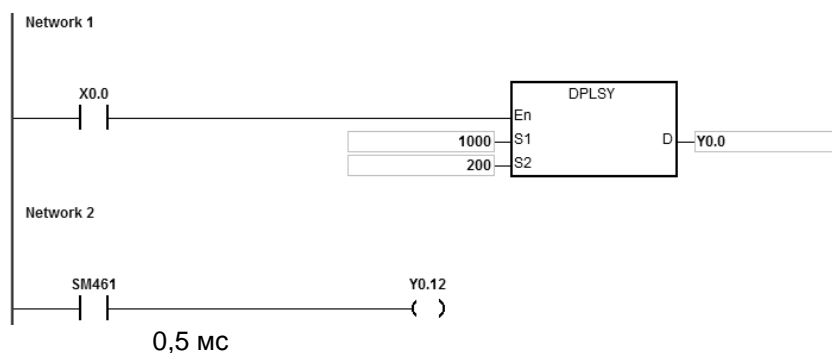
#2: R означает регистры Ready-only, и данные в этих регистрах изменить нельзя. R/W значит, что данные в таких регистрах можно считывать и записывать.

#3: Предполагается, что флаг ЗАВЕРШЕНО должен сниматься инженерами по проектированию. Если флаг ЗАВЕРШЕНО не снят, он будет снят автоматически, когда инструкция высокоскоростного выхода будет запущена в следующий раз.

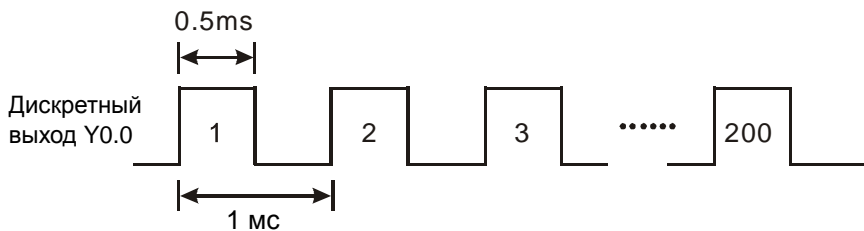
#4: Флаг автоматического сброса завершения выдачи импульсов ставится инженерами по проектированию. ПЛК снимает этот флаг автоматически по завершении выдачи импульсов.

**Пример 1:**

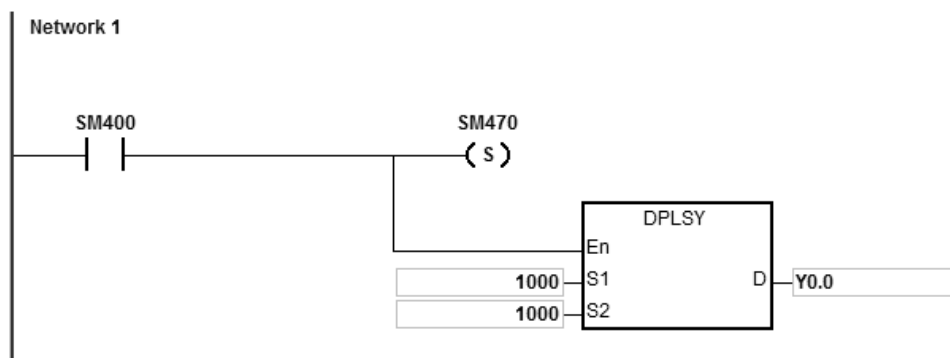
1. Если X0.0 = ON, дискретный выход Y0.0 подает 200 импульсов с частотой 1 кГц. Когда все импульсы выданы, SM461 = ON и потом Y0.12 = ON.
2. Если X0.0 = OFF, дискретный выход Y0.0 прекращает выдавать импульсы. Выдача импульсов возобновляется, когда вход X0.0 снова становится активным (ON).



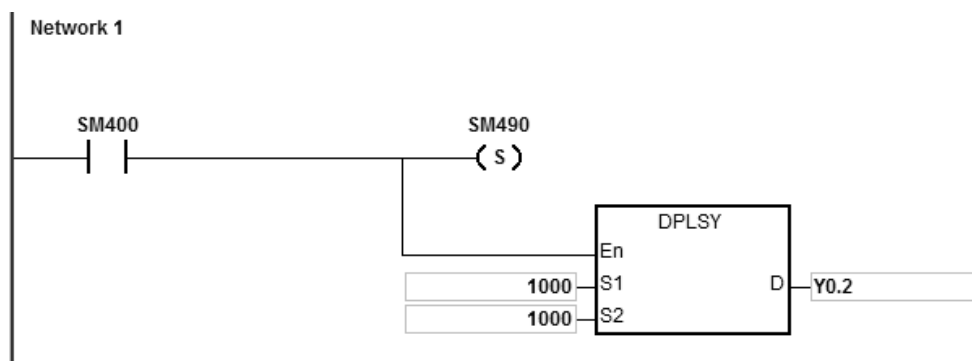




**Пример 2:**



Вход внешнего сигнала прерывания X0.1



**Описание:**

1. Выход Y0.0 выдает 1000 импульсов каждый раз, когда на вход X0.0 поступает один внешний сигнал прерывания. Выход Y0.2 выдает 1000 импульсов каждый раз, когда на вход X0.1 поступает один внешний сигнал прерывания.
2. Когда на вход X поступает внешний сигнал прерывания и выход Y соответственно начинает выдавать импульсы, интервал времени между моментом, выдача импульсов выходом Y завершена, и моментом, когда будет получен следующий внешний сигнал прерывания на входе X, должен равняться одному циклу опроса ПЛК или дольше.

**Пример 3: (программа ST)**

```
0001 IF M0 THEN
0002 DPLSY (1000,1000,Y0.0) ;
0003 ELSIF SM461 THEN
0004 SM470:=TRUE;
0005 END_IF;
0006
0007
```

Описание:

1. Если M0 = ON, выход Y0.0 подает 1000 импульсов с частотой 1 кГц.
2. Когда все импульсы выданы, SM461 = ON и потом SM470= ON.
3. Когда состояние M0 меняется с OFF на ON, выдача импульсов начинается заново.
4. Если в программе указано более двух высокоскоростных инструкций вывода, не используйте имя переменной M1 повторно.
5. Предлагается не использовать язык ST в программе, если есть вероятность, что выход необходимо остановить.

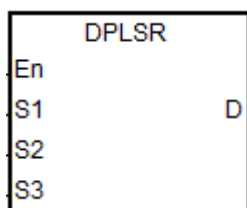
API	Инструкция			Операнд							Описание					
2701	D	PLSR		$S_1, S_2, S_3, D$							Высокоскоростной импульсный выход (с линейным повышением/понижением)					

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
D		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**

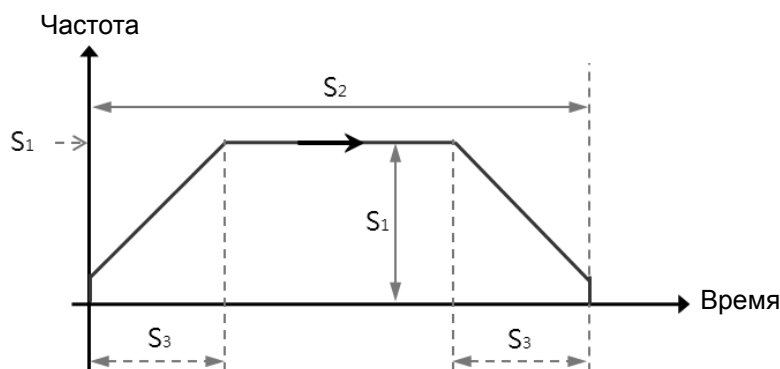


- $S_1$  : Заданная частота выходных импульсов
- $S_2$  : Количество выходных импульсов
- $S_3$  : Время линейного повышения/понижения
- D : Импульсный выход

**Описание:**

- Операнд  $S_1$  задает частоту выходных импульсов равной 4 МГц для моделей с дифференциальным выходом и значение в диапазоне 0 Гц ~ 200 кГц для моделей с выходом типа открытый коллектор. Единица измерения частоты выхода 1 Гц. Допустимая погрешность частоты для 200 кГц составляет примерно 0,02 %, а для частоты 100 кГц примерно 0,01 %. Погрешность снижается по мере понижения частоты. Например, если частота выходных импульсов равна 199990 Гц, реальная частота выходных импульсов будет 199960 Гц. Если частота выходных импульсов равна 99999 Гц, реальная частота выходных импульсов будет 99990 Гц. Если частота выходных импульсов вне пределов допустимого диапазона, ПЛК автоматически выставляет на выходе минимальную или максимальную частоту.
- После запуска инструкции частоту выходных импульсов, задаваемую операндом  $S_1$ , можно изменить, и будет выполняться процесс линейного повышения/понижения частоты в соответствии с заданным временем. Момент, когда происходит смена частоты, наступает при сканировании выполняемой инструкции по завершении выдачи всех импульсов данной инструкции.

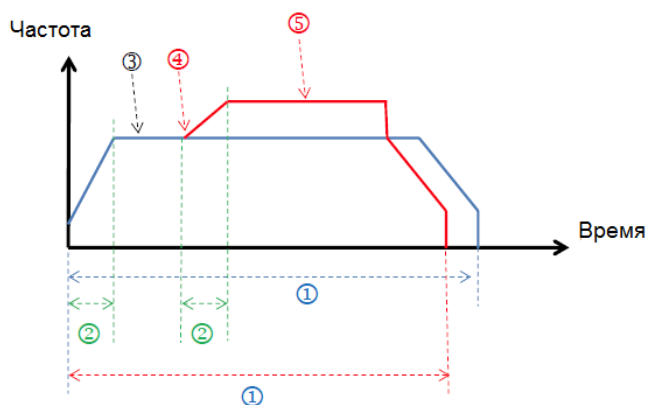
3. Операнд **S<sub>2</sub>** определяет количество выходных импульсов. Диапазон: 0~2,147,483,647. Если количество выходных импульсов равно 0, значит количество не ограничено и импульсы могут выдаваться постоянно, пока инструкция не будет выключена.
4. Если выдача выходных импульсов начата, их количество, задаваемое операндом **S<sub>2</sub>**, изменить нельзя.
5. Операнд **S<sub>3</sub>** определяет время линейного повышения/понижения частоты с дискретностью 1 мс. Значение операнда действительно, когда инструкция запускается в первый раз. Если частота выходных импульсов, заданная операндом **S<sub>1</sub>**, изменяется во время линейного повышения частоты, время линейного повышения/понижения частоты будет перезагружено для выполнения. Но если частота импульсов меняется, когда выход находится в состоянии понижения частоты, изменение будет недействительным.
6. Операнд **D** только присваивает дискретному выходу Y0.0~Y0.11 функцию импульсного выхода. Когда инструкция запущена, выход становится высокоскоростным выходом. Стандартный механизм управления выходом становится недействительным. Предполагается, что высокоскоростной выход не будет использоваться в программе для других функций.
7. Заданная частота выходных импульсов **S<sub>1</sub>**, количество выходных импульсов **S<sub>2</sub>** и время линейного повышения/понижения частоты **S<sub>3</sub>** показаны на следующем рисунке.



8. Отношение времени включенного состояния (Duty-OFF Time) и выключенного (Duty-ON Time) импульсного выхода составляет 1:1.
9. Ограничений по количеству применений инструкции нет, равно как нет ограничений по количеству повторного применения выходов. Но за один цикл сканирования каждый выход может становиться активным и выдавать импульсы только под управлением одной инструкции высокоскоростного выхода и та инструкция, которая запустила выход первой, первой и берет на себя управление этим дискретным выходом.
10. После запуска инструкции высокоскоростного выхода для любого дискретного выхода другая инструкция, которая попытается использовать этот же выход, запущена быть не может, пока выполняемая в данный момент инструкция высокоскоростного выхода не будет остановлена.
11. Если инструкция высокоскоростного выхода запущена в программе прерывания или не отредактирована в главной программе, предполагается, что эта инструкция будет использоваться вместе с функцией автоматического сброса по завершении выдачи выходных импульсов и ПЛК обновит состояние выхода в инструкции END.

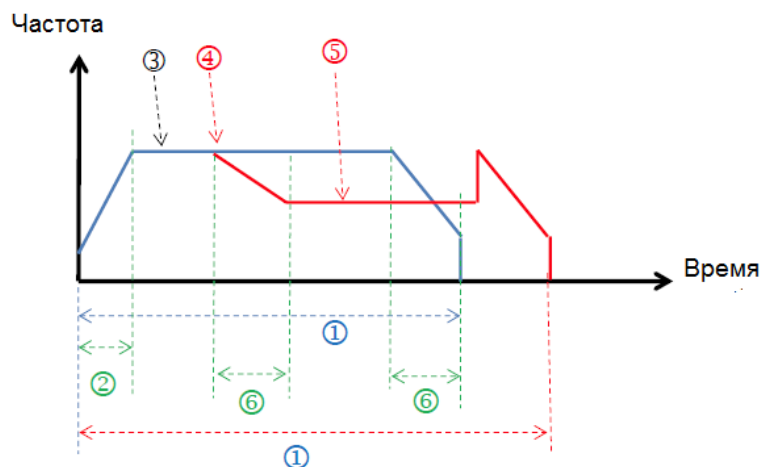
12. После постановки флага СТОП ПЛК выполняет линейное понижение частоты или немедленно прекращает выдавать импульсы только после выполнения инструкции СТАРТ во второй раз и после выдачи всех импульсов.
13. Подробнее см. описание высокоскоростных выходов и соответствующие SM/SR в инструкции PLSY.
14. Обратитесь к примеру 3 из инструкции PLSY (API 2700) для программирования на языке ST.
15. Когда заданная выходная частота будет достигнута, вы можете изменить установленную заданную выходную частоту. Частоты для нарастания и замедления уже установлены, если вы измените заданную выходную частоту во время выполнения инструкции, частота будет либо нарастать, либо уменьшаться. Если установлено несколько заданных выходных частот, предлагается установить последнюю новую заданную частоту так же, как и исходную заданную частоту, чтобы избежать резких изменений при останове. См. Пример ниже. Синяя линия указывает исходную заданную частоту, а красная линия указывает на новую заданную частоту.

**Ситуация А: разгон по рампе**



- ① Число выходных импульсов
- ② Время разгона по рампе
- ③ Первая заданная частота
- ④ Позиция ввода новой заданной частоты
- ⑤ Новая заданная частота

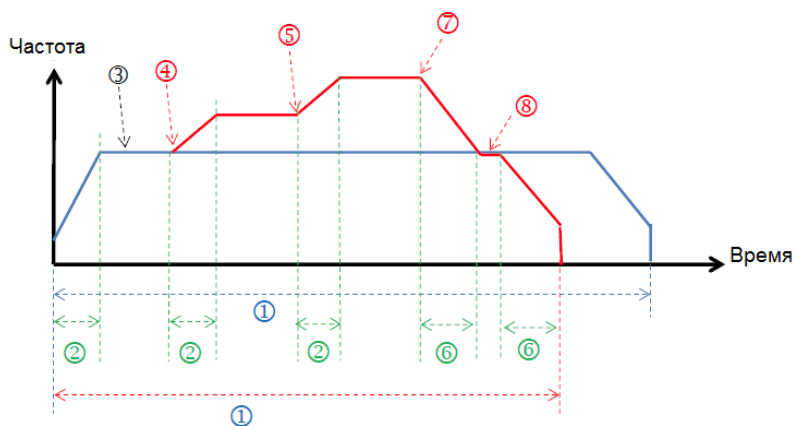
**Ситуация В: торможение по рампе**



- ① Число выходных импульсов
- ② Время разгона по рампе
- ③ Первая заданная частота
- ④ Позиция ввода новой заданной частоты
- ⑤ Новая заданная частота
- ⑥ Время торможения по рампе

**Предлагаемое решение:** предлагается установить последнюю новую заданную частоту так же, как и исходную заданную частоту, чтобы избежать резких изменений при останове.

**6**



- ① Число выходных импульсов
- ② Время разгона по рампе
- ③ Первая заданная частота
- ④ Позиция ввода второй заданной частоты
- ⑤ Новая заданная частота
- ⑥ Время торможения по рампе
- ⑦ Позиция ввода третьей заданной частоты
- ⑧ Позиция изменения первоначальной заданной частоты, а затем ее уменьшения до останова

API	Инструкция			Операнд								Описание				
2702	D	PWM		$S_1, S_2, D$								Импульсный выход с ШИМ				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$	●	●			●	●	●	●	●		○		○	○		
$S_2$	●	●			●	●	●	●	●		○		○	○		
D		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	AS

Символьное обозначение:

PWM		DPWM	
$E_n$		$E_n$	
$S_1$	D	$S_1$	D
$S_2$		$S_2$	

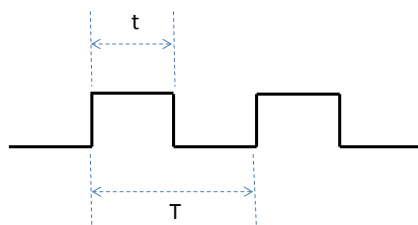
- $S_1$  : Ширина импульса
- $S_2$  : Продолжительность периода
- D : Импульсный выход

Описание:

1. 16-битная инструкция PWM имеет дискретность задания 100 мкс, а 32-битная инструкция DPWM - 1 мкс.

Инструкция	PWM	DPWM
Диапазон ширины импульса, $S_1$	0~30000	0~60000
Диапазон цикла импульса, $S_2$	0~30000	0~60000

2. Операнд  $S_1$  определяет длительность импульса (Duty ON) как время t, операнд  $S_2$  определяет время цикла импульса (время цикла) как T, как показано ниже. Рекомендуется выставлять  $S_1 \leq S_2$ .



3. Операнд D только задает дискретный выход в диапазоне от Y0.0 до Y0.11.
4. Ограничений по количеству использования инструкции в программе нет. Но в рамках одного цикла сканирования может выполняться только одна инструкция, использующая один и тот же дискретный выход

(нельзя использовать две инструкции, использующие один и тот же выход). Если несколько инструкций высокоскоростных выходов запускают в программе один выход, ПЛК первым активирует выход, инструкция которого выполняется первой.

5. Если  $S_1 \neq 0$  или  $S_2 \neq 0$ , выход будет 0 (импульсный выход ВЫКЛЮЧЕН). Если  $S_1 > S_2$ , это расценивается как  $S_1 = S_2$ . Если  $S_1 = S_2$  и  $S_2$  не равен 0, импульсный выход всегда активен (ON).
6. Длительность импульса  $S_1$ , и цикл импульса  $S_2$  можно изменять во время выполнения PWM.
7. Соответствующие специальные регистры (SR) перечислены ниже в таблице.

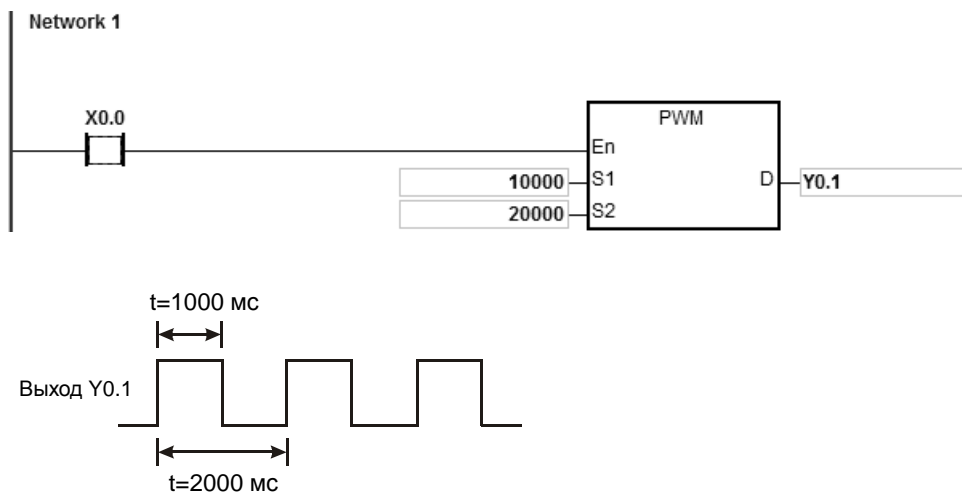
Номер дискретного выхода	Атрибут	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Текущее состояние выхода <sup>#1</sup> (32 бита)	R/W	SR460	SR474	SR480	SR494	SR500	SR514
		SR461	SR475	SR481	SR495	SR501	SR515
Номер дискретного выхода	Атрибут	Y0.6	Y0.7	Y0.8	Y0.9	Y0.10	Y0.11
Текущее состояние выхода <sup>#1</sup> (32 бита)	R/W	SR520	SR534	SR540	SR554	SR560	SR574
		SR521	SR535	SR541	SR555	SR561	SR575

#1: все текущие состояния выходов блокируются после выключения питания.

8. См. Пример 3 инструкции DJOG (API 2703) для программирования на языке ST.

**Пример:**

Если X0.0 = ON, выход Y0.1 передает следующие импульсы. Если X0.0 = OFF, выход Y0.1 переходит в состояние OFF.





API	Инструкция			Операнд						Описание			
2703	D	JOG		$S_1, S_2, S_3, D_1, D_2$						Выход JOG			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$					●	●	●	●	●		○		○	○		
$S_2$					●	●	●	●	●		○		○	○		
$S_3$					●	●	●	●	●		○		○	○		
$D_1$		○														
$D_2$		○	○													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●	●		●	●	●						
$S_2$		●	●		●	●	●						
$S_3$		●	●		●	●	●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	AS

Символьное обозначение:

JOG		DJOG	
En		En	
S1	D1	S1	D1
S2	D2	S2	D2
S3		S3	

- $S_1$  : Время линейного повышения
- $S_2$  : Заданная частота выходных импульсов
- $S_3$  : Время линейного понижения
- $D_1$  : Импульсный выход
- $D_2$  : Доп. выход

Описание:

- Операнд  $S_1$  определяет время линейного повышения частоты с дискретностью 10 мс. Например: если введено значение 10, значит время линейного повышения частоты равно 100 мс. Если инструкция включена, частоту выходных импульсов можно делить на десять, и будет производиться повышение частоты каждые 10 мс. По истечении времени линейного повышения частоты, заданной операндом  $S_1$ , частота будет поднята до частоты толчкового хода (JOG), заданной операндом  $S_2$ .
- Если инструкция выключена, частота выходных импульсов будет линейно понижаться каждые 10 мс в соответствии с заданным временем линейного понижения частоты. Выход будет оставаться в активном состоянии, пока не истечет время линейного понижения частоты. Если время линейного понижения частоты задано равным 0, выход немедленно переходит в состояние OFF.

3. Диапазон значений операндов **S<sub>1</sub>** (время линейного повышения частоты) и **S<sub>3</sub>** (время линейного понижения частоты): 0~3000 (т. е. 0~30 с). Если значение вне пределов допустимого диапазона, ПЛК автоматически выставляет на выходе минимальное или максимальное значение. Время сканирования влияет на время линейного повышения/понижения частоты. Если нужно точное время переключения операций линейного понижения/повышений частоты, рекомендуется пользоваться инструкцией типа DDRVI, имеющей подходящую функцию линейного повышения/понижения частоты.
4. Диапазон значений операнда **S<sub>2</sub>**: -200К~200 кГц. Если значение вне пределов допустимого диапазона, ПЛК автоматически выставляет на выходе минимальное или максимальное значение. Если частота выхода положительная, значит направление прямое (forward output). Если частота выхода отрицательная, значит направление обратное (reverse output).
5. Диапазон значений операнда **D<sub>1</sub>** от Y0.0 до Y0.11, как показано в таблице ниже. В операнде **D<sub>2</sub>** можно использовать выходы, приведенные ниже. Если пользователь выбирает другой выход или регистр M, значение специального регистра (SR), показывающее режим выхода, будет недействительным, а режим «импульс + направление» будет действительным по умолчанию. Операнд **D<sub>2</sub>** в данном случае задаёт выход направления.

Операнд **D<sub>1</sub>** использует четный номер дискретного выхода.

Выход для <b>D<sub>1</sub></b>	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Выход направления для <b>D<sub>2</sub></b>	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Флаг ЗАНЯТО	SM460	SM480	SM500	SM520	SM540	SM560
Режим выхода	SR462	SR482	SR502	SR522	SR542	SR562
Текущее состояние выхода	SR460 SR461	SR480 SR481	SR500 SR501	SR520 SR521	SR540 SR541	SR560 SR561

Операнд **D<sub>1</sub>** использует нечетный номер дискретного выхода.

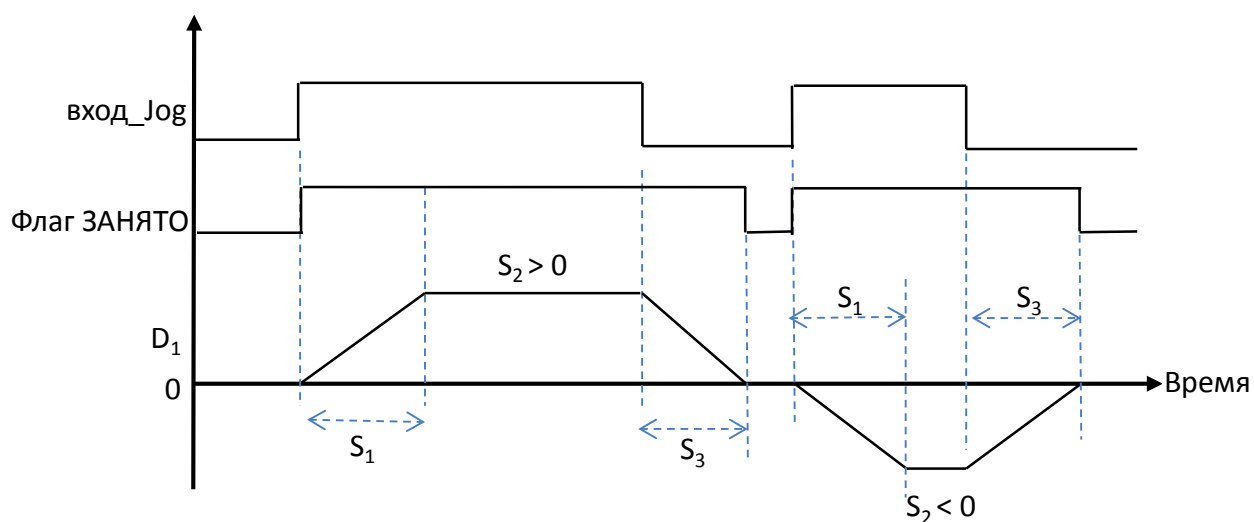
Выход для <b>D<sub>1</sub></b>	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Выход направления для <b>D<sub>2</sub></b>	Y1.0 или больше или любой регистр M (Bool)					
Флаг ЗАНЯТО	SM472	SM492	SM512	SM532	SM552	SM572
Текущее состояние выхода	SR474 SR475	SR494 SR495	SR514 SR515	SR534 SR535	SR554 SR555	SR574 SR575

6. Если SR=0, значит режим выхода «импульс + направление». Если SR=1, значит выхода фазы A/B. (Прим.: режим выхода можно выбрать, только если у операнда **D<sub>1</sub>** четный номер дискретного выхода, а у операнда **D<sub>2</sub>** рекомендованный дискретный выход направления)
7. Когда выходы направления не используют выходы по умолчанию для вывода, вы можете обратиться к приведенной ниже таблице, чтобы установить время вывода, которое выводит импульс первым, прежде чем выводится направление. Таким образом, вы можете быть уверены, что вывод происходит только при переключении на выходы направления. Единица измерения SR составляет миллисекунду, а диапазон составляет 0-20 мс. Значение по умолчанию равно 0 указывает на неактивность.

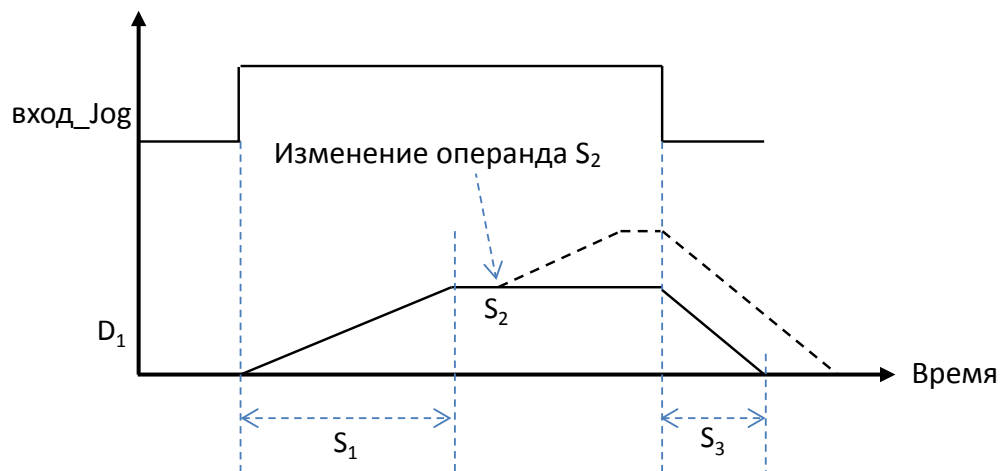
Выход D1	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Номер SR	SR640	SR641	SR642	SR643	SR644	SR645
Выход D1	Y0.6	Y0.7	Y0.8	Y0.9	Y1.0	Y1.1
Номер SR	SR646	SR647	SR648	SR649	SR650	SR651

Примечание: источник времени здесь - от времени сканирования ПЛК. (доступно для версии прошивки 1.02.20 или выше)

- Когда выход начинает работать, флаг ЗАНЯТО SM переходит в состояние ON. По завершении работы выхода флаг ЗАНЯТО переходит в состояние OFF автоматически и флаг ЗАВЕРШЕНО не переходит в состояние ON.
- Временная диаграмма работы выхода: (по флагу ЗАНЯТО на следующем рисунке см. флаги ЗАНЯТО по осям).



- Заданную частоту выхода можно изменить во время выполнения инструкции. Если новая частота больше предыдущей, будет производиться линейное повышение частоты. А если новая частота ниже, тогда частота будет линейно понижаться.
- Линейное повышение/понижение частоты задается через изменение времени и заданной частоты после включения инструкции. Таким образом, линейное изменение частоты не меняется при изменении заданной частоты во время работы инструкции выхода. Например, изначально частота выхода была 1 кГц и далее была изменена до 2 кГц. Время линейного понижения частоты будет отличаться от первого заданного времени. На рисунке ниже пунктирной линией показано линейное повышение/понижение частоты после изменения заданной частоты.



#### Пример (программирование на языке ST):

В этой инструкции предлагается использовать лестничные диаграммы в качестве языка программирования. Если вам нужно использовать язык ST, необходимо использовать флаг завершения вывода для того, чтобы ПЛК определил, что выходная точка свободна и готова к использованию снова, когда выход завершен. Принимая `Y0.0` в качестве выходной точки, программа должна быть записана, как показано на примере ниже.

```

0001 IF M0 THEN
0002     DJOG(1000,5000,100,Y0.0,Y0.1);
0003     M1 := TRUE;
0004 ELSIF (NOT M0) AND M1 THEN
0005     SM470 := TRUE;
0006     M1 := FALSE;
0007 END IF;

```

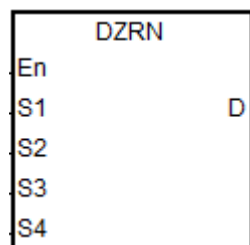
API	Инструкция			Операнд								Описание				
2704	D	ZRN		$S_1, S_2, S_3, S_4, D$								Возврат в ноль				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
$S_4$	○															
D		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$	●												
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**



- $S_1$  : Заданная частота для возврата в ноль
- $S_2$  : Частота JOG для DOG
- $S_3$  : Режим возврата в ноль
- $S_4$  : Дискретный вход для DOG
- D : Импульсный выход

**Описание:**

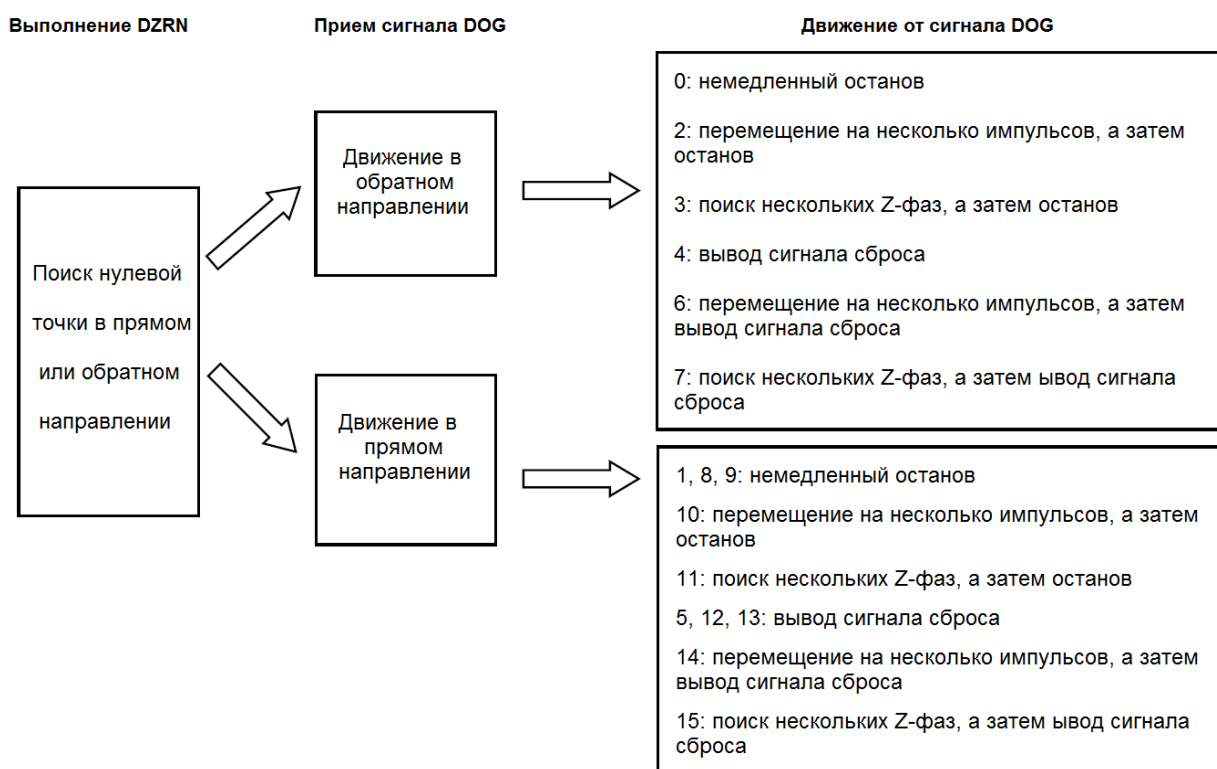
1. Диапазон значений заданной частоты операнда  $S_1$  от 1 Гц до 200 кГц. Частота JOG, задаваемая в операнде  $S_2$ , должна быть меньше частоты, задаваемой в операнде  $S_1$ . Частота JOG, задаваемая в операнде  $S_2$ , представляет собой начальную частоту. Если  $S_1$  меньше  $S_2$ ,  $S_1$  автоматически скинется до значения, равного значению операнда  $S_2$ .
2. Дискретный вход операнда  $S_4$  и дискретный выход операнда D связаны друг с другом. Запрещается менять их во время выполнения инструкции. Вход операнда  $S_4$  может быть только одним из 16 дискретных высокоскоростных входов X0.0~X0.15. Дискретный выход операнда D и выход направления выбираются по таблице ниже. Если операнд D не является текущим выходом «импульс + направление» (по умолчанию: 0), измените режим на выход фаза A/B, выставив регистр SR = 1.

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Дискретный вход операнда <b>S<sub>4</sub></b>	Может быть любым дискретным входом от X0.0 до X0.15. Один вход нельзя выбрать для разных выходов осей. Если есть пульсации на входе DOG или есть дребезг контактов выключателя, введите время фильтра входа в HWCONFIG.					
Дискретный выход операнда <b>D</b>	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Дискретный выход направления	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Режим выхода	SR462	SR482	SR502	SR522	SR542	SR562
Флаг ЗАНЯТО	SM460	SM480	SM500	SM520	SM540	SM560
Флаг ЗАВЕРШЕНО	SM461	SM481	SM501	SM521	SM541	SM561
Текущее состояние выхода	SR460 SR461	SR480 SR481	SR500 SR501	SR520 SR521	SR540 SR541	SR560 SR561

3. Операнд **S<sub>3</sub>** предназначен для выбора режима возврата в ноль, и функциональный код задается двумя младшими и старшими 16-битными кодами. См. подробности ниже.

<b>S<sub>3</sub>: Выбор режима возврата в нулевую точку</b>				
Старш. 16 бит	Младш. 16 бит			
b31~b16	b15~b6	b5	B4	b3~b0
Число импульсов для движения	Зарезервированы	Настройка направления: 0: обратное 1: прямое	Настройка сигнала DOG: 0: контакт А 1: контакт В	Выбор режима: 0~15 (F)
Число Z-фаз				

Примечание 1: Эта функция доступна только для версии прошивки 1.04.30 или выше. См. диаграмму ниже для настройки режима.



4. Операнд **S<sub>3</sub>** предназначен для выбора режима возврата в ноль, и функциональный код задается двумя младшими и старшими 16-битными кодами. См. подробности ниже.

Функции	Код		Описание
	Старший 16-битный	Младший 16-битный	
Оставляет нулевую точку в отрицательном направлении и потом останавливается (режим 0)	0	0	Когда инструкция включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой. Когда нулевая точка переходит в состояние ON (состояние сигнала нулевой точки меняется с OFF на ON), частота понижается до частоты JOG, продолжается движение в отрицательном направлении и не прекращается, пока состояние сигнала нулевой точки не изменится с ON на OFF.
Оставляет нулевую точку в положительном направлении и потом останавливается (режим 1)	0	1	Когда инструкция включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой. Когда нулевая точка переходит в состояние ON (состояние сигнала нулевой точки меняется с OFF на ON), частота резко понижается до нуля, движение идет в положительном направлении с частотой JOG и не прекращается, пока состояние сигнала нулевой точки не изменится с ON на OFF.
Режим 0 Снова движется после возврата в нулевую точку	Количество импульсов для движения	2	Операция по возврату в нулевую точку аналогична операции для младшего 16-битного кода. Когда сигнал нулевой точки переходит в состояние ON, движение продолжается в соответствии с заданным количеством импульсов. Когда старший 16-битный код становится положительным числом, поиск начинается в положительном направлении. Отрицательное значение означает, что поиск идет в отрицательном направлении.
Режим 0 Поиск фазы Z после возврата в нулевую точку (вход фазы Z задан в HWCONFIG)	Количество фаз Z	3	Операция по возврату в нулевую точку аналогична операции для младшего 16-битного кода. Когда сигнал нулевой точки возвращается, движение продолжается в соответствии с количеством фаз Z. Когда старший 16-битный код становится положительным числом, поиск начинается в положительном направлении. Отрицательное значение означает, что поиск идет в отрицательном направлении. Предположим, что передний фронт импульса на входе X0.1 задан как условие входа фазы Z в HWCONFIG, тогда отсчет будет выполняться каждый раз по переднему фронту импульса по поступлению его на вход X0.1.
Режим 0 Выдача сигнала сброса после возврата в нулевую точку. (дискретный выход сброса Z задан в HWCONFIG)	Количество импульсов для движения или количество фаз Z	4+0=4 4+1=5 4+2=6 4+3=7 (бит2=ON)	Значения 4~7 соответствуют кодам 0~3, и заданный выход передает включающий сигнал (ON) длительностью примерно 20 мс по завершении выполнения функции. Диапазон выходов от Y0.12 до Y0.15 и от Y1.0 до Y1.15. Например, если в HWCONFIG в качестве выхода задан Y1.12, значит выход Y1.12 будет выдавать сигналы сброса.
Оставляет нулевую точку в положительном направлении и потом останавливается	0	8+0=8 8+1=9 (бит3=ON)	Операция возврата в нулевую точку аналогична приведенной для кода 1 (режим 1)

(режим 1)			
Режим 1 Выдается серия импульсов после возврата в нулевую точку	Количество импульсов для движения	8+2=10 (бит3=ON)	Операция возврата в нулевую точку аналогична приведенной для младшего 16-битного кода 1. Когда сигнал нулевой точки возвращается, движение продолжается в соответствии с заданным количеством импульсов. Если значение старшего 16-битного кода положительное, движение будет в положительном направлении. Если значение отрицательное, движение будет в отрицательном направлении.
Режим 1 Поиск фазы Z после возврата в нулевую точку (вход фазы Z задан в HWCONFIG)	Количество фаз Z	8+3=11 (бит3=ON)	Операция возврата в нулевую точку аналогична приведенной для младшего 16-битного кода 1. Когда сигнал нулевой точки возвращается, движение продолжается в соответствии с искомым количеством фаз Z. Если значение старшего 16-битного кода положительное, движение будет в положительном направлении. Если значение отрицательное, движение будет в отрицательном направлении. Предположим, что передний фронт импульса на входе X0.1 задан как условие входа фазы Z, тогда отсчет будет выполняться каждый раз по переднему фронту импульса по поступлению его на вход X0.1.
Режим 1 Выдача сигнала сброса после возврата в нулевую точку. (выход сброса Z задан в HWCONFIG)	0 или количество импульсов или количество фаз Z	12~15 (бит3=бит2=ON)	После возврата нулевой точки в режиме 1, выдается сигнал сброса длительностью 20 мс.
Точка DOG – это точка В		+16 (бит4=ON)	Если у младшего 16-битного кода бит 4 имеет состояние ON, это значит, что появляется сигнал нулевой точки (ON), когда точка DOG меняет свое состояние с ON на OFF, а когда состояние меняется с OFF на ON, нулевая точка остается как точка DOG.

5. Последовательность выполнения будет следующей в зависимости от значения младшего 16-битного кода в таблице выше:

Сигнал DOG определяется значением бит4 → Режим 0, или режим 1 для возврата нулевой точки выбран в зависимости от значения бита 3. → Операция возврата в нулевую точку выполняется в соответствии со значениями бита 1 и бита 0 → Сигнал сброса выдается в соответствии с битом 2.

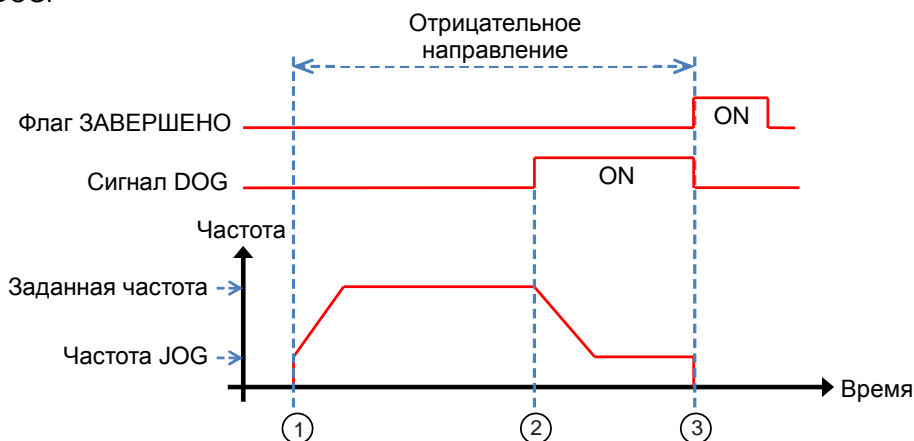
6. Задайте вход и условие срабатывания по переднему фронту импульса в HWCONFIG, когда системе позиционирования требуются входы концевых выключателей в положительном и отрицательном направлении. Обратите внимание, что входы концевых выключателей не должны быть такими же, что нулевая точка и вход фазы Z.
7. Флаг ЗАВЕРШЕНО переходит в состояние ON, когда инструкция завершает выполнение своей функции. Например, если выбран функциональный код 6, ПЛК ставит флаг ЗАВЕРШЕНО только по завершении поиска фазы Z.



8. При работе инструкции DZRN: если задан вход для нулевой точки такой же, что и для внешнего прерывания в программе, обработка прерываний не будет выполняться, пока инструкция DZRN не выключится.
9. Если концевой выключатель запускается под управлением HWCONFIG и есть обработка внешних прерываний, программа прерываний будет исполняться вместе.
10. Шаги выполнения функций:

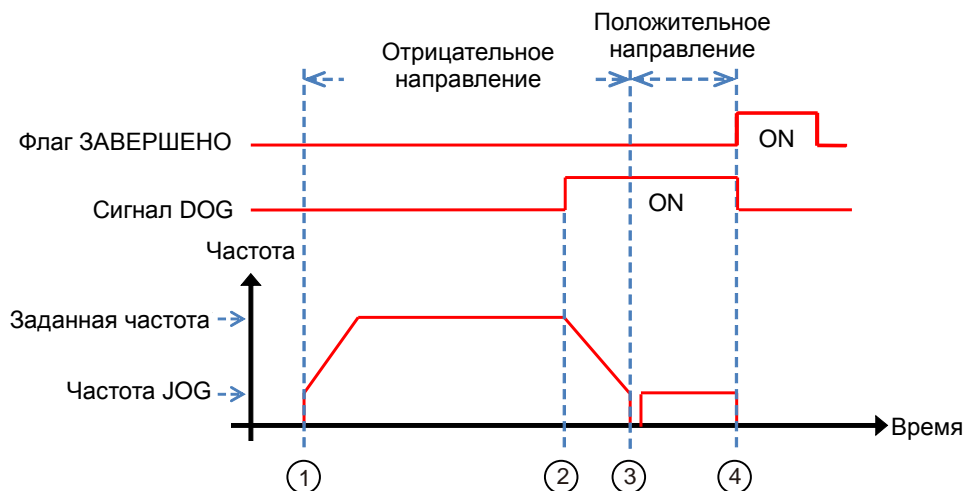
Функциональный код 0:

- ① Когда инструкция DZRN включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой.
- ② После получения сигнала DOG частота выхода понижается до частоты JOG. Выход продолжает работать в отрицательном направлении и не прекращает это делать, пока состояние сигнала нулевой точки не изменится с ON на OFF.
- ③ Выход прекращает выдавать импульсы, когда состояние сигнала меняется с ON на OFF после снятия сигнала DOG.



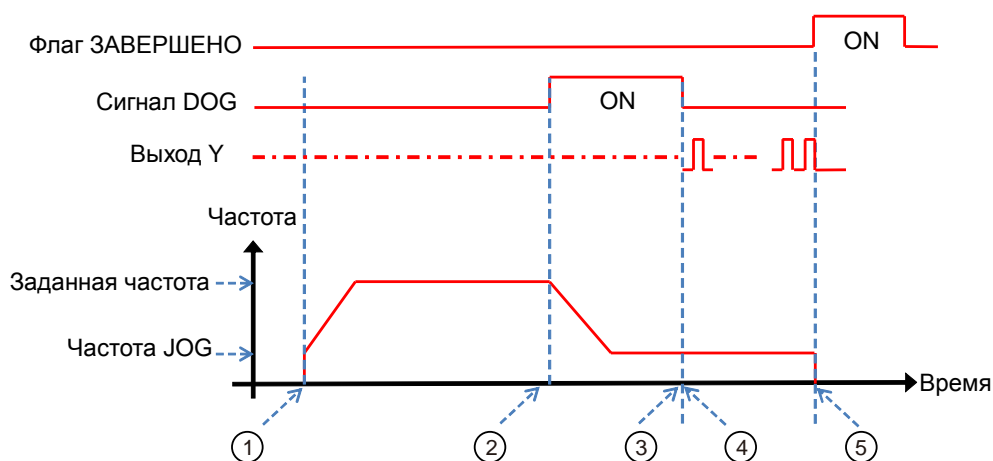
Функциональный код 1:

- ① Когда инструкция DZRN включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой.
- ② Если получен сигнал DOG, выход работает в положительном направлении с частотой JOG после понижения частоты выхода и смены направления движения. Выход не прекращает выдавать импульсы, пока состояние сигнала нулевой точки не сменится с ON на OFF.
- ③ Сигнал DOG снимается, и ПЛК останавливается, когда состояние сигнала меняется с ON на OFF.



Функциональный код 2:

- ① Когда инструкция DZRN включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой.
- ② После получения сигнала DOG выход повышает частоту до частоты JOG и продолжает работать в отрицательном направлении.
- ③ Когда сигнал DOG снимается и сигнал переходит из состояния ON в OFF, выдается заданное количество импульсов.
- ④ Идет первый импульс.
- ⑤ По завершении выдачи 100-го импульса ПЛК останавливается и ставит флаг ЗАВЕРШЕНО.

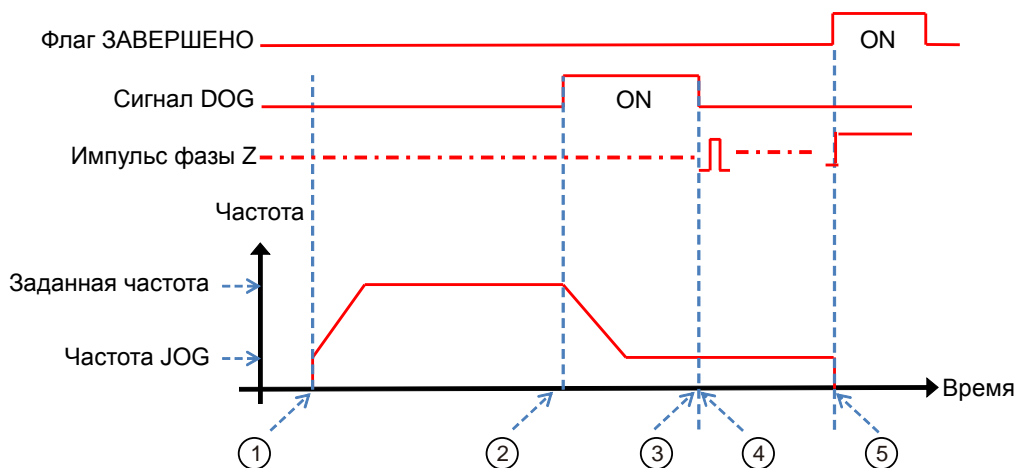


Функциональный код 3:

- ① Когда инструкция DZRN включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой.
- ② После получения сигнала DOG выход понижает частоту до частоты JOG, и движение продолжается в отрицательном направлении.
- ③ После снятия сигнала DOG и когда состояние сигнала меняется с ON на OFF, движение производится в

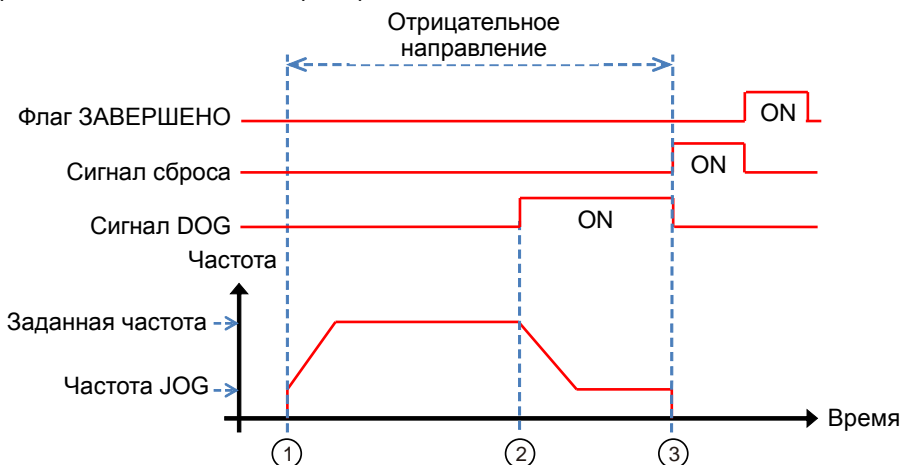
соответствии с количеством фаз Z.

- ④ Импульс первой фазы Z
- ⑤ Движение прекращается после завершения 2-й фазы Z, и после этого ставится флаг ЗАВЕРШЕНО.



Функциональный код 4:

- ① Когда инструкция DZRN включена, поиск нулевой точки ведется в отрицательном направлении с заданной частотой.
- ② После получения сигнала DOG выход понижает частоту до частоты JOG и продолжает работать в отрицательном направлении. Выход не прекращает выдавать импульсы, пока состояние сигнала нулевой точки не сменится с ON на OFF.
- ③ После снятия сигнала DOG выход останавливается, когда состояние сигнала меняется с ON на OFF и сигнал сброса выдается в течение примерно 20 мс.



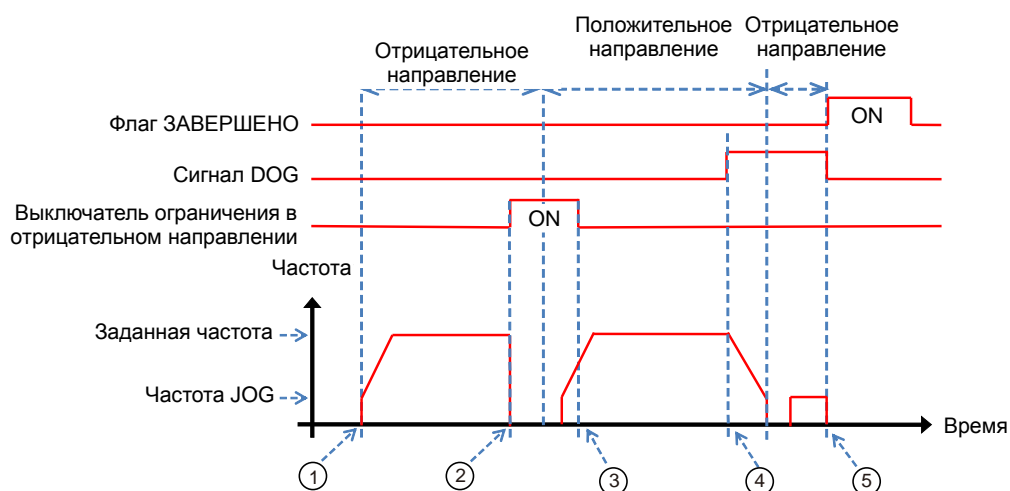
Функциональный код 0+ функция ограничения в отрицательном направлении включена:

Вход для конечного выключателя в отрицательном направлении задан в HWCONFIG и затем загружен в память ПЛК (ПЛК автоматически оценивает функцию ограничения в отрицательном направлении после включения инструкции).

- ① Когда инструкция DZRN включена, поиск нулевой точки ведется в отрицательном направлении с

заданной частотой.

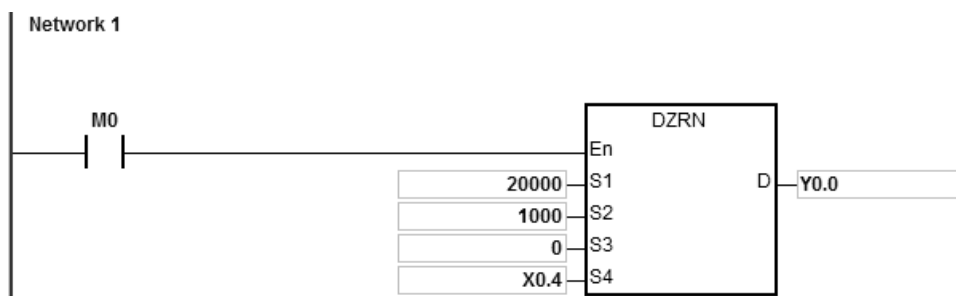
- ② Когда конечный выключатель в отрицательном направлении переходит в состояние ON, движение прекращается и начинается в положительном направлении после смены направления.
- ③ Движение продолжается в положительном направлении после выключателя ограничения в отрицательном направлении.
- ④ Частота выхода линейно понижается после получения сигнала DOG. После смены направления происходит реверсирование выхода с частотой JOG.
- ⑤ Выход прекращает выдавать импульсы, когда состояние сигнала меняется с ON на OFF после снятия сигнала DOG.



**Пример 1:**

Если M0 = ON, начинается возврат в нулевое положение путем выдачи импульсов с выхода Y0.0 с частотой 20 кГц.

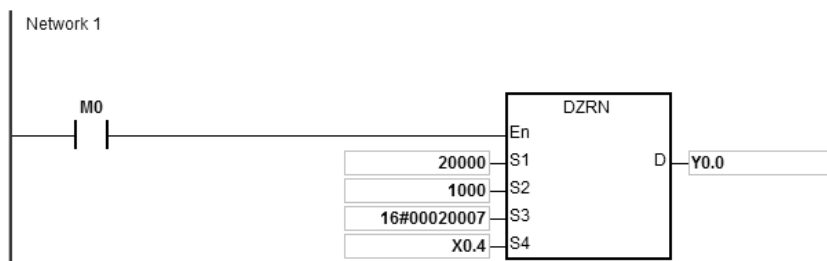
При получении сигнала DOG и переходе входа X0.4 в состояние ON, выход Y0.0 начинает выдавать импульсы JOG с частотой 1 кГц. Выход не прекращает выдавать импульсы, пока состояние входа X0.4 не сменится с ON на OFF.



**Пример 2:**

Если M0 = ON, начинается возврат в нулевое положение путем выдачи импульсов с выхода Y0.0 с частотой 20 кГц.

Если получен сигнал DOG и вход X0.4 перешел в состояние ON, ПЛК понижает частоту до частоты JOG, равной 1 кГц, и выходные импульсы выдаются с частотой JOG, равной 1 кГц. Если X0.4 = OFF, ПЛК начинает искать импульс фазы Z. Если на вход X0.5 поступает два импульса, ПЛК останавливается и выход Y1.4 выдает импульс длительностью 20 мс.

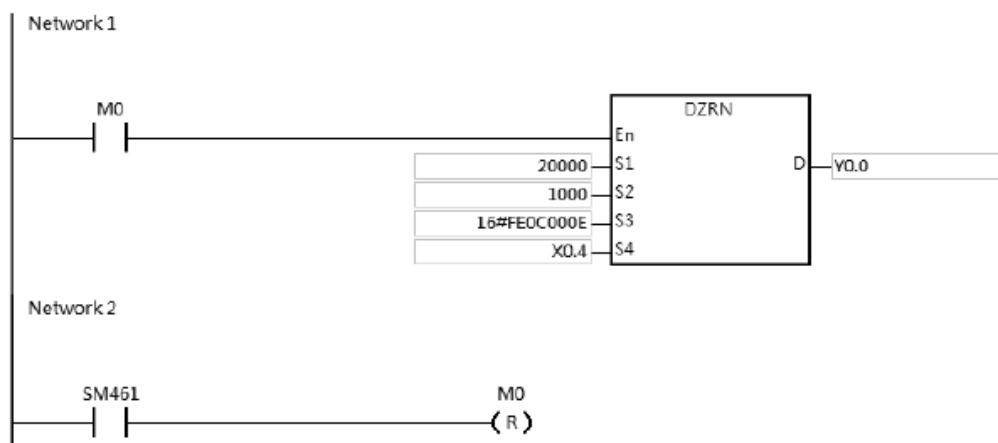


Описание:

1. Если передний фронт импульса на входе X0.5 задан как условие входа фазы Z в HWCONFIG, тогда отсчет будет выполняться каждый раз по переднему фронту импульса по поступлению его на вход X0.5.
2. Y1.4 задан как дискретный выход для выдачи сигнала сброса в HWCONFIG.
3. Операнд S<sub>3</sub> задан равным 16#00020007. Значение старшего 16-битного кода, равное 02, означает поиск фазы Z дважды. Значение младшего 16-битного кода, равное 07, означает выдачу сигнала сброса (длительностью 20 мс) после нахождения фазы Z.

### Пример 3:

Когда M0 включен, происходит вывод импульса с Y0.0 с частотой 20 кГц для поиска нулевой точки в обратном направлении. Когда принимается сигнал DOG и X0.4 включен, ПЛК уменьшает частоту до частоты JOG 1 кГц и продолжает двигаться в прямом направлении с частотой JOG 1 кГц. Когда X0.4 выключен, после завершения вывода 500 импульсов начинается движение в обратном направлении. Y1.4 выдает импульс шириной 20 мс и затем останавливает вывод.



Описание:

1. Y1.4 задан как дискретный выход для выдачи сигнала сброса в HWCONFIG.
2. S<sub>3</sub> задается как 16#FE0C0007. Старшие 16 бит [FE0C = -500]: как только ось движется от сигнала DOG.

Управляемая ось начинает движение после завершения вывода 500 импульсов в обратном направлении.  
Младшие 16 бит [000E]: режим 14; когда принимается сигнал DOG, управляемая ось движется в прямом направлении; после того, как ось движется от сигнала DOG, выводится сигнал сброса (20 мс).

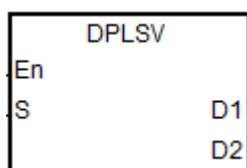
API	Инструкция			Операнд								Описание			
2705	D	PLSV		S, D <sub>1</sub> , D <sub>2</sub>								Регулируемый импульсный выход			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S							●	●	●		○		○	○		
D <sub>1</sub>		○														
D <sub>2</sub>		○	○													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S			●				●						
D <sub>1</sub>	●												
D <sub>2</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

Символьное обозначение:



- S** : Частота выходных импульсов
- D<sub>1</sub>** : Импульсный выход
- D<sub>2</sub>** : Выход задания направления вращения

Описание:

- Операнд **S<sub>1</sub>** определяет частоту выходных импульсов. Диапазон значений от -4 МГц до +4 МГц для моделей с дифференциальным выходом и от -200 кГц до +200 кГц для моделей с выходом типа открытый коллектор. Знаки минус и плюс показывают отрицательное и положительное направления. Частота выходных импульсов может меняться во время выдачи импульсов. Но если новая частота отличается от предыдущей по направлению, инструкция останавливает выдачу импульсов, и через один цикл импульсы будут выдаваться снова на заданной частоте.
- Подробнее см. следующую таблицу, где рассмотрены дискретные выходы операндов **D<sub>1</sub>** и **D<sub>2</sub>**. Если режим выхода операндов **D<sub>1</sub>** и **D<sub>2</sub>** не является выходом по умолчанию «импульс + направление» (0 – значение по умолчанию), измените режим, выставив специальный регистр (SR) равным 1. И режим сменится на выход фазы A/B.

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Дискретный выход для <b>D<sub>1</sub></b>	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10

Дискретный выход направления для <b>D<sub>2</sub></b>	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Режим выхода	SR462	SR482	SR502	SR522	SR542	SR562
Флаг ЗАНЯТО	SM460	SM480	SM500	SM520	SM540	SM560
Регистр текущего состояния выхода	SR460 SR461	SR480 SR481	SR500 SR501	SR520 SR521	SR540 SR541	SR560 SR561

3. На выход направления вращения, заданный операндом **D<sub>2</sub>**, влияет цикл сканирования, если это не выход, рекомендованный к использованию.
4. Выход направления импульсов, заданный операндом **D<sub>2</sub>**, меняет свое состояние в зависимости от знака минус или плюс операнда **S**. Операнд **D<sub>2</sub>** имеет состояние OFF, если у операнда **S** знак плюс (+), а если у операнда **S** знак минус (-), тогда операнд **D<sub>2</sub>** имеет состояние ON.
5. В данной инструкции линейное понижение/повышение не предусмотрено. Поэтому линейное повышение частоты в начале и линейное понижение частоты до остановки не выполняется. Если нужно линейное повышение/понижение частоты, см. инструкцию API 0703 DRAMP, которая может повышать и понижать частоту выходных импульсов.
6. Если производится выдача импульсов по данной инструкции и состояние привода меняется на OFF, выход немедленно выключается.
7. Когда выходы направления не используют выходы по умолчанию для вывода, вы можете обратиться к приведенной ниже таблице, чтобы установить время вывода, которое выводит импульс первым, прежде чем выводится направление. Таким образом, вы можете быть уверены, что вывод происходит только при переключении на выходы направления. Единица измерения SR составляет миллисекунду, а диапазон составляет 0-20 мс. Значение по умолчанию равно 0 указывает на неактивность.

Выход <b>D<sub>1</sub></b>	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Номер SR	SR640	SR641	SR642	SR643	SR644	SR645
Выход <b>D<sub>1</sub></b>	Y0.6	Y0.7	Y0.8	Y0.9	Y1.0	Y1.1
Номер SR	SR646	SR647	SR648	SR649	SR650	SR651

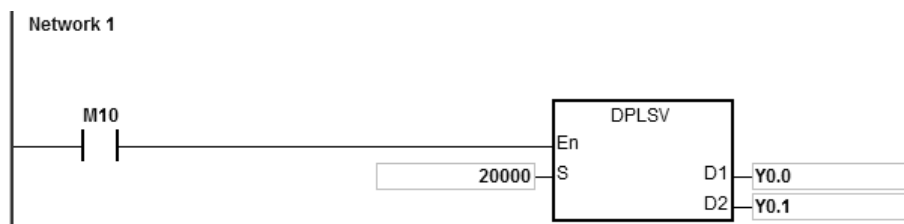
Примечание: источник времени здесь - от времени сканирования ПЛК. (доступно для версии прошивки 1.02.20 или выше)

8. См пример 3 инструкции DJOG (API 2703) для программирования на языке ST.

**Пример:**

Если M10 = ON, выход Y0.0 подает импульсы с частотой 20 кГц. Если выход Y0.1 = OFF, это означает положительное направление выходных импульсов.





API	Инструкция			Операнд								Описание				
2706	D	DRVI		$S_1, S_2, D_1, D_2$								Позиционирование в относительных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$D_1$		○														
$D_2$		○	○													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**

DDRVI	
$E_n$	
$S_1$	$D_1$
$S_2$	$D_2$

- $S_1$  : Количество выходных импульсов (относительное позиционирование)
- $S_2$  : Частота выходных импульсов
- $D_1$  : Импульсный выход
- $D_2$  : Выход направления импульсов

**Описание:**

- Операнд  $S_1$  задает количество выходных импульсов (относительное позиционирование). Диапазон значений: -2,147,483,648~+2,147,483,647. Знак «+/-» показывает положительное и отрицательное направление.
- Операнд  $S_2$  определяет частоту выходных импульсов. Допустимый диапазон значений: 1 Гц ~ +200 кГц для моделей с выходом типа открытый коллектор и 1 Гц ~ +4 МГц для моделей с дифференциальным выходом.
- Подробнее см. следующую таблицу, где рассмотрены дискретные выходы операндов  $D_1$  и  $D_2$ . Если режим выхода операндов  $D_1$  и  $D_2$  не является выходом по умолчанию «импульс + направление» (0 – значение по умолчанию), измените режим, выставив специальный регистр (SR) равным 1. И режим сменится на выход фазы A/B.

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Дискретный выход для $D_1$	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Дискретный выход направления для $D_2$	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Режим выхода	SR462	SR482	SR502	SR522	SR542	SR562

DDRVI и DDRVA поддерживают дискретные выходы, задаваемые операндом **D<sub>1</sub>**, которыми могут быть нечетные входы, как показано в следующей таблице. Поддерживается только режим выхода «импульс + направление». Дискретные выходы Y0.0, Y0.2...Y0.10 не рекомендуется использовать как высокоскоростные входы.

Номер оси	Ось 7	Ось 8	Ось 9	Ось 10	Ось 11	Ось 12
Дискретный выход для <b>D<sub>1</sub></b>	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Дискретный выход направления для <b>D<sub>2</sub></b>	Y1.0 или больше или любой регистр M (Bool)					

- Если выход направления импульсов, заданный операндом **D<sub>2</sub>**, не является выходом, рекомендованным к использованию в таблице выше, на выход будет влиять время сканирования.
- Если у операнда **D<sub>2</sub>** режим «импульс + направление», а у операнда **S<sub>1</sub>** знак минус, тогда **D<sub>2</sub> = ON**. Если у операнда **S<sub>1</sub>** знак плюс, тогда **D<sub>2</sub> = OFF**. Операнд **D<sub>2</sub>** не сразу получает состояние OFF после завершения выдачи импульсов отрицательного направления. Операнд **D<sub>2</sub>** переходит в состояние OFF, когда контакт выполнения инструкции переходит в состояние OFF.
- Для выполнения инструкции необходимы такие параметры, как начальная/конечная частота и время линейного повышения/понижения частоты. Соответствие осей регистрам SM/SR см. в следующей таблице.

Если у операнда **D<sub>1</sub>** в качестве выхода выбран четный выход.

Выход для <b>D<sub>1</sub></b>	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Флаг ЗАНЯТО	SM460	SM480	SM500	SM520	SM540	SM560
Флаг ЗАВЕРШЕНО	SM461	SM481	SM501	SM521	SM541	SM561
Смена направления вращения	SM462	SM482	SM502	SM522	SM542	SM562
Флаг СТОП	SM463	SM483	SM503	SM523	SM543	SM563
Включение повышения/понижения частоты по S-образной кривой	SM468	SM488	SM508	SM528	SM548	SM568
Включение линейного повышения/понижения частоты	SM469	SM489	SM509	SM529	SM549	SM569
Автоматический сброс завершения выдачи импульсов	SM470	SM490	SM510	SM530	SM550	SM570
Выполнение программы прерывания по завершении выдачи импульсов	SM471	SM491	SM511	SM531	SM551	SM571
Соответствующий номер прерывания I	I500	I501	I502	I503	I504	I505

Выход немедленно выключается после выключения или остановки выполнения инструкции	SM476	SM496	SM516	SM536	SM556	SM576
Текущее состояние выхода	SR460 SR461	SR480 SR481	SR500 SR501	SR520 SR521	SR540 SR541	SR560 SR561
Начальная/конечная частота (Гц)	SR463	SR483	SR503	SR523	SR543	SR563
Время повышения частоты (мс)	SR464	SR484	SR504	SR524	SR544	SR564
Время понижения частоты (мс)	SR465	SR485	SR505	SR525	SR545	SR565
Заданная частота для прямой линии	SR472 SR473	SR492 SR493	SR512 SR513	SR532 SR533	SR552 SR553	SR572 SR573
Режим S-образной кривой	SR604	SR605	SR606	SR607	SR608	SR609
Текущая частота выхода	SR610 SR611	SR612 SR613	SR614 SR615	SR616 SR617	SR618 SR619	SR620 SR621

Если у операнда **D<sub>1</sub>** в качестве выхода выбран нечетный выход.

Выход для <b>D<sub>1</sub></b>	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Флаг ЗАНЯТО	SM472	SM492	SM512	SM532	SM552	SM572
Флаг ЗАВЕРШЕНО	SM473	SM493	SM513	SM533	SM553	SM573
Флаг СТОП	SM474	SM494	SM514	SM534	SM554	SM574
Автоматический сброс завершения выдачи импульсов	SM475	SM495	SM515	SM535	SM555	SM575
Выход немедленно выключается после выключения или остановки выполнения инструкции	SM477	SM497	SM517	SM537	SM557	SM577
Текущее состояние выхода	SR474 SR475	SR494 SR495	SR514 SR515	SR534 SR535	SR554 SR555	SR574 SR575
Начальная/конечная частота (Гц)	SR476	SR496	SR516	SR536	SR556	SR576
Время повышения/понижения частоты (мс)	SR477	SR497	SR517	SR537	SR557	SR577

Такие флаги, как повышение/понижение частоты по S-образной кривой, линейное повышение/понижение частоты и выполнение программы прерывания по завершении выдачи импульсов, не поддерживаются нечетными дискретными выходами.

Если во время выдачи импульсов инструкция выключается или ее выполнение временно приостанавливается, текущая частота выходных импульсов понижается до полной остановки. Если выход должен выключаться сразу же, необходимо поставить флаг СТОП.

7. См. атрибуты, значения по умолчанию и защелки триггера SM/SR, которым соответствуют оси.

Параметр	Атрибут	Стоп↔Пуск	Пуск↔Стоп	Питание ВКЛ.	По умолчанию	Триггер
Флаг ЗАНЯТО	R	OFF	OFF	OFF	OFF	n
Флаг ЗАВЕРШЕНО	R/W	OFF	OFF	OFF	OFF	n
Флаг СТОП	R/W	OFF	OFF	OFF	OFF	n
Автоматический сброс завершения выдачи импульсов	R/W	OFF	OFF	OFF	OFF	n
Включение повышения/понижения частоты по S-образной кривой	R/W	OFF	OFF	OFF	OFF	n
Включение линейного повышения/понижения частоты	R/W	OFF	OFF	OFF	OFF	n
Выполнение программы прерывания по завершении выдачи импульсов	R/W	OFF	OFF	OFF	OFF	n
Выход немедленно выключается после выключения или остановки выполнения инструкции	R/W	OFF	OFF	OFF	OFF	n
Текущее состояние выхода	R/Wd	Не меняется	Не меняется	Не меняется	0	Y
Начальная/конечная частота (Гц)	R/Wd	Не меняется	Не меняется	Не меняется	200	Y
Время повышения частоты (мс)	R/Wd	Не меняется	Не меняется	Не меняется	200	Y
Время понижения частоты (мс)	R/Wd	Не меняется	Не меняется	Не меняется	200	Y
Заданная частота для прямой линии	R/Wd	Не меняется	Не меняется	Не меняется	0	Y
Режим S-образной кривой	R/Wd	Не меняется	Не меняется	0	0	n
Текущая частота выхода	R	0	0	0	0	n

Примечание: R – регистр чтения. W – регистр записи. Wd – в регистр можно записывать в любое время, кроме работы высокоскоростного выхода.

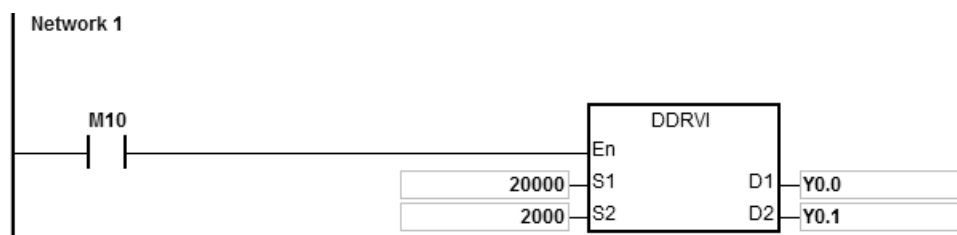
8. После включения функции автоматического сброса после завершения выдачи импульсов флаг должен переходить в состояние OFF автоматически после завершения выдачи импульсов. Поэтому флаг нужно ставить снова, если функция автоматического сброса нужна для следующей операции выдачи импульсов. Рекомендуется всегда менять состояние флага по переднему или заднему фронту импульса. Данная функция, как правило, используется, когда после включения инструкции DDRVI ее не удается отсканировать и выполнить программой ПЛК. Например, инструкция добавлена в программу прерывания.
9. Если флаг ЗАВЕРШЕНО используется для работы функции прерывания I, этот флаг должен автоматически переходить в состояние OFF, когда выдача импульсов завершена и вводится программа прерывания. Поэтому флаг нужно ставить (ON) снова для следующей выдачи импульсов, где будет использоваться прерывание. Рекомендуется всегда менять состояние флага по переднему или заднему фронту импульса.
10. Когда выходы направления не используют выходы по умолчанию для вывода, вы можете обратиться к приведенной ниже таблице, чтобы установить время вывода, которое выводит импульс первым, прежде чем выводится направление. Таким образом, вы можете быть уверены, что вывод происходит только при переключении на выходы направления. Единица измерения SR составляет миллисекунду, а диапазон составляет 0-20 мс. Значение по умолчанию равно 0 указывает на неактивность.

Выход D1	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Номер SR	SR640	SR641	SR642	SR643	SR644	SR645
Выход D1	Y0.6	Y0.7	Y0.8	Y0.9	Y1.0	Y1.1
Номер SR	SR646	SR647	SR648	SR649	SR650	SR651

11. Примечание: источник времени здесь - от времени сканирования ПЛК. (доступно для версии прошивки 1.02.20 или выше)

**Пример 1:**

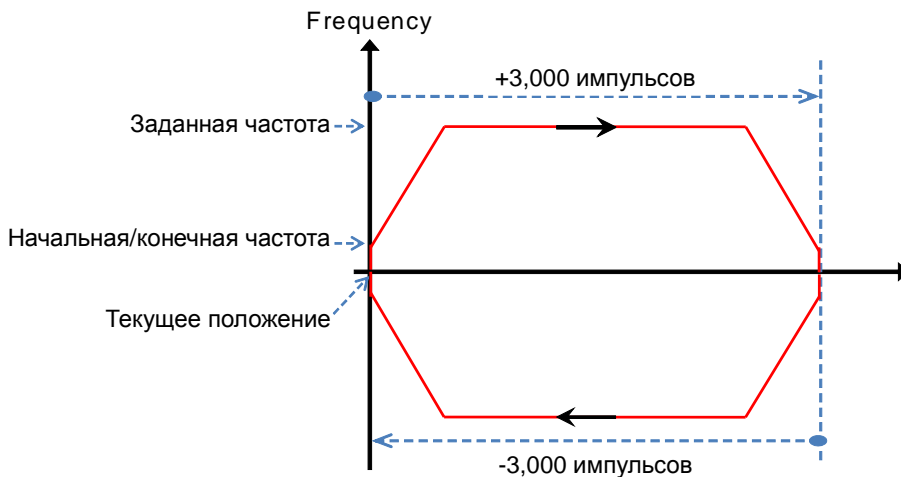
Если M10 = ON, выход Y0.0 выдает 20,000 импульсов с частотой 2 кГц (относительное позиционирование). Y0.1 = OFF, это значит положительное направление.



**Дополнительные замечания:**

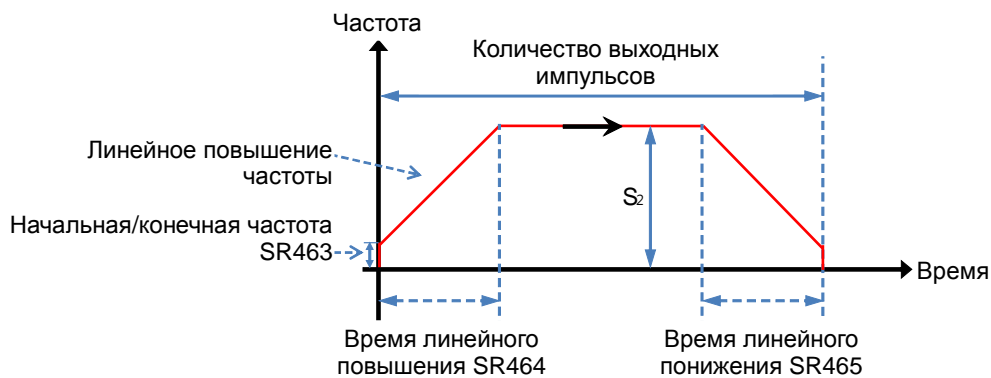
1. Описание относительного позиционирования: укажите дистанцию движения из текущего положения со знаком «+/-».

Частота



2. Настройка повышения/понижения частоты и параметров относительного позиционирования

a) Диаграмма выхода Y0.0

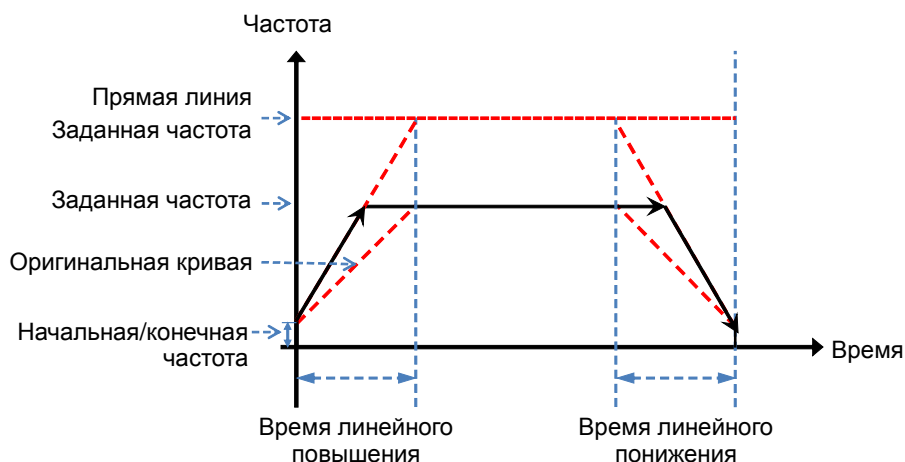


b) Несколько инструкций DRVI в пользовательской программе могут использовать один выход. Во время работы ПЛК может выполняться только одна инструкция. Например, если одна инструкция включает выход Y0.0 для выдачи импульсов, другие инструкции, которые задействуют выход Y0.0 для выдачи импульсов, выполняться не могут. Инструкция, которая запускает выход первым, первым его и использует.

c) После включения инструкции изменение всех других параметров является недопустимым, пока эта инструкция не будет выключена.

3. Линейное повышение и понижение частоты.

a) Диаграмма выхода Y0.0



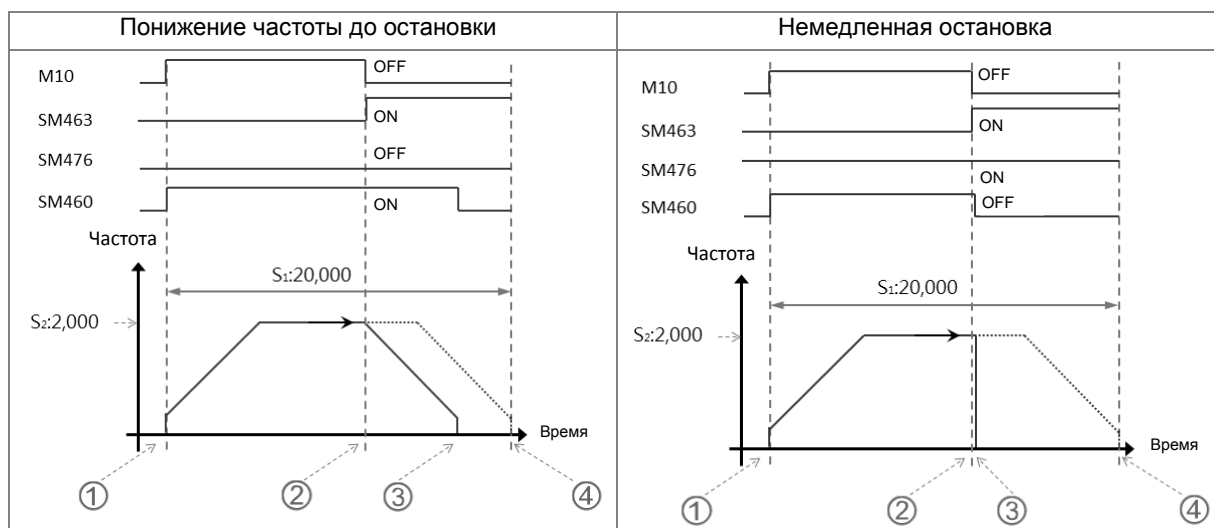
- b) Поставьте флаг линейного изменения частоты SM469 и запишите итоговую частоту, которая должна получиться после изменения, в регистре SR472.
- c) Новая прямая станет вместо исходной, и позиционирование будет выполнено инструкцией DDRVI, как показано на рисунке выше.
- d) Общая кривая определяется начальной/конечной частотой, итоговой частотой и временем повышения/понижения частоты.

Прямая линия определяется начальной/конечной частотой, итоговой частотой и временем повышения/понижения частоты.

- e) Если итоговая частота при линейном изменении частоты меньше итоговой частоты, функция линейного изменения частоты не запустится.

4. Флаг СТОП ставится немедленно

- a) Диаграмма выхода показана на примере 1



- b) Понижение частоты до остановки

① Если M10 = ON, инструкция DDRVI выполняется и начинается выдача импульсов. Флаг SM460



ЗАНЯТО стоит.

② Если инструкция выключена (M10 = OFF) или стоит флаг SM463, состояние флага немедленной остановки SM476 будет подтверждено.

③ Если флаг немедленной остановки SM476 = OFF, частота выхода будет понижаться до остановки, и потом флаг SM460 ЗАНЯТО будет снят.

④ Если M10 не равен OFF и флаг остановки SM463 не стоит после включения инструкции, выход Y0.0 выдает 20,000 импульсов и выключается.

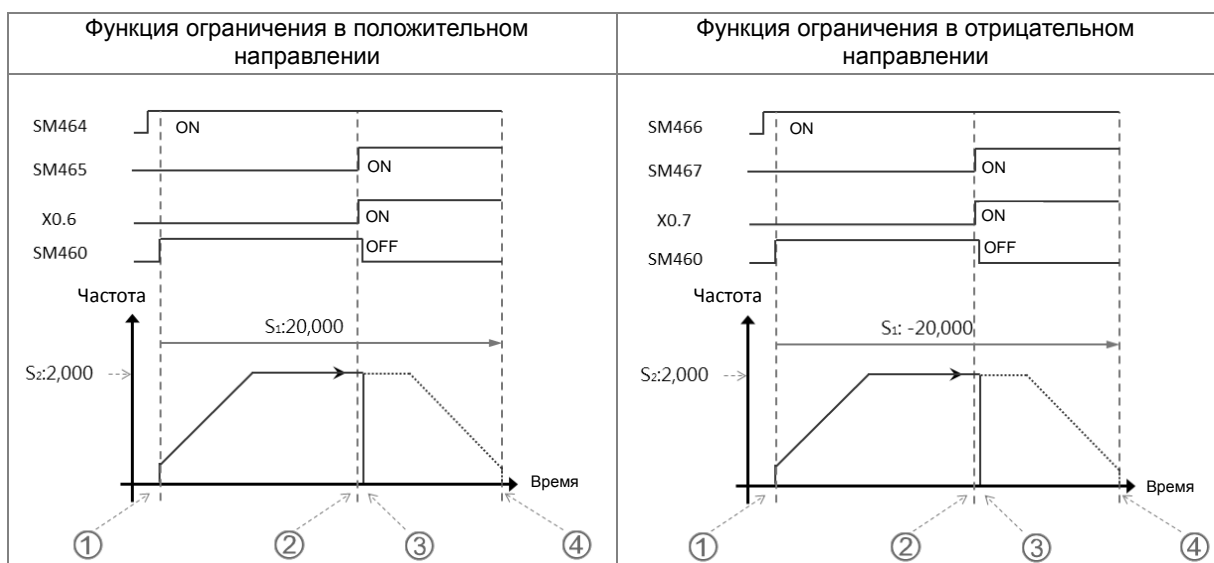
с) Немедленная остановка

② Если инструкция выключена (M10 = OFF) или стоит флаг SM463, состояние флага немедленной остановки SM476 будет подтверждено.

③ Если флаг немедленной остановки SM476 = ON, частота выхода немедленно понижается до остановки и потом флаг SM460 ЗАНЯТО будет снят.

5. Функция аппаратного ограничения

а) Диаграмма работы выхода для оси 1 (Y0.0/Y0.1)



б) Работа ограничения в положительном направлении

① Вход для конечного выключателя в положительном направлении X0.6 задан в HWCONFIG и затем занесен в память ПЛК. Если SM464 = ON, это значит, что функция ограничения в положительном направлении запущена.

① Инструкция DDRVI выдает 20,000 импульсов в положительном направлении. (Прим.: Инструкция не выдает импульсы в положительном направлении, если вход для конечного выключателя в положительном направлении имеет состояние ON в момент запуска функции.)

② Если вход для конечного выключателя в положительном направлении X0.6 меняет свое состояние под действием внешнего механизма и переходит в состояние ON, это значит, что условие функции

аппаратного ограничения в положительном направлении выполнено.

③ Инструкция DDRVI немедленно выключает выход, и регистр тревоги ограничения в положительном направлении SM465 переходит в состояние ON.

④ Если условие аппаратного ограничения не удовлетворено, выход Y0.0 выдает 20,000 импульсов и выключается.

с) Работа ограничения в отрицательном направлении

① Вход для конечного выключателя в отрицательном направлении X0.7 задан в HWCONFIG и затем занесен в память ПЛК. Если SM466 = ON, это значит, что функция ограничения в отрицательном направлении запущена.

② Инструкция DDRVI выдает 20,000 импульсов в отрицательном направлении. (Прим.: Инструкция не выдает импульсы в отрицательном направлении, если вход для конечного выключателя в отрицательном направлении имеет состояние ON в момент запуска функции.)

③ Если вход для конечного выключателя в отрицательном направлении X0.7 меняет свое состояние под действием внешнего механизма и переходит в состояние ON, это значит, что условие функции аппаратного ограничения в отрицательном направлении выполнено.

④ Инструкция DDRVI немедленно выключает выход, и регистр тревоги ограничения в отрицательном направлении SM465 переходит в состояние ON.

⑤ Если условие аппаратного ограничения не удовлетворено, выход Y0.0 выдает -20,000 импульсов и выключается.

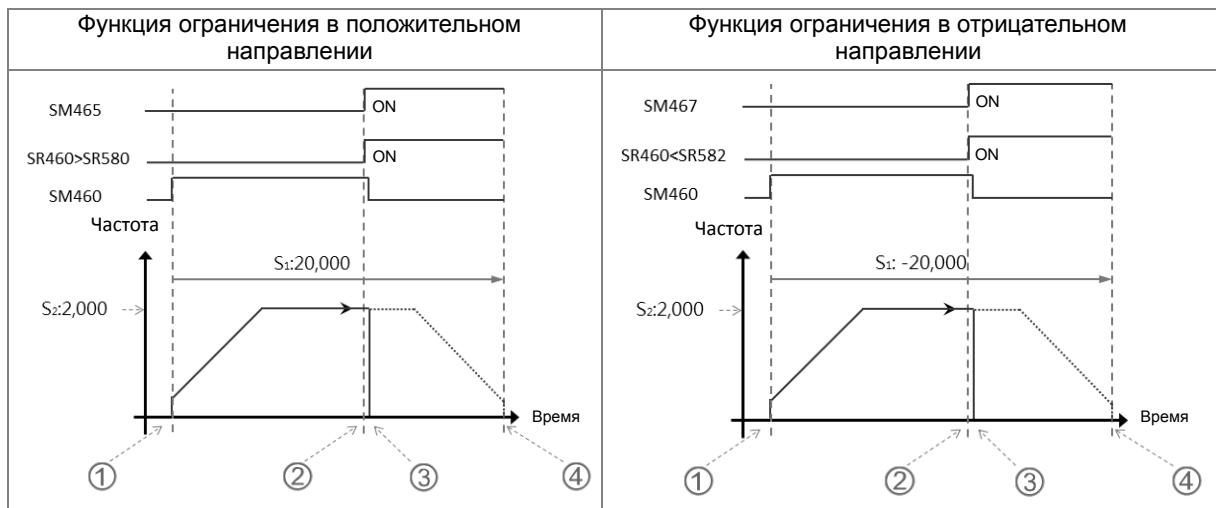
д) Флаг SM относится к функции ограничения

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Вкл. аппаратного ограничения в положительном направлении	SM464	SM486	SM506	SM526	SM546	SM566
Флаг тревоги ограничения в положительном направлении	SM465	SM487	SM507	SM527	SM547	SM567
Вкл. аппаратного ограничения в отрицательном направлении	SM466	SM488	SM508	SM528	SM548	SM568
Флаг тревоги ограничения в отрицательном направлении	SM467	SM489	SM509	SM529	SM549	SM569

Примечание: Флаг тревоги ограничения представляет собой флаг типа только чтение, который ставится и снимается автоматически контроллером в зависимости от состояния флага СТАРТ.

6. Функция программного ограничения

а) Диаграмма работы выхода для оси 1 (Y0.0/Y0.1)



б) Работа ограничения в положительном направлении

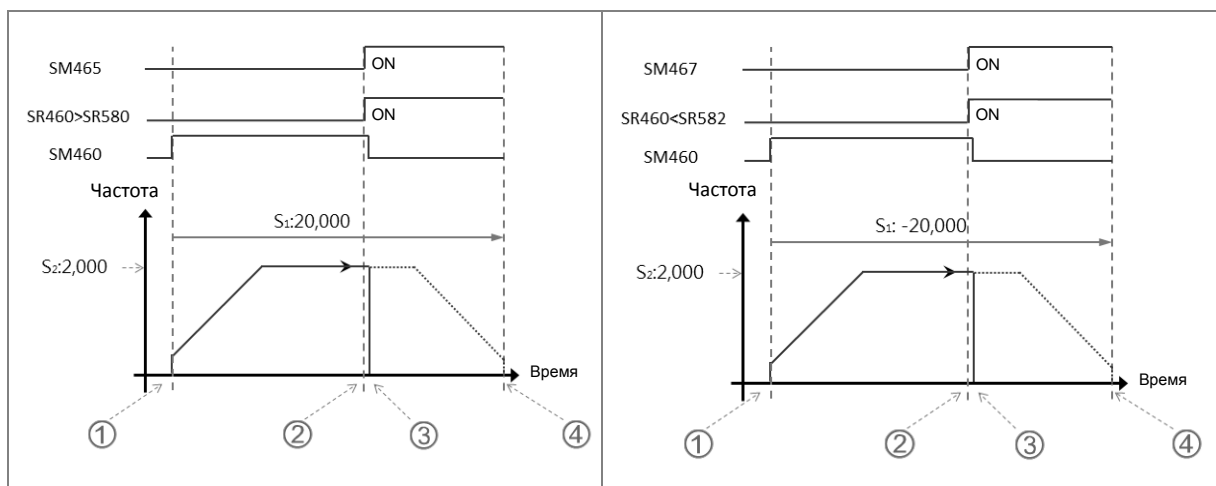
ⓐ Позиция программного ограничения по положительному направлению задана в HWCONFIG (например, SR580=11,000). После загрузки в память ПЛК функция ограничения в положительном направлении запущена.

① Инструкция DDRVI включает выход, и он выдает 20,000 импульсов в положительном направлении. (Прим.: Инструкция не выдает импульсы в положительном направлении, если значение ограничения по положительному направлению превышено в момент запуска функции)

② Если текущая позиция в регистре SR460 > значения ограничения в регистре SR580, значит условие программного ограничения удовлетворено.

③ Инструкция DDRVI немедленно выключает выход, и регистр тревоги ограничения в положительном направлении SM465 переходит в состояние ON.

④ Если условие программного ограничения не удовлетворено, выход Y0.0 выдает 20,000 импульсов и выключается.





а) Работа ограничения в отрицательном направлении

ⓐ Позиция программного ограничения по отрицательному направлению задана в HWCONFIG (например, SR582=-11,000). После загрузки в память ПЛК функция ограничения в отрицательном направлении запущена.

- ① Инструкция DDRVI включает выход, и он выдает 20,000 импульсов в отрицательном направлении. (Прим.: Инструкция не выдает импульсы в отрицательном направлении, если значение ограничения по отрицательному направлению превышено в момент запуска функции.)
- ② Если текущая позиция в регистре SR460 < значения ограничения в регистре SR582, значит условие программного ограничения по отрицательному направлению удовлетворено.
- ③ Инструкция DDRVI немедленно выключает выход, и регистр тревоги ограничения в отрицательном направлении SM465 переходит в состояние ON.
- ④ Если условие программного ограничения не удовлетворено, выход Y0.0 выдает -20,000 импульсов и выключается.

б) Триггер по переднему фронту

Функция предела	Контакт А	Контакт В
Тип	Триггер по переднему фронту	Триггер по заднему фронту
Описание	 <p>При запуске: от меньшей к большей и поддерживается на большей</p>	 <p>При запуске: от большей к меньшей и поддерживается на меньшей</p>

с) Регистры SR, относящиеся к функции ограничения

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Позиция программного ограничения по положительному направлению	SR580	SR584	SR588	SR592	SR596	SR600
	SR581	SR585	SR589	SR593	SR597	SR601
Позиция программного ограничения по отрицательному направлению	SR582	SR586	SR590	SR594	SR598	SR602
	SR583	SR587	SR591	SR595	SR599	SR603

Ограничение выхода по положительному направлению: выход немедленно прекращает выдачу импульсов, если текущая позиция больше значения, заданного функцией ограничения по положительному направлению.

Ограничение выхода по отрицательному направлению: выход немедленно прекращает выдачу импульсов, если текущая позиция меньше значения, заданного функцией ограничения по отрицательному направлению.

Если значения функций ограничения в положительном и отрицательном направлении равны 0, функция программного ограничения не запускается. Поскольку функция программного ограничения проверяет

текущую выходную позицию в момент сканирования инструкции выхода, на выключение выхода повлияет сканирование ПЛК. Чтобы выход быстро выключался в реальном времени, используйте вход для внешнего сигнала ограничения.

7. **Пример ST:**

```

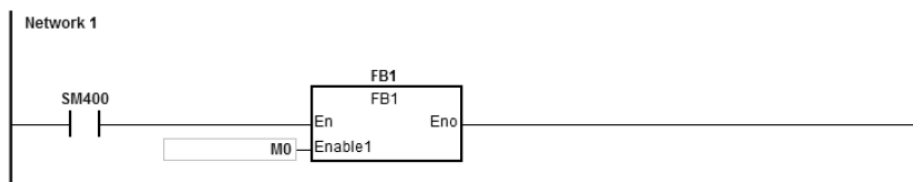
0001 IF M0 THEN
0002 DDRVI (1000, 1000, Y0.0, Y0.1) ;
0003 ELSIF SM461 THEN
0004 SM470:=TRUE;
0005 END_IF;
0006
0007
    
```

Описание

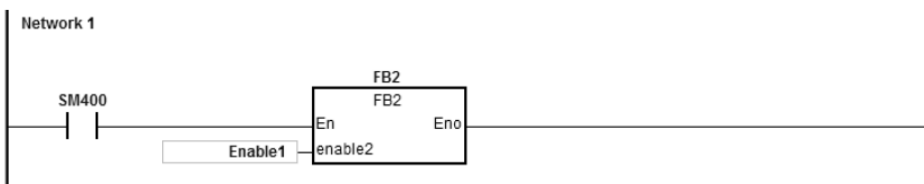
1. Если M0 = ON, выход Y0.0 подает 1000 импульсов с частотой 1 кГц.
2. Когда все импульсы выданы, SM461 = ON и потом SM470= ON.
3. Когда состояние M0 снова меняется с OFF на ON, выдача импульсов начинается заново.

8. **Пример функциональных блоков:**

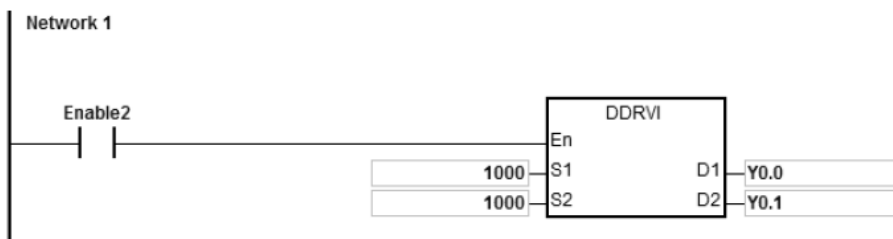
Основная программа:



Функциональный блок 1:



Функциональный блок 2:



**Описание:**

Когда M0 включен, Y0.0 выдает 10000 импульсов на 1 кГц. Y0.1 выключен, это означает, что направление импульсного выхода прямое.

9. Обратитесь к инструкции PLAN (API2701) для получения более подробной информации об изменении заданной частоты во время вывода.

10. Изменение заданной позиции во время вывода:

10.1 Эта функция доступна только для версии прошивки 1.02.30 и выше. Вы не можете изменить направление вывода даже после изменения заданной позиции. Например, если исходная заданная позиция находится в прямом направлении, после того, как заданная позиция будет изменена, направление останется прямым.

10.2 После установки флага изменения, изменение происходит при сканировании инструкции. После изменения флаг будет сброшен. Но если заданное положение установлено неверно, флаг изменения остается включенным.

10.3 Обратитесь к следующей таблице для описания соответствующих выходов и флагов SM.

Выход	Y0.0	Y0.1	Y0.2	Y0.3	Y0.4	Y0.5
Соответствующий флаг	SM478	SM479	SM498	SM499	SM518	SM519
Выход	Y0.6	Y0.7	Y0.8	Y0.9	Y1.0	Y1.1
Соответствующий флаг	SM538	SM539	SM558	SM559	SM578	SM579

10.4 Функция изменения заданной позиции доступна для инструкций DDRVI и DDRVA. Когда используется инструкция DDRVI и установлен флаг изменения, изменение будет выполнено. Даже если номер выхода равен 0, операция будет выполнена и выход будет остановлен. Когда используется инструкция DDRVA, если измененная заданная позиция совпадает с предыдущей, изменение не будет выполнено.

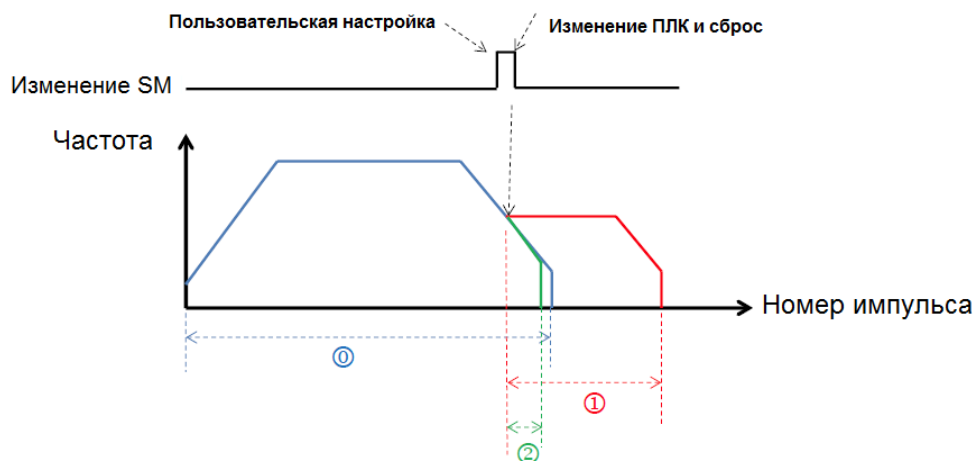
10.5 При изменении заданного положения во время работы выхода возможны две ситуации.

Ситуация А: изменение происходит во время вывода по заднему фронту.

① Исходная заданная позиция при старте (голубая линия)

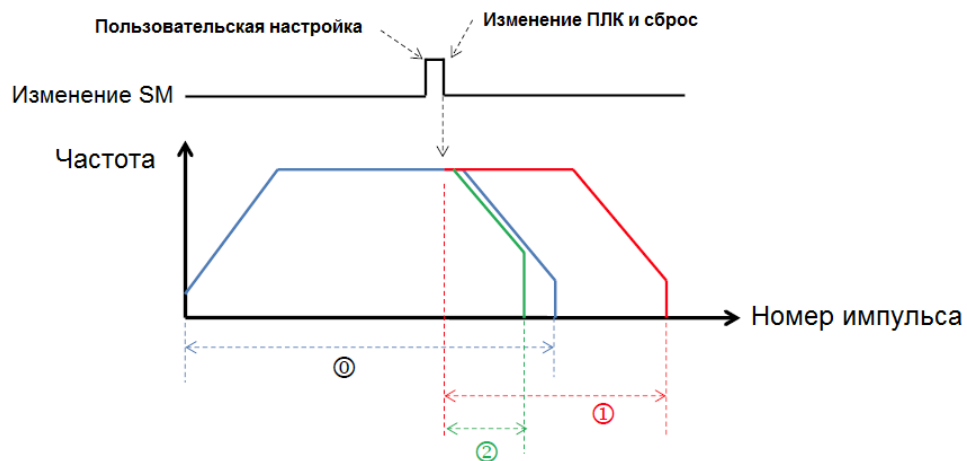
① Новый номер выходного импульса  $\geq$  исходного номера выходного импульса, выходная кривая – красная линия.

② Новый номер выходного импульса  $<$  исходного номера выходного импульса, выходная кривая – зеленая линия.



Ситуация В: изменение происходит во время вывода в заданной частотной области.

- ① Исходная заданная позиция при старте (голубая линия)
- ① Новый номер выходного импульса  $> =$  номера выходного импульса по заднему фронту, выходная кривая – красная линия.
- ② Новый номер выходного импульса  $<$  номера выходного импульса по заднему фронту, выходная кривая – зеленая линия.



API	Инструкция			Операнд								Описание				
2707	D	DRVA		<b>S<sub>1</sub>, S<sub>2</sub>, D<sub>1</sub>, D<sub>2</sub></b>								Позиционирование в абсолютных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>							●	●	●		○		○	○		
<b>S<sub>2</sub></b>							●	●	●		○		○	○		
<b>D<sub>1</sub></b>		○														
<b>D<sub>2</sub></b>		○	○													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>			●				●						
<b>S<sub>2</sub></b>			●				●						
<b>D<sub>1</sub></b>	●												
<b>D<sub>2</sub></b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**

DDRVA	
En	
S1	D1
S2	D2

- S<sub>1</sub>** : Количество выходных импульсов (абсолютное позиционирования)
- S<sub>2</sub>** : Частота выходных импульсов
- D<sub>1</sub>** : Импульсный выход
- D<sub>2</sub>** : Выход направления

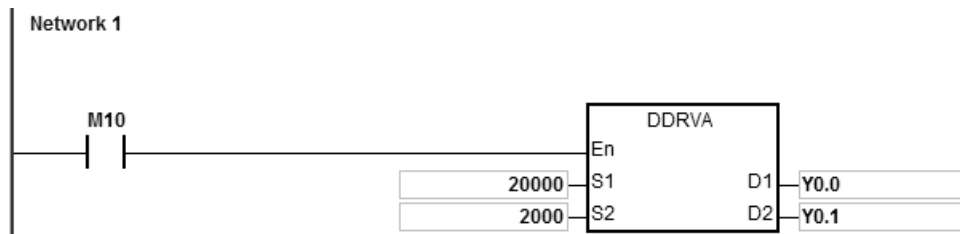
**Описание:**

- Операнд **S<sub>1</sub>** определяет количество выходных импульсов (относительного позиционирования). Диапазон значений: -2,147,483,648~+2,147,483,647. Знак «+/-» показывает положительное и отрицательное направление.
- Операнд **S<sub>2</sub>** определяет частоту выходных импульсов. Допустимый диапазон значений: 1 Гц ~ +200 кГц для моделей с выходом типа открытый коллектор и 1 Гц ~ +4 МГц для моделей с дифференциальным выходом.
- При абсолютном позиционировании выдача импульсов производится с текущего положения и до момента достижения заданного положения. Например, количество импульсов в текущем положении равно 100, а количество импульсов в заданном операндом **S<sub>1</sub>** положении равно 1000. Таким образом, фактическое количество выдаваемых импульсов получается 1000-100=900.
- Подробнее см. описание инструкции DDRVI (API 2706).



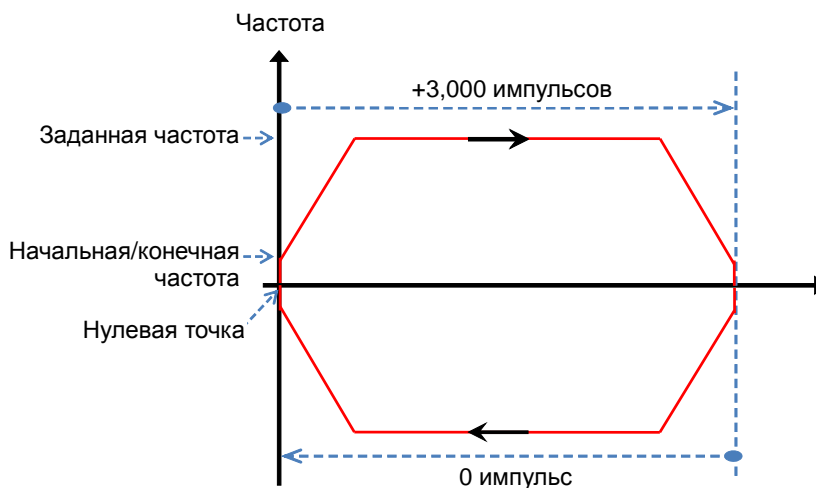
**Пример:**

Если значение регистра текущего положения SR460 (32-бита) равно 100 и M10 = ON, по команде инструкции DDRVA выход Y0.0 начинает выдавать импульсы с частотой 2 кГц, пока значение регистра SR460 не станет равным 20,000 (позиционирование в абсолютных координатах). Y0.1 = OFF, это значит положительное направление.



**Дополнительные замечания:**

1. Позиционирование в абсолютных координатах: способ указания расстояния от центра (нулевой точки).



API	Инструкция		Операнд									Описание				
2708		CSFO	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D</b>									Захват скорости и пропорциональный выход				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>	○															
S <sub>2</sub>								●								
S <sub>3</sub>								●								
S <sub>4</sub>								●								
S <sub>5</sub>								●								
D <sub>1</sub>		○														
D <sub>2</sub>								●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>	●												
S <sub>2</sub>			●				●						
S <sub>3</sub>			●				●						
S <sub>4</sub>		●											
S <sub>5</sub>		●											
D <sub>1</sub>	●												
D <sub>2</sub>			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	-

**Символьное обозначение:**

CSFO	
En	
S1	D1
S2	D2
S3	
S4	
S5	

- S<sub>1</sub> : Регистр входного сигнала
- S<sub>2</sub> : Количество входных импульсов
- S<sub>3</sub> : Частота входных импульсов
- S<sub>4</sub> : Установка знаменателя (частоты на выходе) в пропорции частоты на входе и частоты на выходе
- S<sub>5</sub> : Установка числителя (частоты на входе) в пропорции частоты на входе и частоты на выходе
- D<sub>1</sub> : Импульсный выход
- D<sub>2</sub> : Частота выходных импульсов

**Описание:**

- В операнде **S<sub>1</sub>** можно задавать в качестве входов только X0.0, X0.2, X0.4, X0.6, X0.8 и X0.10. Операнд **S<sub>1</sub>** занимает два входа подряд. Инструкция не будет выполняться, если входы отличаются от указанных выше. После задания входов высокоскоростной счетчик будет выбран автоматически. Если в этой программе так же есть инструкция DCNT, использующая такой же высокоскоростной счетчик, контроллер первой выполнит

инструкцию, которая первой запускает счетчик. Входы и соответствующие высокоскоростные счетчики приведены в следующей таблице.

Номер группы	1	2	3	4	5	6
Вход $S_{1+0}$ (фаза А)	X0.0	X0.2	X0.4	X0.6	X0.8	X0.10
Вход $S_{1+1}$ (фаза В)	X0.1	X0.3	X0.5	X0.7	X0.9	X0.11
Количество высокоскоростных счетчиков	HC202	HC206	HC210	HC214	HC218	HC222
Флаг смены направления входа	SM270	SM271	SM272	SM273	SM274	SM275

- Если высокоскоростные счетчики данной инструкции могут использовать только режим входа А/В, необходимо поставить флаг смены направления входа в момент, когда MPG подсоединен, но еще не вращается, а вход ПЛК в состоянии ON.
- Операнд  $S_2$  определяет количество входных импульсов. Данный параметр объявляется 32-битной переменной.
- Операнд  $S_3$  определяет частоту входных импульсов. Данный параметр объявляется 32-битной переменной с дискретностью 1 Гц.
- Операнд  $S_4$  устанавливает знаменатель (частоту на выходе) в пропорции частоты на входе и частоты на выходе. Операнд  $S_5$  устанавливает числитель (частоту на входе) в пропорции частоты на входе и частоты на выходе. У операндов  $S_4$  и  $S_5$  диапазон значений от 1 до 255. Если значение вне пределов данного диапазона, в инструкции автоматически выставляется минимальное или максимальное значение. Например, если частота на входе: частота на выходе = 5: 3, знаменатель (частота на выходе) будет К3, а числитель (частота на входе) будет К5. Если частота на входе: частота на выходе = 1: 2, знаменатель (частота на выходе) будет К2, а числитель (частота на входе) будет К1.
- В операнде  $D_1$  можно задавать в качестве выхода Y0.0, Y0.2, Y0.4, Y0.6, Y0.8 и Y0.10.  $D_1$  занимает два дискретных выходов.

Выходы и соответствующие регистры (SR) режима выхода приведены в следующей таблице.

Номер оси	1	2	3	4	5	6
Дискретный выход $D_{1+0}$	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Дискретный выход $D_{1+1}$	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11

Режим выхода	SR462	SR482	SR502	SR522	SR542	SR562
--------------	-------	-------	-------	-------	-------	-------

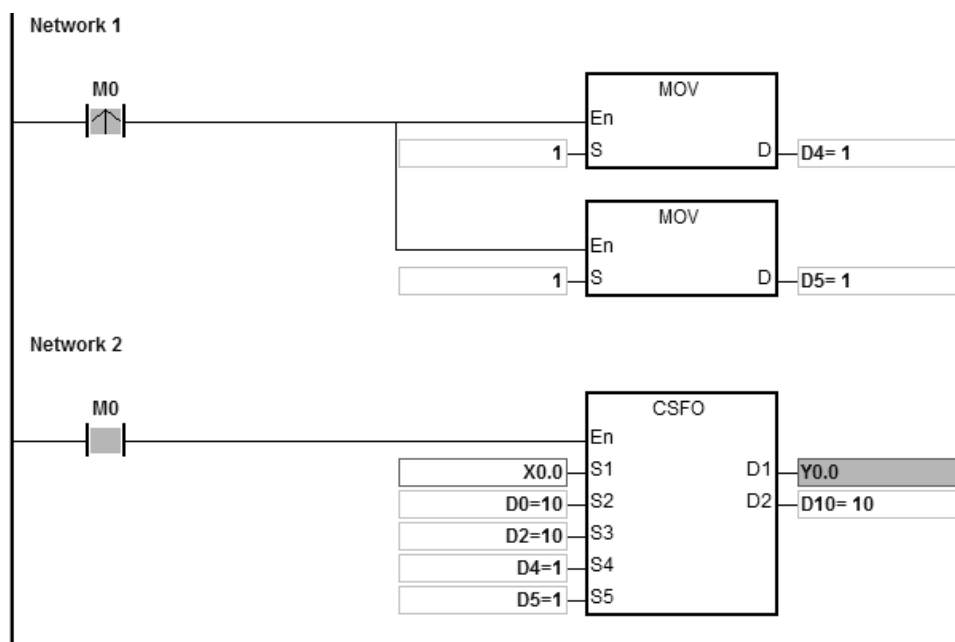
- Операнд  $D_2$  определяет частоту выходных импульсов. Данный параметр объявляется 32-битной переменной с дискретностью 1 Гц.
- Ограничений по количеству использования инструкции в программе нет. Следует отметить, что если инструкция включена, используемые в ней высокоскоростные входы и выходы не могут использоваться другими инструкциями (инструкции выполняться не будут).
- Эта инструкция не может использоваться в языке программирования ST, для прерывания или единократно вызываемых функциональных блоков.

**Примечание:**

- Частота импульсов на входе вычисляется по длительности входного импульса (ON) в положительной половине цикла. Если длительность наличия импульса (ON) и длительность отсутствия импульса (OFF) не равны 1:1, ПЛК берет длительность наличия импульса (ON) стандартной для преобразования по умолчанию.
- Если входной импульс = ON, значит светодиод входа загорается. Обратите внимание, что светодиод входа не горит (OFF) и необходимо поставить флаг смены направления на входе, если используется MPG, но вращение еще не происходит.

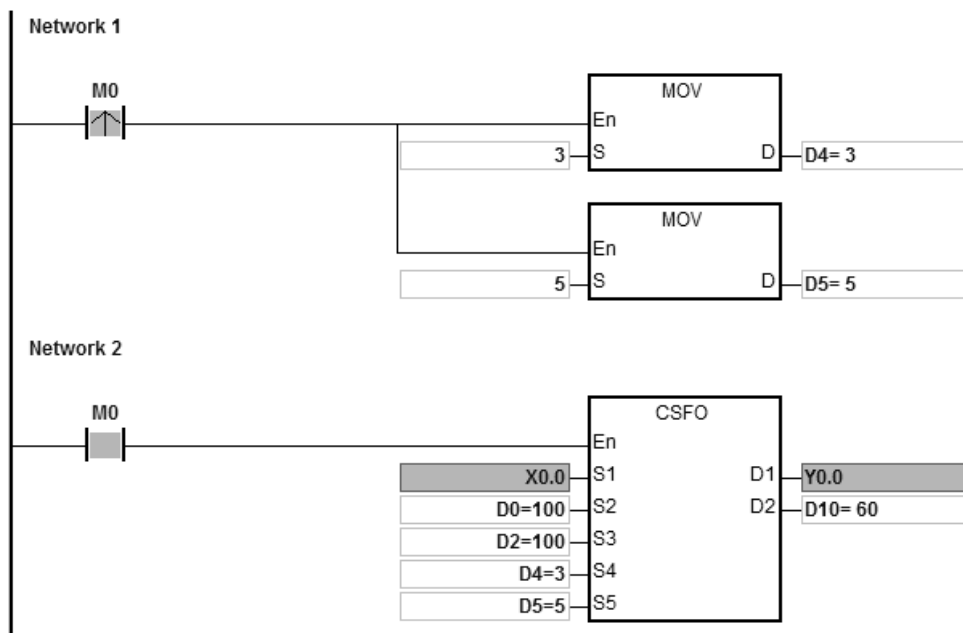
**Пример 1:**

Вход X0.0/X0.1 принимает импульсы для определения MPG. Если M0 = ON, значения операндов S<sub>4</sub> и S<sub>5</sub> равны 1 (D4=1 и D5=1). Если частота на входе 10 Гц (D0, D1=10) и количество импульсов 10 (D<sub>2</sub>, D3=10), выход (Y0.0/Y0.1) выдает 10 импульсов (SR460, SR461=10) с частотой 10 Гц (D10, D11=10).



**Пример 2:**

Если M0 = ON, значения операндов S4 и S5 равны 3 и 5 (D4=3, D5=5) соответственно. Если частота MPG на входе равна 100 Гц (D0, D1=100) и количество импульсов равно 100 (D2, D3=100), выход (Y0.0/Y0.1) выдает 60 импульсов (SR460, SR461=60) с частотой 60 Гц (D10, D11=60).



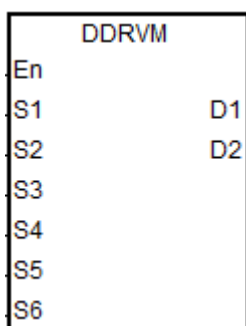
API	Инструкция			Операнд								Описание			
2709	D	DRVM		$S_1, S_2, S_3, S_4, S_5, S_6, D_1, D_2$								Метка выравнивания позиционирования			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$								●	●				○	○		
$S_2$								●	●				○	○		
$S_3$	○		○													
$S_4$								●	●				○	○		
$S_5$								●	●				○	○		
$S_6$								●	●				○	○		
$D_1$		○														
$D_2$		○	○													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$	●												
$S_4$			●				●						
$S_5$			●				●						
$S_6$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**



- $S_1$  : Заданное количество выходных импульсов
- $S_2$  : Заданная частота выходных импульсов
- $S_3$  : Номер входа для приема внешнего сигнала прерывания
- $S_4$  : Количество выходных импульсов в конце передней секции маскирования
- $S_5$  : Количество выходных импульсов в начале задней секции маскирования
- $S_6$  : Количество выходных импульсов в процессе понижения частоты после получения сигнала прерывания
- $D_1$  : Импульсный выход
- $D_2$  : Выход направления импульсов

**Описание:**

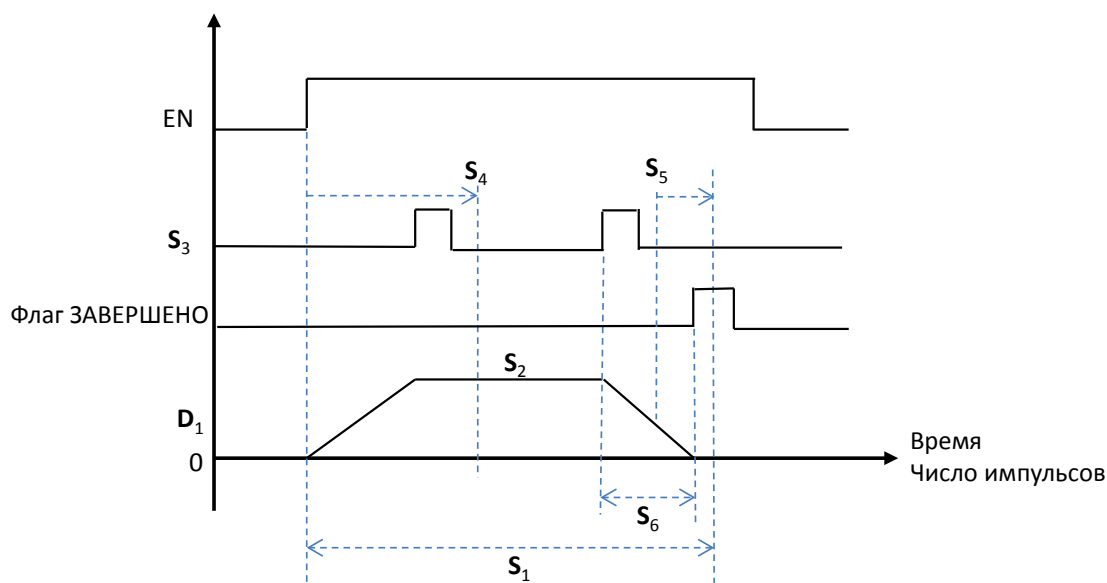
1. Если  $S_1 = 0$ , значит количество выходных импульсов максимальное 32-битное значение (со знаком) и частота выхода не будет понижаться до остановки, пока не будет получен сигнал метки.

Если операнд  $S_1$  не равен 0 и нет внешнего сигнала прерывания, выход передает импульсы в заданном количестве. Диапазон значений операнда  $S_1$ : -2,147,483,648~+2,147,483,647. Знак «+/-» показывает положительное/отрицательное направление.

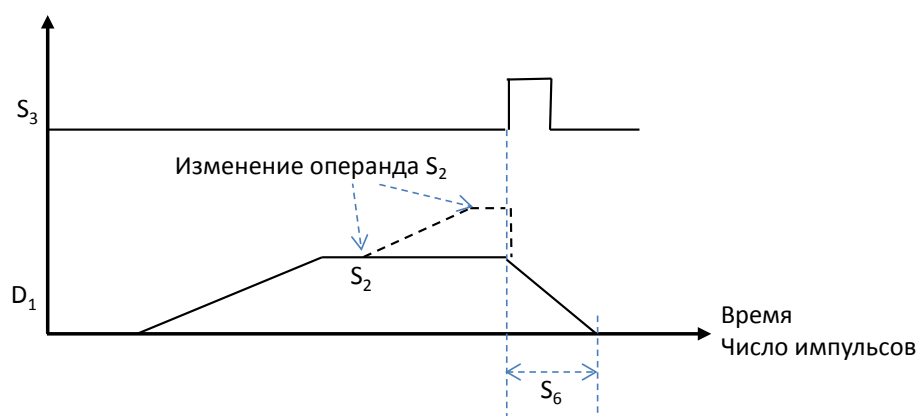
2. Если значение операнда  $S_2$  меньше или равно 0, выдача импульсов не производится. Если частота выходных импульсов больше максимальной частоты, ПЛК берет за частоту импульсов максимальную частоту.
3. Операнд  $S_3$  определяет регистр M или X в качестве входа. На входной сигнал источника влияет цикл опроса, если выбран вход вне диапазона X0.0~X0.15 ПЛК.
4. Инструкция используется с комбинацией номера входа и программы внешнего сигнала прерывания, чтобы гарантировать понижение частоты выхода в реальном времени. Для срабатывания по переднему или заднему фронту импульса внешнего прерывания необходимо выбрать условие на этапе редактирования программы внешнего прерывания.
  - A. С внешним сигналом прерывания (I0xx, I1xx): когда поступает внешний сигнал прерывания, выполняется понижение частоты после получения сигнала метки. Если внешнего прерывания нет, ничего не выполняется.
  - B. Без внешнего сигнала прерывания (I0xx, I1xx): во время выполнения инструкции понижение частоты до остановки выполняется после получения сигнала метки при условии, что на входе X получен передний фронт импульса. На это влияет время сканирования.
5. Количество выходных импульсов в передней секции маскирования от 1 до  $S_4$ . Если операнд  $S_4 = 0$ , значит функция передней секции маскирования выключена. Если количество маскируемых импульсов больше значения, заданного операндом  $S_1$ , инструкция берет количество импульсов, заданное операндом  $S_1$ , в качестве условия количества маскируемых импульсов. Если внешний сигнал прерывания поступает до окончания маскирования импульсов, он автоматически считается недействительным.
6. Количество выходных импульсов в задней секции маскирования от  $S_5$  до  $S_1$ . Если  $S_5 = 0$  или  $S_5 \geq S_1$ , значит функция задней секции маскирования выключена. Если внешний сигнал прерывания оказывается среди секций маскирования, он автоматически считается недействительным. Если  $S_4 > S_5$ , значит внешний сигнал прерывания недействительный в процессе выдачи импульсов.
7. Если заданы обе секции маскирования (передняя и задняя) и  $S_4 < S_5$ , значит сигнал прерывания в секции между  $S_4+1$  и  $S_5-1$  действительный.
8. Если после получения сигнала маски  $S_6$  равен 0, значит понижение частоты до остановки будет выполняться по времени понижения частоты. Если после получения сигнала маски  $S_6 = -1$  или  $< 0$ , значит выход немедленно прекращает выдавать импульсы. Если заданное значение операнда  $S_6$  недостаточное для понижения частоты до остановки за отведенное для этого время, инструкция ограничивает заданную частоту и выполняет операцию понижения частоты до остановки в соответствии с заданным временем

понижения частоты. Если  $S_6$  больше количество импульсов, выдаваемых за отведенное на понижение частоты время, инструкция выдает некоторое количество резервных импульсов с частотой, соответствующей частоте на момент получения прерывания, а затем выполняет операцию понижения частоты.

9. Если выход находится в состоянии понижения частоты и в этот момент поступает внешний сигнал прерывания, инструкция завершает выдачу всех импульсов, заданных операндом  $S_6$ .
10. На приведенном ниже графике показан выходной сигнал, сигнал прерывания, маска и соответствующие флаги (по флагу ЗАВЕРШЕНО см. флаг ЗАВЕРШЕНО SM осей в инструкции DDRVI.)



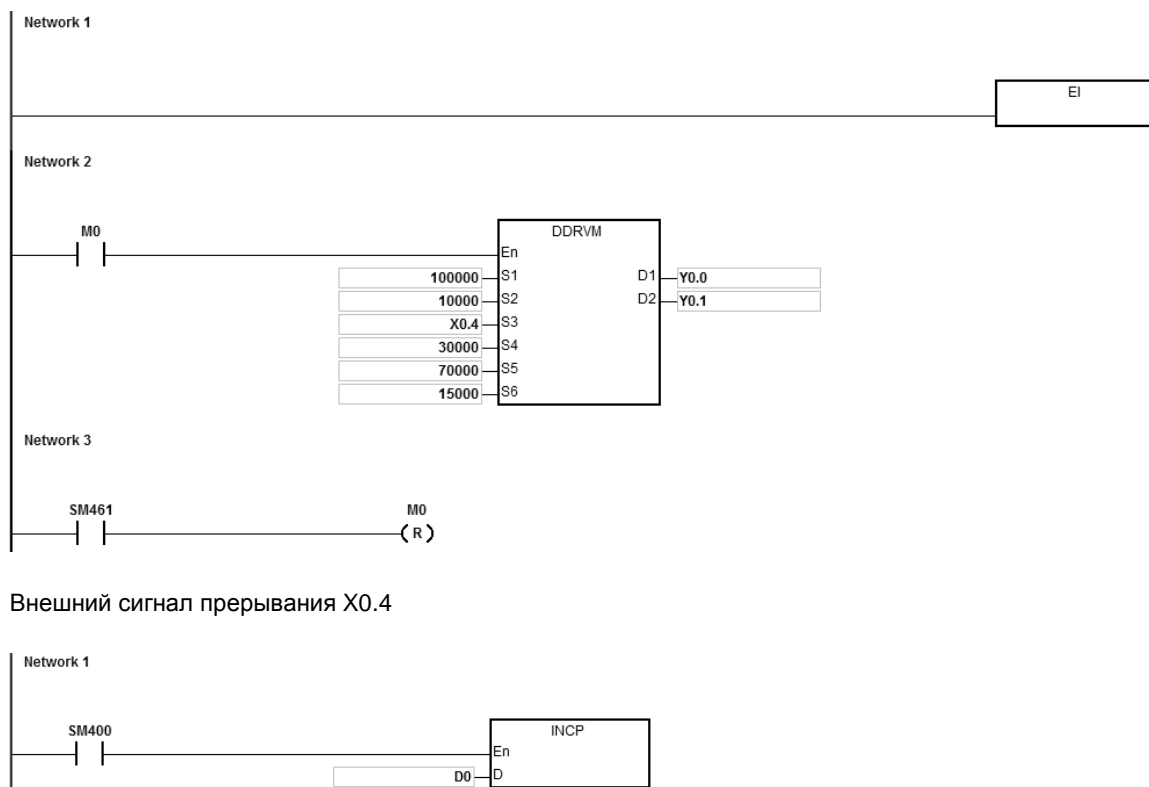
11. Частоту, заданную операндом  $S_2$ , можно изменить в процессе выдачи импульсов. Поскольку количество выходных импульсов в секции понижения частоты задано после включения инструкции, кривая изменений частоты показана на временном графике пунктирной линией.



12. Подробнее см. выходы операндов  $D_1$  и  $D_2$  в инструкции DDRVI. Если выходы не являются рекомендованными, на операнд направления  $D_2$  будет влиять время сканирования.



**Пример 1:**



Внешний сигнал прерывания X0.4

**Примечание:**

1. Если M0 меняет состояние с OFF на ON, выход Y0.0 начинает выдавать импульсы. Спустя более чем 30,000 выданных импульсов на вход X0.4 поступает внешний сигнал прерывания. Значение регистра D0 увеличивается на 1 и после передачи 20,000 выходных импульсов (S<sub>6</sub>: 20,000) выход перестает их выдавать. Если сигнал прерывания не поступает, выход продолжает выдавать импульсы, пока их количество не достигнет 100,000.
2. Если количество выходных импульсов от 1 до 30,000, они принадлежат передней секции маскирования. В этот момент на вход X0.4 поступает сигнал прерывания, и инструкция не выполняет понижение частоты до остановки.
3. Если количество выходных импульсов от 70,000 до 100,000, они принадлежат задней секции маскирования. В этот момент на вход X0.4 поступает сигнал прерывания, и инструкция не выполняет понижение частоты до остановки.
4. Когда все импульсы выданы, SM461 = ON и M0 сбрасывается.

**Дополнительные замечания:**

1. Функция метки выравнивания и функции маскирования:
  - ① Если инструкция DDRVM включена, начинается выдача импульсов.

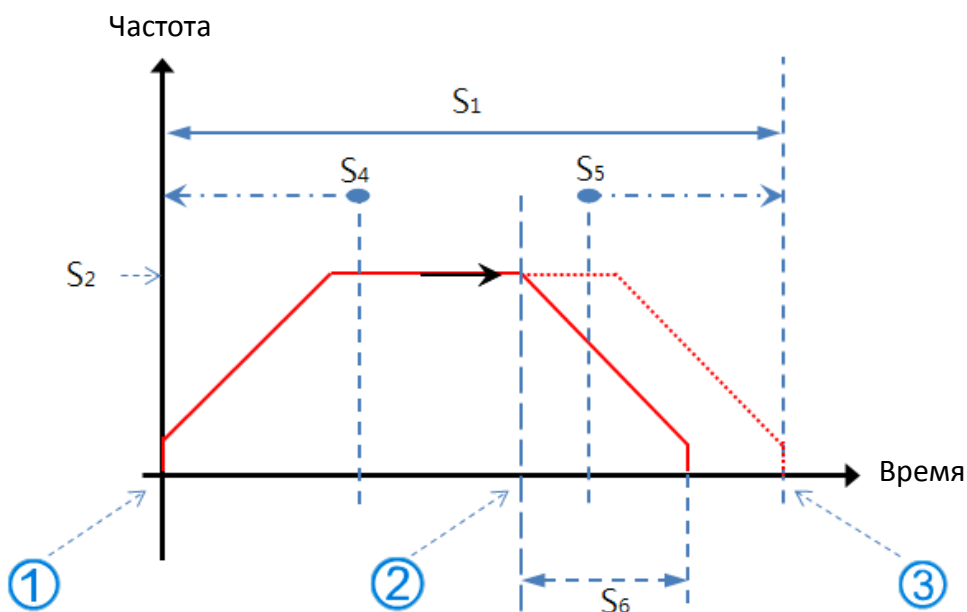
② Если поступает сигнал прерывания для метки выравнивания, частота понижается и выход, выдав импульсы в количестве, заданном операндом  $S_6$ , выключается.

③ Если сигнал прерывания для метки выравнивания не поступает или недействителен, выход по команде инструкции DDRVM выдает импульсы в количестве, заданном операндом  $S_1$ , и затем выключается.

$S_4$ : Количество выходных импульсов в передней секции маскирования. Если сигнал прерывания поступает в течение данной секции, он считается недействительным для метки выравнивания.

$S_5$ : Количество выходных импульсов в задней секции маскирования. Если сигнал прерывания поступает в течение данной секции, он считается недействительным для метки выравнивания.

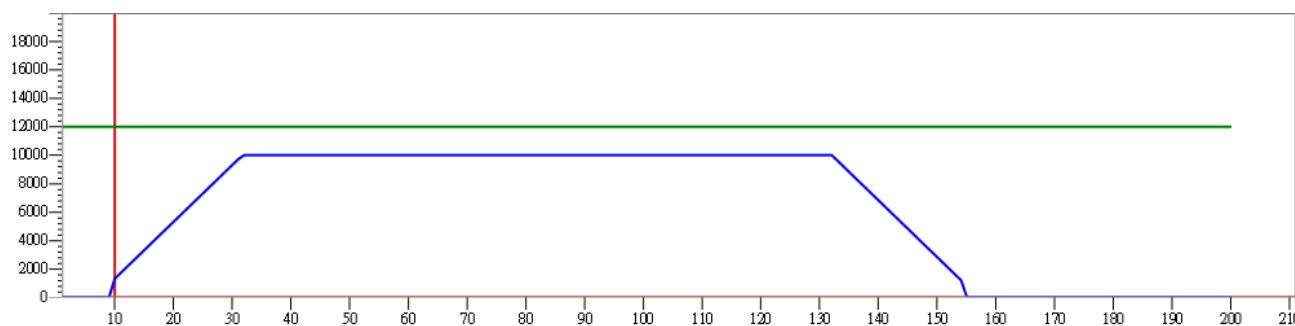
$S_2$ : Заданная частота выходных импульсов



2. Функция метки выравнивания включена в секции повышения частоты, заданной частоты и понижения частоты.

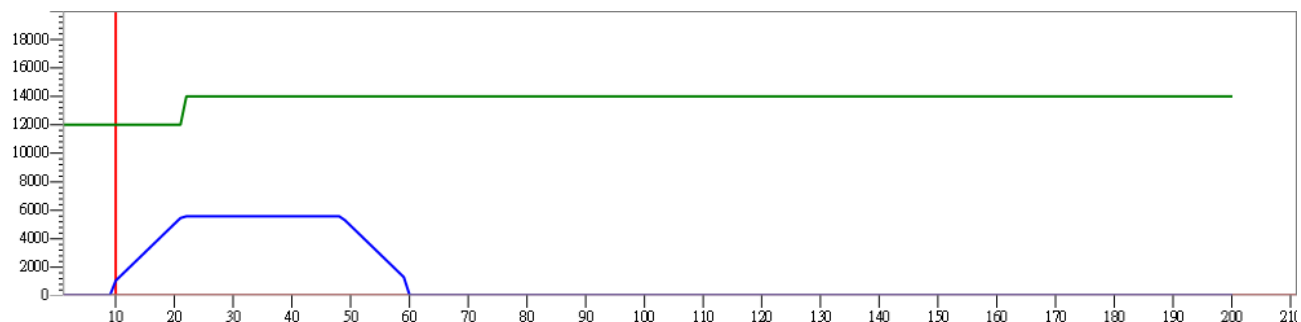
Если функция метки выравнивания не включена:

Если функция метки выравнивания не включена, выход передает 100,000 импульсов и выключается.



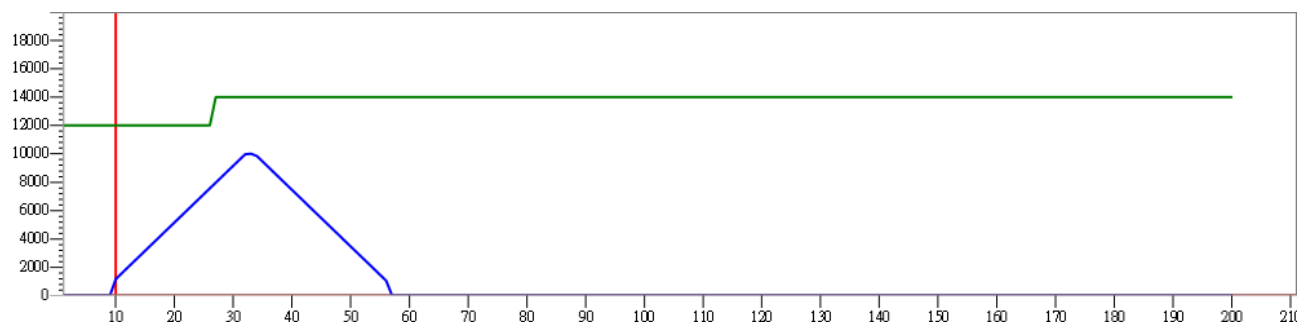
Вариант 1: метка выравнивания поступает на стадии повышения частоты

Функция метки выравнивания выполняется, когда число импульсов 3000. На заданную частоту, равную 10 кГц, нельзя выйти, если  $S_6 = 15,000$ .



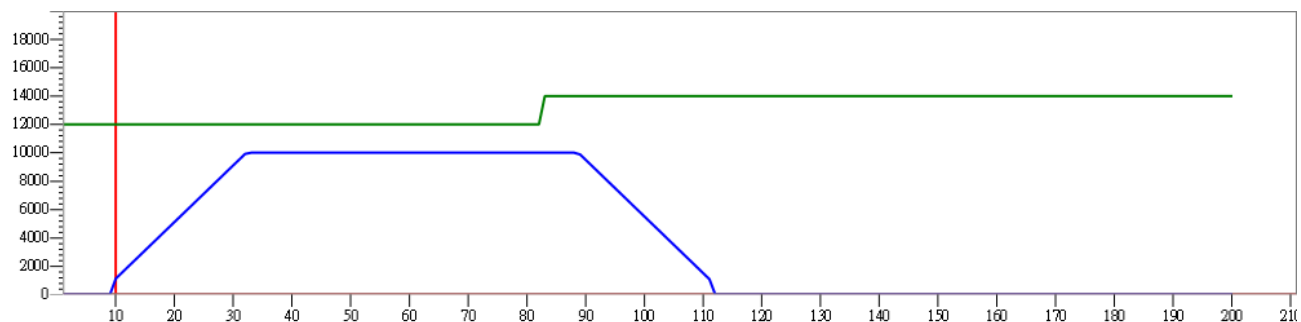
Вариант 2: метка выравнивания поступает на стадии повышения частоты

Функция метки выравнивания выполняется, когда число импульсов 6000. На заданную частоту, равную 10 кГц, можно выйти, если  $S_6 = 15,000$ .



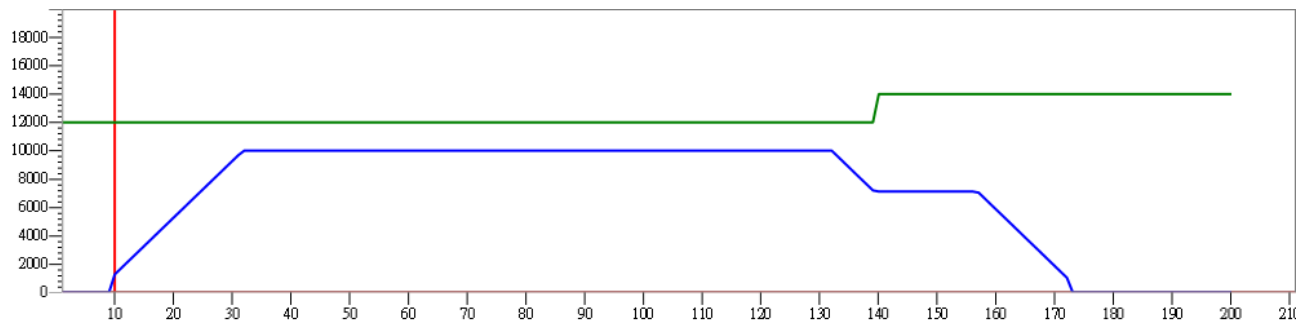
Если метка выравнивания поступает на стадии заданной частоты

Функция метки выравнивания выполняется, когда число импульсов 50,000.



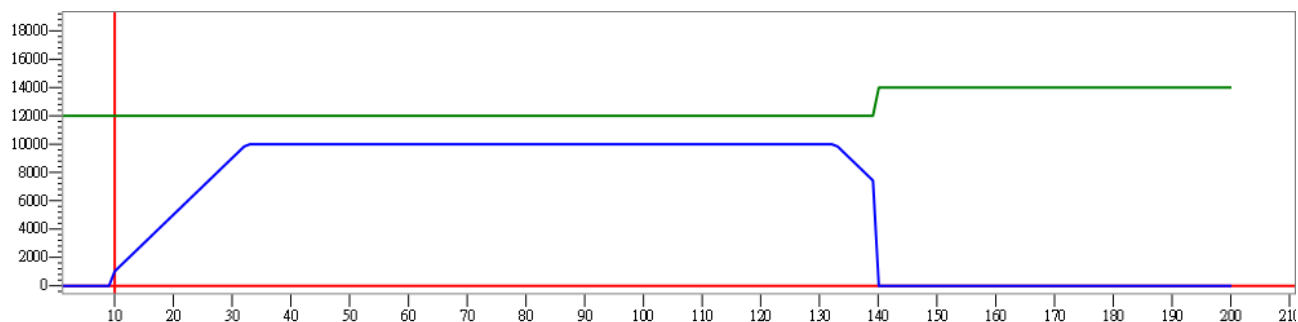
Вариант 1: метка выравнивания поступает на стадии понижения частоты

Функция метки выравнивания выполняется, когда число импульсов 95,000.



Вариант 2: метка выравнивания поступает на стадии понижения частоты

Функция метки выравнивания выполняется, когда число импульсов 95,000. После метки выравнивания количество выходных импульсов для  $S_6$  будет -1 (выход немедленно перестает выдавать импульсы).



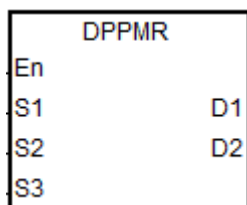
API	Инструкция			Операнд								Описание				
2710	D	PPMR		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D<sub>1</sub>, D<sub>2</sub></b>								2-осевая относительная синхронизация по двум точкам				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>							●	●	●		○		○	○		
<b>S<sub>2</sub></b>							●	●	●		○		○	○		
<b>S<sub>3</sub></b>							●	●	●		○		○	○		
<b>D<sub>1</sub></b>		○														
<b>D<sub>2</sub></b>		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>			●				●						
<b>S<sub>2</sub></b>			●				●						
<b>S<sub>3</sub></b>			●				●						
<b>D<sub>1</sub></b>	●												
<b>D<sub>2</sub></b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**



- S<sub>1</sub>** : Количество выходных импульсов по оси X
- S<sub>2</sub>** : Количество выходных импульсов по оси Y
- S<sub>3</sub>** : Максимальная выходная частота при перемещении от точки к точке
- D<sub>1</sub>** : Импульсный выход для оси X
- D<sub>2</sub>** : Импульсный выход для оси Y

**Описание:**

- Операнды **S<sub>1</sub>** и **S<sub>2</sub>** определяют количество выходных импульсов (относительное позиционирование) по осям X и Y. Диапазон: -2,147,483,648 ~ +2,147,483,647 (знак «+/-» показывает положительное/отрицательное направление).
- Операнд **S<sub>3</sub>** определяет максимальную выходную частоты при движении от точки к точке. Диапазон: 1 Гц ~ +200 кГц.
- Операнды **D<sub>1</sub>** и **D<sub>2</sub>** являются дискретными выходами осей X и Y соответственно. Выходу можно назначить следующие 6 осей, но фиксированные выходы направления менять нельзя. Сигнал направления: если сигнала нет (OFF), значит положительное, а если есть (ON), то отрицательное. Если текущий режим выхода «импульс + направление» не используется (по умолчанию: 0), измените режим выхода, выставив SR = 1.

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Выходы операндов <b>D<sub>1</sub></b> и <b>D<sub>2</sub></b>	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Фиксированные выходы направления	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Режим выхода	SR462	SR482	SR502	SR522	SR542	SR562

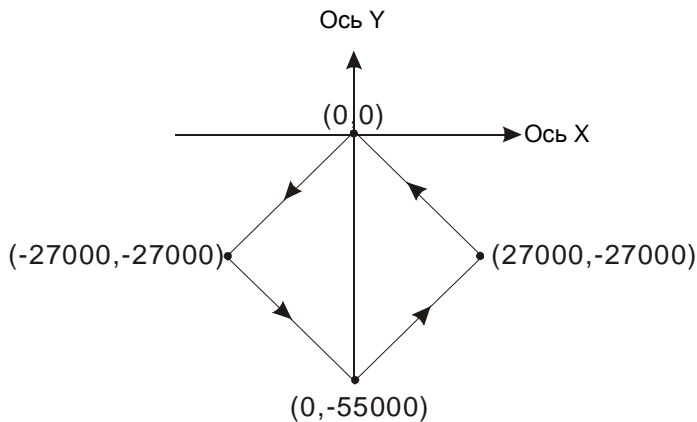
- ПЛК назначает значение, заданное операндом **S<sub>3</sub>**, оси X или Y, у которой больше количество импульсов. Если количество импульсов двух осей так сильно отличается, что правильную выходную частоту невозможно вычислить по количеству их выходных импульсов, ПЛК автоматически уменьшает максимальную частоту перемещения от точки к точке и аварийный сигнал не выдается.
- Для выполнения инструкции необходимы такие параметры, как начальная/конечная частота и время повышения/понижения частоты по осям. В качестве эталонного значения вместо исходных параметров осей берутся параметры оси X. Например, если оси X назначена ось 1, а оси Y ось 3, стартовая частота и время понижения/повышения частоты будут взяты из параметров оси 1.
- Флаги осей и соответствующие регистры SM/SR см. в следующей таблице.

Номер оси	Ось 1	Ось 2	Ось 3	Ось 4	Ось 5	Ось 6
Флаг ЗАНЯТО	SM460	SM480	SM500	SM520	SM540	SM560
Флаг ЗАВЕРШЕНО	SM461	SM481	SM501	SM521	SM541	SM561
Текущее состояние выхода	SR460 SR461	SR480 SR481	SR500 SR501	SR520 SR521	SR540 SR541	SR560 SR561
Начальная/конечная частота	SR463	SR483	SR503	SR523	SR543	SR563
Время линейного повышения	SR464	SR484	SR504	SR524	SR544	SR564
Время линейного понижения	SR465	SR485	SR505	SR525	SR545	SR565

- Ограничений по количеству использования инструкции в программе нет. Но если инструкция включена и идет выдача импульсов по оси Y, по ее команде выход не может выдавать импульсы, пока не будет завершена выдача импульсов по оси Y.
- По завершении выдачи импульсов 2-осевой синхронизации по двум точкам ставятся соответствующие флаги ЗАВЕРШЕНО по двум осям. Поскольку одинаковое время завершения 2-осевой синхронизации каждый раз обеспечить нельзя, выполнение следующей пользовательской программы должно начинаться после того, как будут одновременно поставлены флаги ЗАВЕРШЕНО по обеим осям.

**Пример 1:**

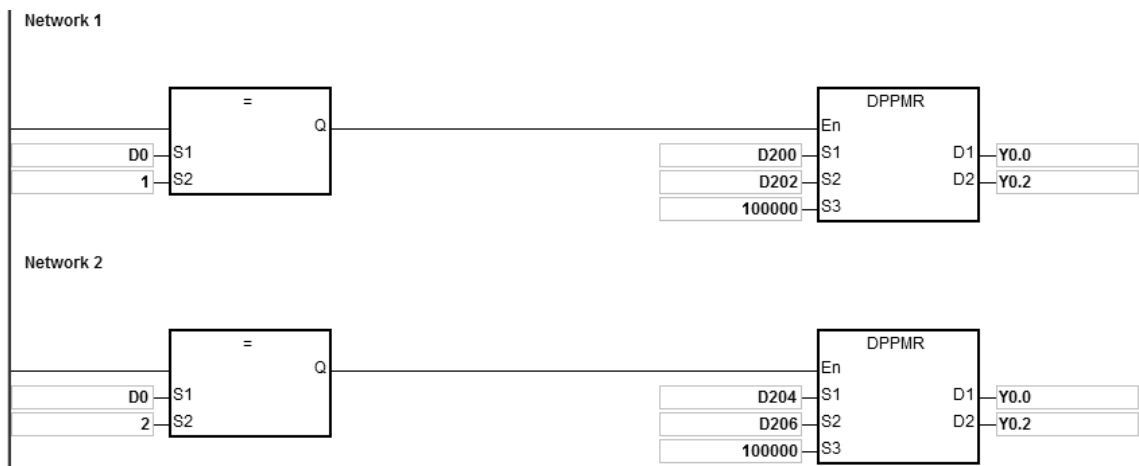
1. Нарисуем ромб, как показано ниже

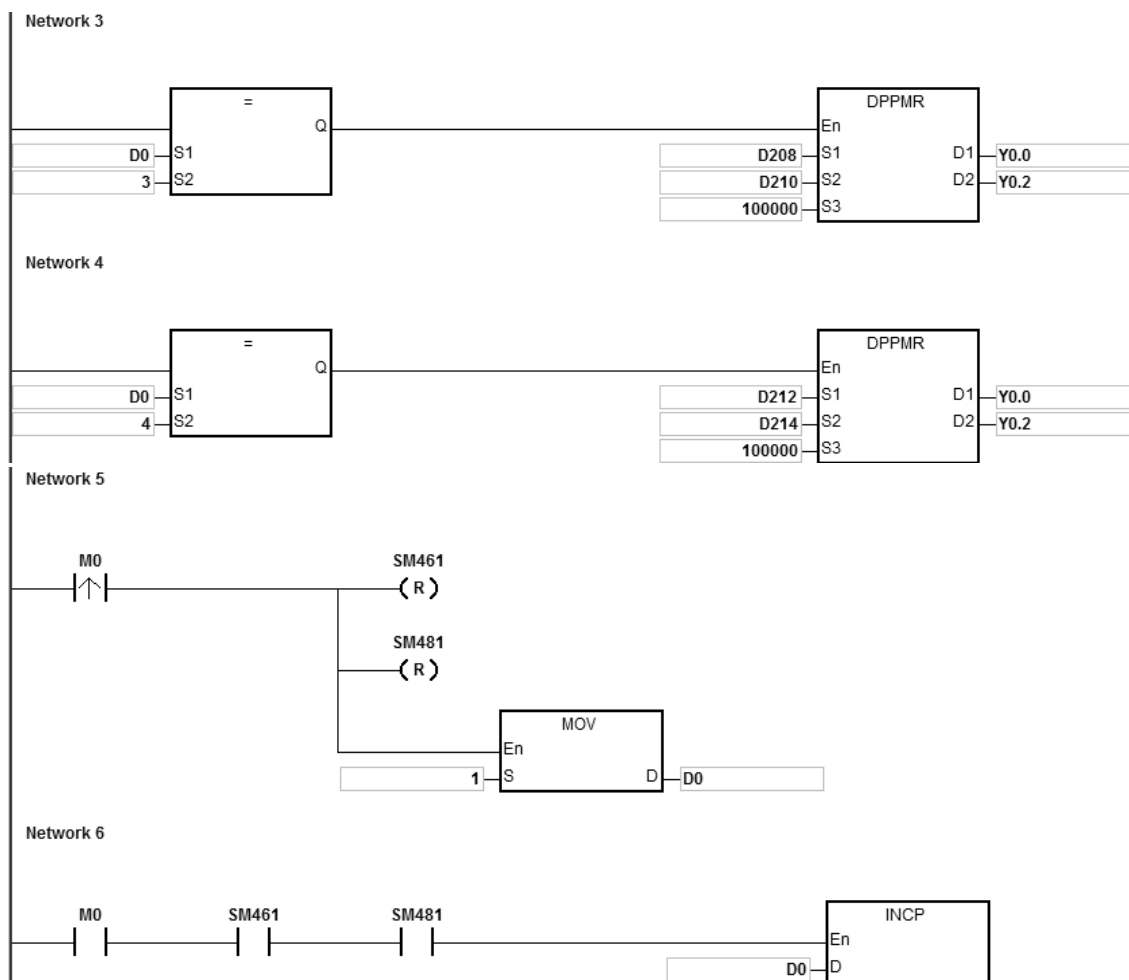


2. Шаги:

- Введем координаты четырех точек (0, 0), (-27000, -27000), (0, -55000), (27000, -27000) (как показано на рисунке выше). Вычислим относительные координаты четырех точек и получим (-27000, -27000), (27000, -28000), (27000, 28000) и (-27000, 27000). Запишем их в 32-битные регистры (D200, D202), (D204, D206), (D208, D210), (D212, D214).
- ВКЛЮЧИМ контроллер. Переведем контакт M0 в состояние ON, чтобы запустить построение линии по 2 осям.

6





3. Работа:

Когда ПЛК включен и M0 = ON, контроллер начинает первое перемещение от точки к точке с частотой 100 кГц. Значение регистра D0 увеличивается на 1 по завершении перемещения от точки к точке и автоматически начинается второе перемещение от точки к точке. Данная схема повторяется, пока не завершится четвертое перемещение от точки к точке.

Пример 2 (программа на языке ST):

```

0001 IF M0 THEN
0002     DPPMR (1000,1000,1000,Y0.0,Y0.2) ;
0003     M1 := TRUE;
0004 ELSIF M1 AND SM461 AND SM481 THEN
0005     SM470 := TRUE;
0006     SM490 := TRUE;
0007     M1 := FALSE;
0008 END_IF;

```

Описание:

1. Когда M0 включен, Y0.0 и Y0.2 одновременно выдают 1000 импульсов на частоте 1 кГц. Когда M1 включен, он указывает, что Y0.0 и Y0.2 используются для высокоскоростного импульсного выхода.



2. Когда импульсный выход Y0.0 завершен, включен SM461. Когда импульсный выход Y0.2 завершен, включен SM481. И если M1 также включен, он активирует SM470 (автоматический сброс завершения вывода для Y0.0) и SM490 (автоматический сброс завершения выхода для Y0.2). Y0.0 и Y0.2 свободны и готовы к повторному использованию, а M1 выключается.
3. Когда включается M0, импульсный выход перезапускается.
4. Если в программе более двух высокоскоростных команд вывода, не используйте имя переменной M1 повторно.
5. Лучше не использовать язык ST в программе, если есть вероятность, что вывод необходимо остановить.

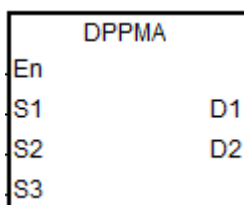
API	Инструкция			Операнд								Описание				
2711	D	PPMA		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D<sub>1</sub>, D<sub>2</sub></b>								2-осевая абсолютная синхронизация по двум точкам				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>							●	●	●		○		○	○		
S <sub>2</sub>							●	●	●		○		○	○		
S <sub>3</sub>							●	●	●		○		○	○		
D <sub>1</sub>		○														
D <sub>2</sub>		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>			●				●						
S <sub>2</sub>			●				●						
S <sub>3</sub>			●				●						
D <sub>1</sub>	●												
D <sub>2</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**



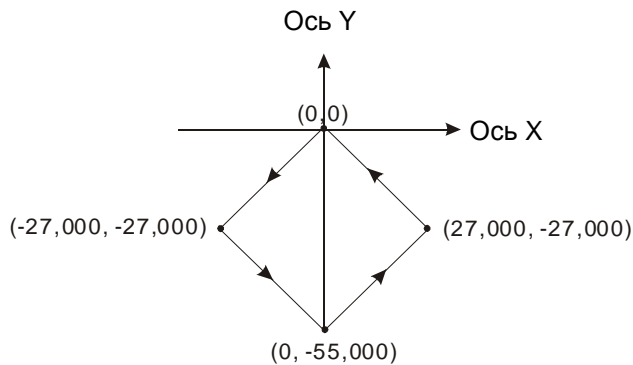
- S<sub>1</sub>** : Количество выходных импульсов по оси X
- S<sub>2</sub>** : Количество выходных импульсов по оси Y
- S<sub>3</sub>** : Максимальная выходная частота при перемещении от точки к точке
- D<sub>1</sub>** : Импульсный выход для оси X
- D<sub>2</sub>** : Импульсный выход для оси Y

**Описание:**

- Операнды **S<sub>1</sub>** и **S<sub>2</sub>** определяют количество выходных импульсов (абсолютного позиционирования) по осям X и Y.
- Подробнее см. описание инструкции DPPMR (API 2710).

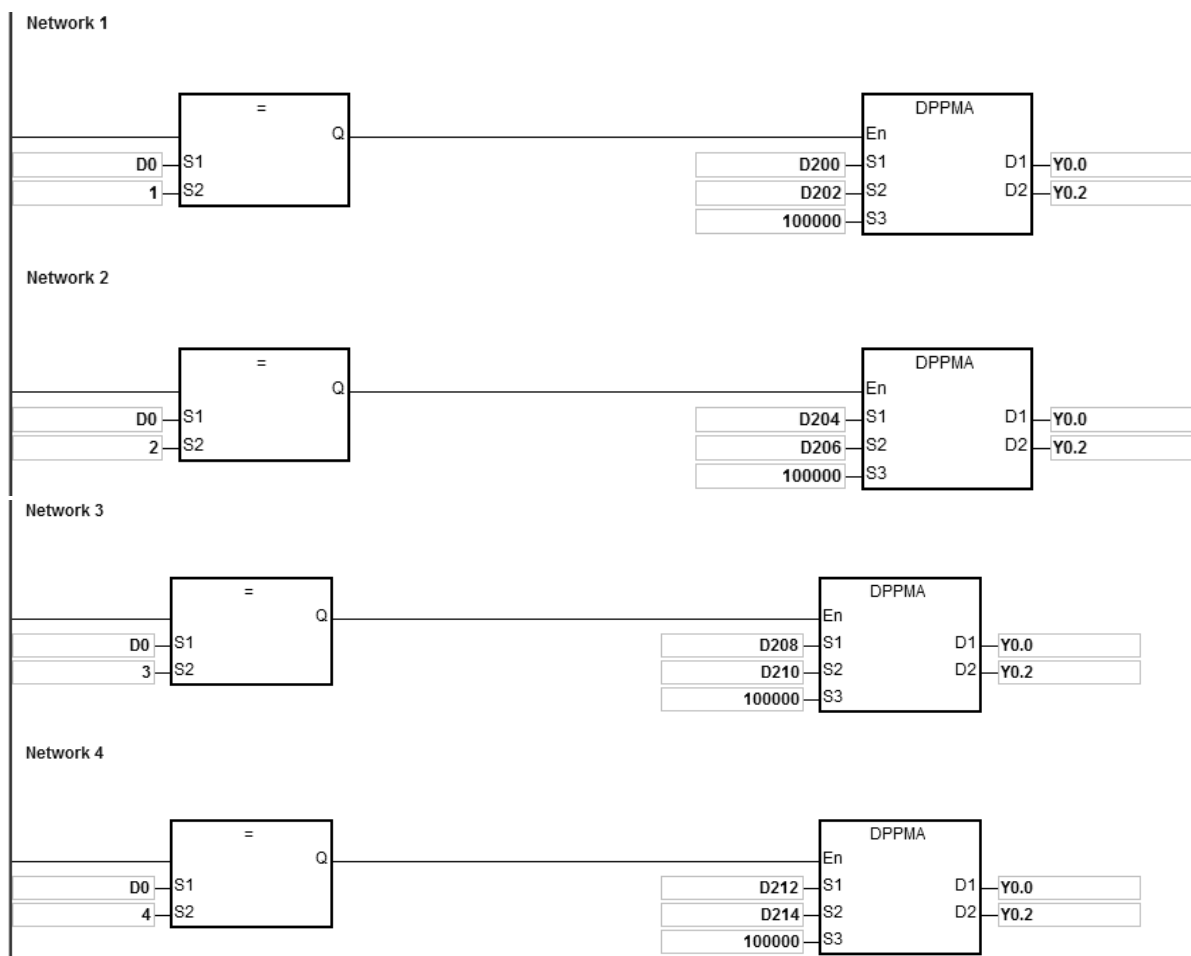
**Пример:**

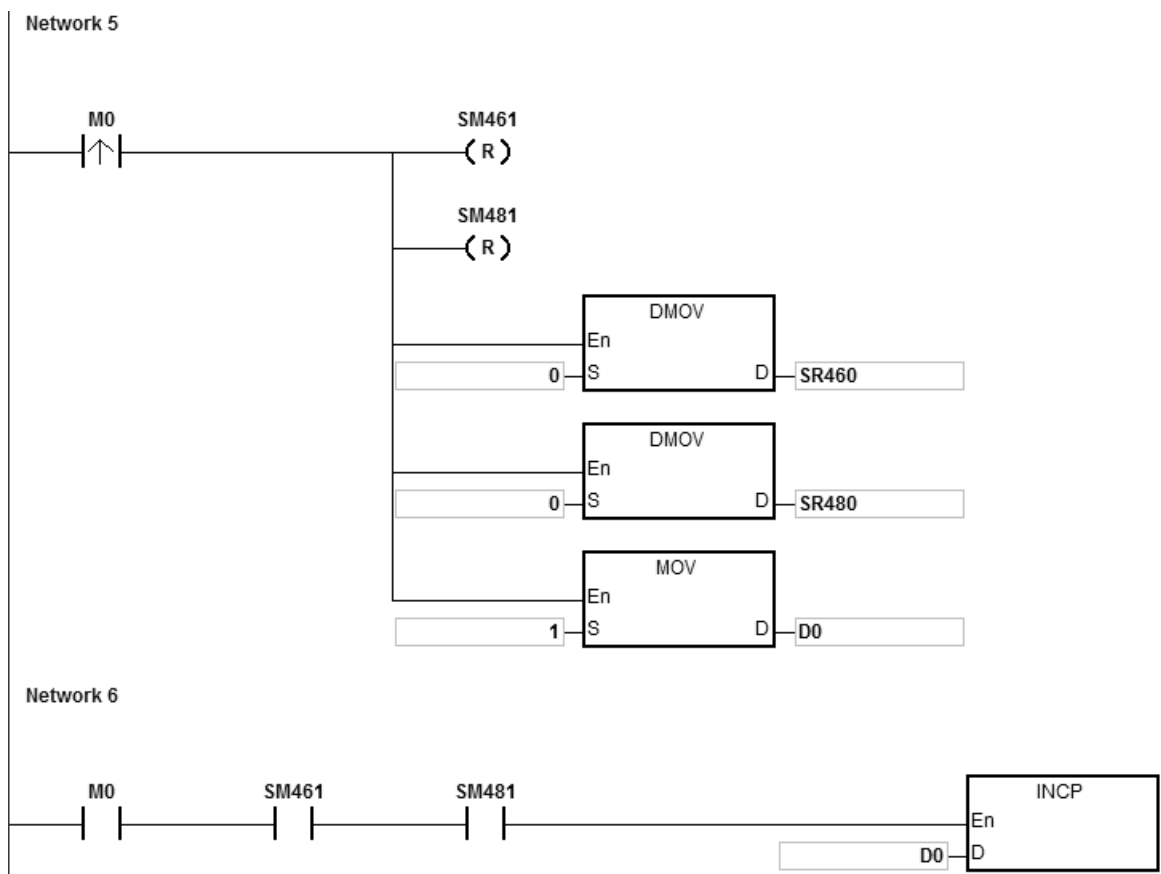
- Построим ромб, как показано на рисунке ниже.



2. Шаги:

- Введем четыре координаты (-27000, -27000), (0, -55000), (27000, -27000) и (0, 0) (как показано на рисунке выше). Запишем их в 32-битные регистры (D200, D202), (D204, D206), (D208, D210), (D212, D214).
- ВКЛЮЧИМ контроллер. Переведем контакт M0 в состояние ON, чтобы запустить построение линии по 2 осям.





3. Работа:

Когда ПЛК включен и M0 = ON, контроллер начинает первое перемещение от точки к точке с частотой 100 кГц. Значение регистра D0 увеличивается на 1 по завершении перемещения от точки к точке и автоматически начинается второе перемещение от точки к точке. Данная схема повторяется, пока не завершится четвертое перемещение от точки к точке.

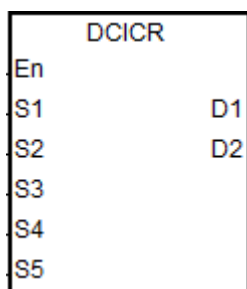
API	Инструкция			Операнд								Описание				
2712	D	CICR		$S_1, S_2, S_3, S_4, S_5, D_1, D_2$								2-осевая дуговая интерполяция по часовой стрелке в относительных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
$S_4$							●	●	●		○		○	○		
$S_5$							●	●	●		○		○	○		
$D_1$		○														
$D_2$		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$			●				●						
$S_5$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

Символьное обозначение:



- $S_1$  : Заданная координата по оси X (относительное позиционирование)
- $S_2$  : Заданная координата по оси Y (относительное позиционирование)
- $S_3$  : Смещение центра
- $S_4$  : Заданная опорная частота
- $S_5$  : Выбор функции
- $D_1$  : Импульсный выход для оси X
- $D_2$  : Импульсный выход для оси Y

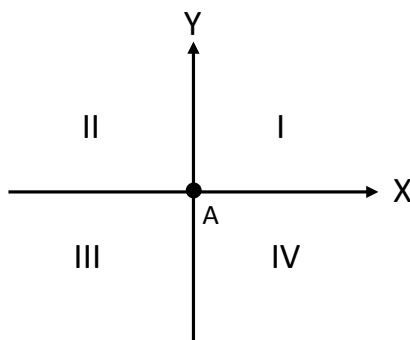
Описание:

- Операнды  $S_1$ ,  $S_2$  и  $S_3$  определяют заданные координаты по оси X, оси Y (относительное положение) и сдвига центра круга соответственно. Подробнее см. описание дуговой интерполяции по часовой стрелке.
- Операнды  $D_1$  и  $D_2$  являются импульсными выходами осей X и Y соответственно. Выбор выходов и режимов выходов осей в инструкции DPPMR.

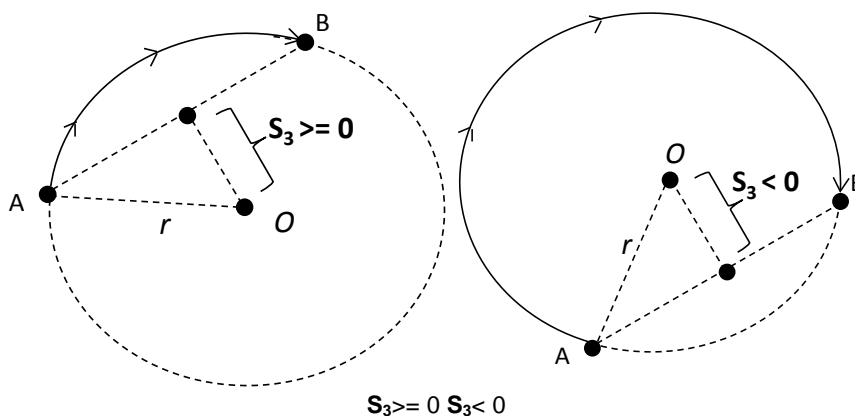
- Операнд  $S_4$  определяет заданную опорную частоту. Заданная опорная частота используется для предварительного вычисления, когда контроллер планирует траекторию движения по дуге после запуска инструкции. Если по результатам предварительного расчета построить траекторию движения по дуге не удастся, выходная частота автоматически снижается, чтобы построение дуги было синхронизировано.
- Операнд  $S_5$  – это выбор функции. Если значение 0, разрешение дуги берется равным  $10^\circ$  в качестве стандартного угла для движения. Если значение 1, разрешение дуги берется равным  $5^\circ$  в качестве стандартного угла для движения. Если значение любое, кроме 1, 0 берется в качестве значения для выполнения инструкции.

**Построение дуги по часовой стрелке:**

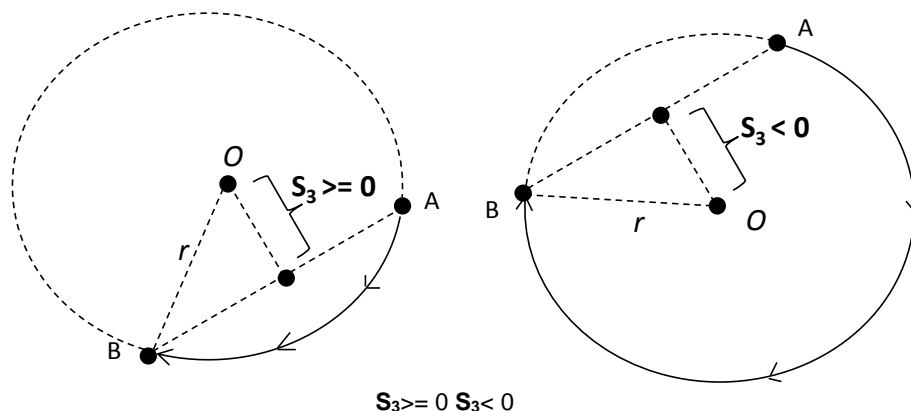
- См. четыре квадранта (QI, QII, QIII и QIV) системы координат, приведенной ниже. Точкой A обозначаются текущие координаты по осям X и Y.



- Точка B – это координаты, заданные операндами  $S_1$  и  $S_2$ . Точка O – это центр окружности, на которой лежат точки A и B.
- Операнд  $S_3$  – это расстояние, на которое сдвигается центр окружности O.
- Если заданная точка B находится в квадранте QI и QIV точки A, траектория движения по дуге показана жирными линиями на рисунке ниже в зависимости от значений операнда  $S_3$  ( $r$ : радиус).



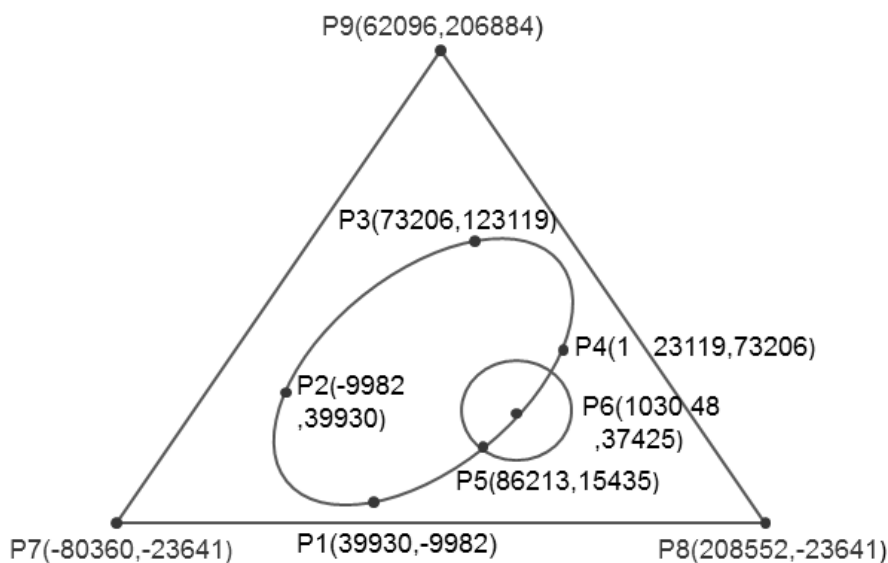
- Если заданная точка B находится в квадранте QII и QIII точки A, траектория движения по дуге показана жирными линиями на рисунке ниже в зависимости от значений операнда  $S_3$ .



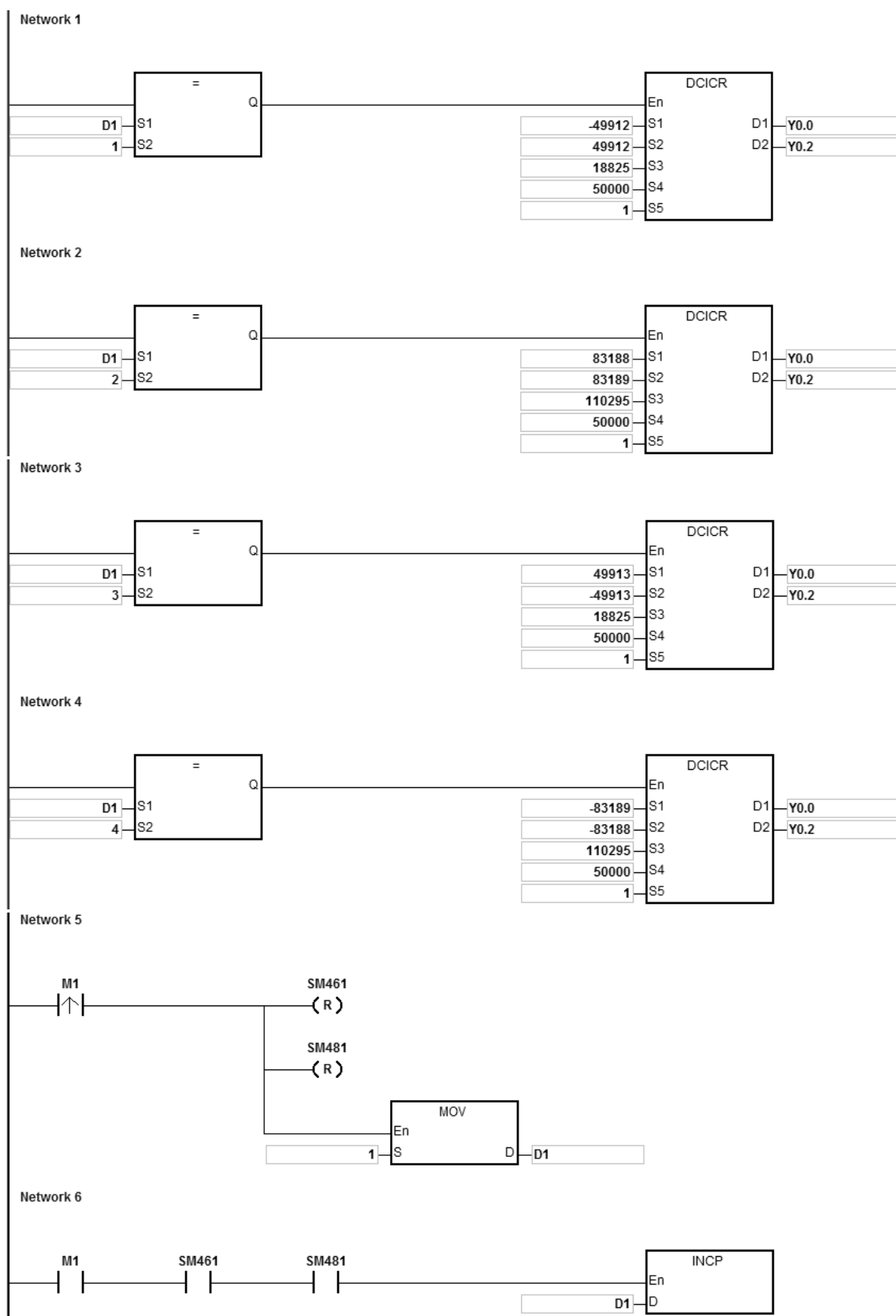
6. Если координата по оси X заданной точки B = 0, а координата по оси Y  $\geq 0$ , точка B считается точкой, находящейся в квадранте QI точки A. Если заданная координата по оси Y < 0, точка B считается точкой, находящейся в квадранте QIII точки A.

**Пример:**

1. Нарисуем логотип компании DELTA, как показано на рисунке ниже.



2. Шаги: логотип можно поделить на три части.
- Для построения эллипса используется инструкция DCICR, осуществляющая дуговую интерполяцию в относительных координатах по часовой стрелке.
  - Абсолютные координаты эллипса: P1 (39930, -9982), P2 (-9982, 39930), P3 (73206, 123119) и P4 (123119, 73206).
  - Возьмем точку (39930, -9982) в качестве начальной и получим относительные координаты: (-49912, 49912), (83188, 83189), (49913, -49913) и (-83189, -83188).



Когда ПЛК работает и M1 =s ON, контроллер начинает построение первого сегмента дуги. Значение операнда D1 увеличивается на 1 по завершении построения сегмента дуги и автоматически начинается построение второго сегмента дуги. Схема повторяется, пока не будет построен четвертый сегмент дуги.



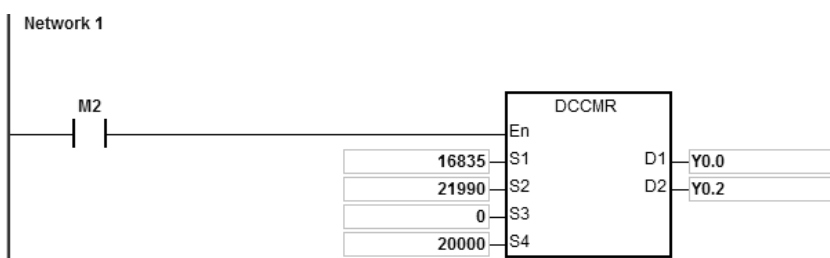
Когда D1=1, при помощи инструкции DCICR ПЛК выполняет построение первого сегмента дуги от P1 до P2 со сдвигом центра окружности: 18825 и разрешение дуги: 5.

Когда D1=2, при помощи инструкции DCICR ПЛК выполняет построение второго сегмента дуги от P2 до P3 со сдвигом центра окружности: 110295 и разрешение дуги: 5.

Когда D1=3, при помощи инструкции DCICR ПЛК выполняет построение третьего сегмента дуги от P3 до P4 со сдвигом центра окружности: 18825 и разрешение дуги: 5.

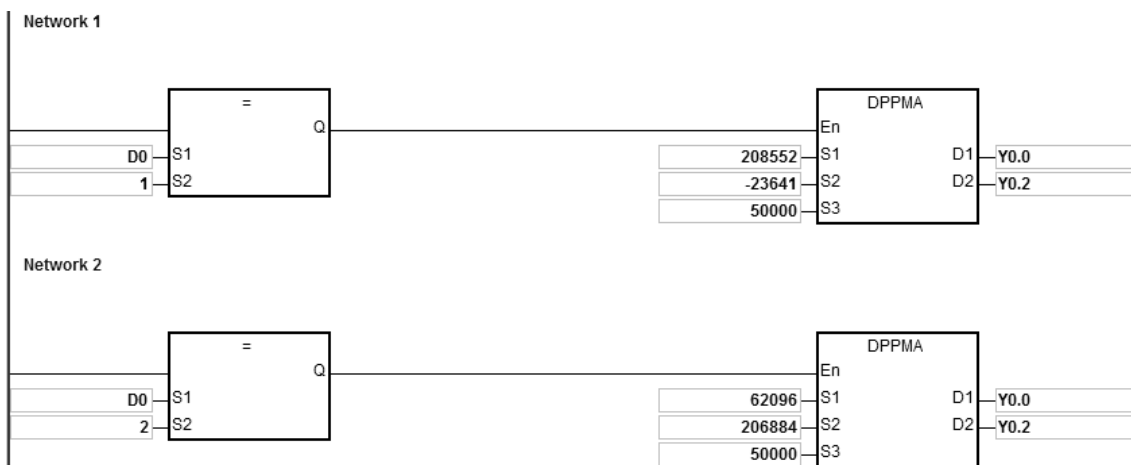
Когда D1=4, при помощи инструкции DCICR ПЛК выполняет построение четвертого сегмента дуги от P4 до P1 со сдвигом центра окружности: 110295 и разрешение дуги: 5.

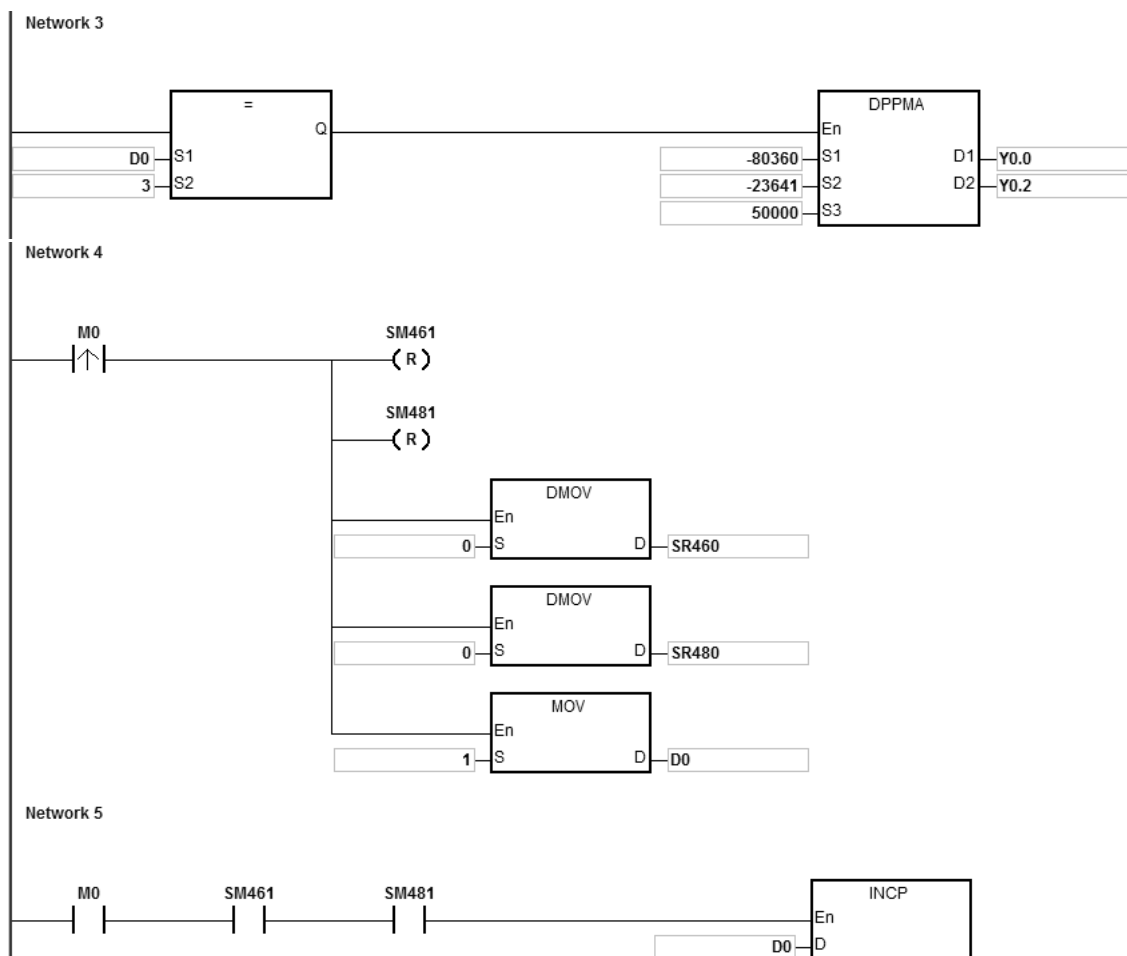
- Для построения окружности используется инструкция DCCMR, которая осуществляет построение окружности в относительных координатах
- Абсолютные координаты окружности: P5 (86213, 15435) и P6 (103048, 37425). Возьмем точку (86213, 15435) за начальную и относительные координаты центра окружности (16835, 21990).



Когда ПЛК работает и M2 = ON, он начинает построение окружности в относительных координатах с заданной опорной частотой 20 кГц.

- Для построения треугольника используется инструкция DPPMA, осуществляющая 2-осевую синхронизацию в абсолютных координатах.
- Абсолютные координаты треугольника = начальная точка P7 (-80360, 23641), P8 (208552, -23641) и P9 (62096, 206884).





Когда ПЛК работает и M0 = ON, контроллер начинает построение первого сегмента в рамках 2-осевой синхронизации с частотой 50 кГц. Значение операнда D1 увеличивается на 1 по завершении построения сегмента в рамках 2-осевой синхронизации, и автоматически начинается построение второго сегмента. Схема повторяется, пока не будет построен третий сегмент в рамках 2-осевой синхронизации.

Когда D1=1, ПЛК начинает построение линии от P7 до P8 при помощи инструкции DPPMA.

Когда D1=2, ПЛК начинает построение линии от P8 до P9 при помощи инструкции DPPMA.

Когда D1=3, ПЛК начинает построение линии от P9 до P7 при помощи инструкции DPPMA.

API	Инструкция			Операнд								Описание				
2713	D	CICA		$S_1, S_2, S_3, S_4, S_5, D_1, D_2$								2-осевая дуговая интерполяция по часовой стрелке в абсолютных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
$S_4$							●	●	●		○		○	○		
$S_5$							●	●	●		○		○	○		
$D_1$		○														
$D_2$		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$			●				●						
$S_5$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

6

**Символьное обозначение:**

DCICA	
En	
S1	D1
S2	D2
S3	
S4	
S5	

- $S_1$  : Заданная координата по оси X (абсолютное позиционирование)
- $S_2$  : Заданная координата по оси Y (абсолютное позиционирование)
- $S_3$  : Смещение центра
- $S_4$  : Заданная опорная частота
- $S_5$  : Функция
- $D_1$  : Импульсный выход для оси X
- $D_2$  : Импульсный выход для оси Y

**Описание:**

- Операнды  $S_1$  и  $S_2$  определяют заданные координаты по осям X и Y соответственно. Подробнее см. описание инструкции DCICR.
- См. Пример 2 из инструкции DPPMR (API 2710) для программирования на языке ST.

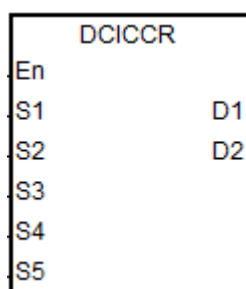
API	Инструкция			Операнд								Описание				
2714	D	CICCR		$S_1, S_2, S_3, S_4, S_5, D_1, D_2$								2-осевая дуговая интерполяция против часовой стрелки в относительных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
$S_4$							●	●	●		○		○	○		
$S_5$							●	●	●		○		○	○		
$D_1$		○														
$D_2$		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$			●				●						
$S_5$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**



- $S_1$  : Заданная координата по оси X (относительное позиционирование)
- $S_2$  : Заданная координата по оси Y (относительное позиционирование)
- $S_3$  : Смещение центра
- $S_4$  : Заданная опорная частота
- $S_5$  : Функция
- $D_1$  : Импульсный выход для оси X
- $D_2$  : Импульсный выход для оси Y

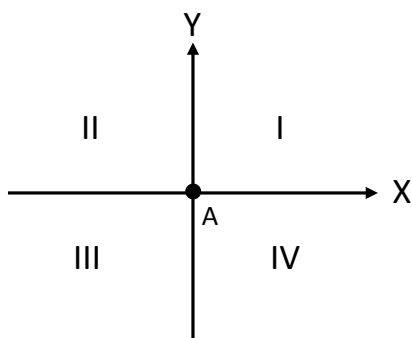
**Описание:**

- Операнды  $S_1$ ,  $S_2$  и  $S_3$  определяют заданные координаты по оси X, оси Y и сдвига центра окружности соответственно. Подробнее см. описание построения дуги против часовой стрелки.
- Операнды  $D_1$  и  $D_2$  задают импульсные выходы для осей X и Y соответственно. Выбор выходов и режимов выходов осей в инструкции DPPMR.

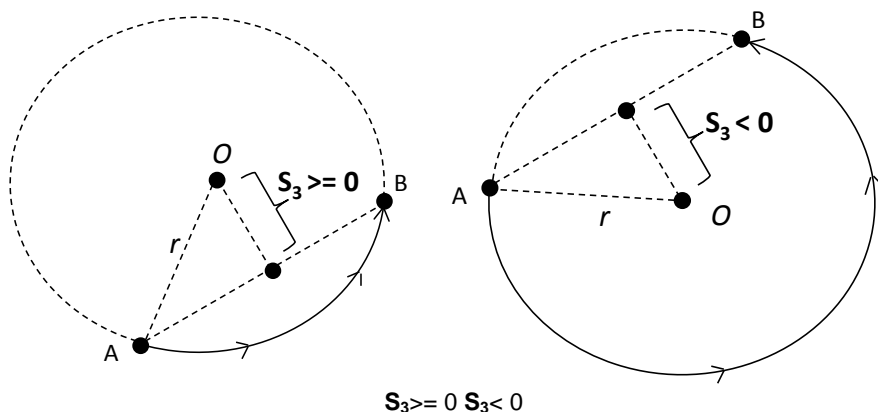
3. Операнд  $S_4$  определяет заданную опорную частоту. Заданная опорная частота используется для предварительного вычисления, когда контроллер планирует траекторию движения по дуге после запуска инструкции. Если по результатам предварительного расчета построить траекторию движения по дуге не удастся, выходная частота автоматически снижается, чтобы построение дуги было синхронизировано.
4. Операнд  $S_5$  – это выбор функции. Если значение 0, разрешение дуги берется равным  $10^\circ$  в качестве стандартного угла для движения. Если значение 1, разрешение дуги берется равным  $5^\circ$  в качестве стандартного угла для движения. Если значение любое кроме 1, 0 берется в качестве значения для выполнения инструкции.
5. См. Пример 2 из инструкции DPPMR (API 2710) для программирования на языке ST.

**Построение дуги против часовой стрелки:**

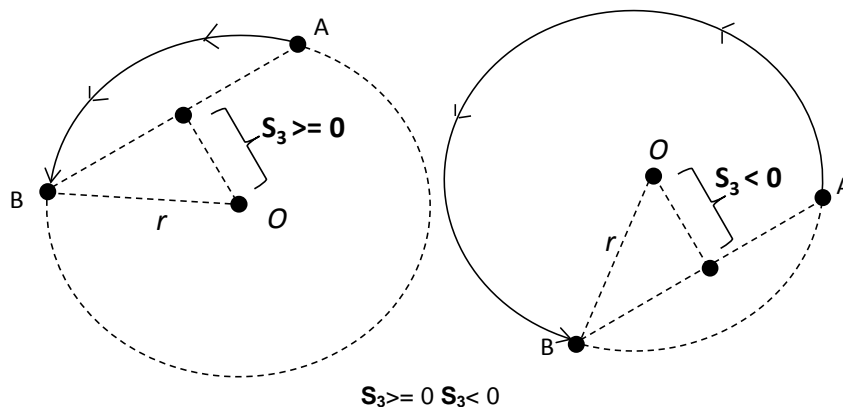
1. См. четыре квадранта (QI, QII, QIII и QIV) системы координат, приведенной ниже. Точкой A обозначаются текущие координаты по осям X и Y.



2. Точка B – это координаты, заданные операндами  $S_1$  и  $S_2$ . Точка O – это центр окружности, на которой лежат точки A и B.
3. Операнд  $S_3$  – это расстояние, на которое сдвигается центр окружности O.
4. Если заданная точка B находится в квадранте QI и QIV точки A, траектория движения по дуге показана жирными линиями на двух рисунках ниже в зависимости от значений операнда  $S_3$  (r: радиус).



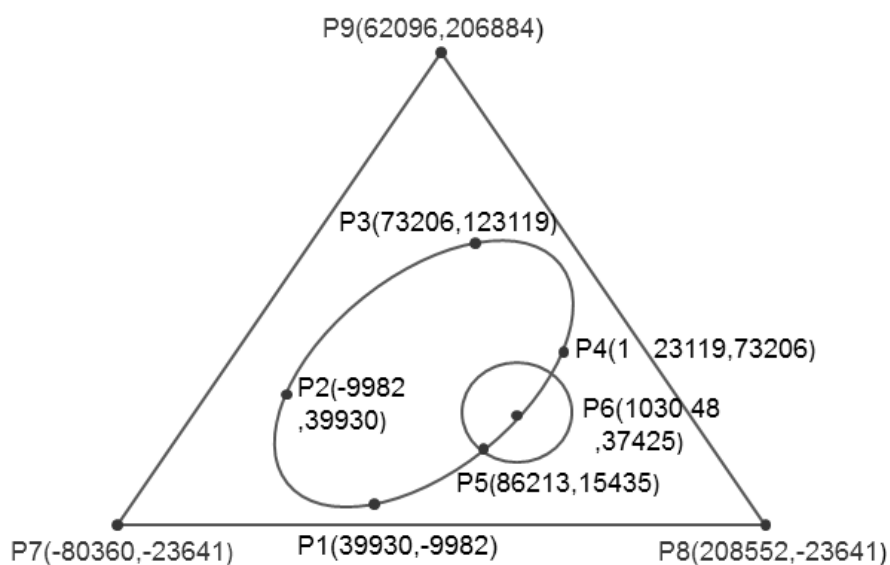
5. Если заданная точка B находится в квадранте QII и QIII точки A, траектория движения по дуге показана жирными линиями на рисунке ниже в зависимости от значений операнда  $S_3$ .



6. Если координата по оси X заданной точки B = 0, а координата по оси Y  $\geq 0$ , точка B считается точкой, находящейся в квадранте QI точки A. Если заданная координата по оси Y < 0, точка B считается точкой, находящейся в квадранте QIII точки A.

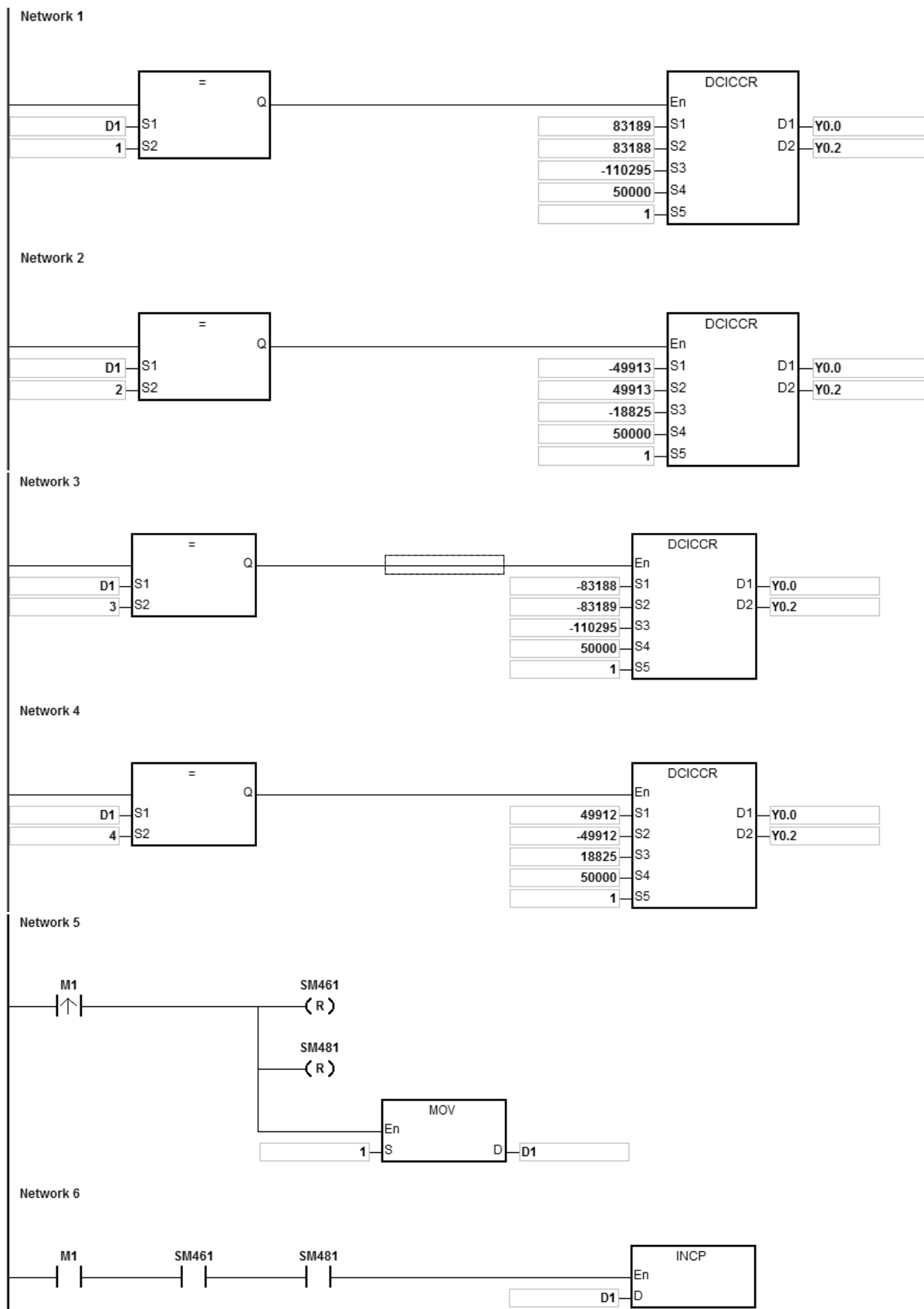
**Пример:**

1. Нарисуем логотип компании DELTA, как показано на рисунке ниже.



2. Шаги: логотип можно поделить на три части.
- Для построения эллипса используется инструкция DCICR, осуществляющая дуговую интерполяцию в относительных координатах против часовой стрелки.
  - Абсолютные координаты эллипса: P1 (39930, -9982), P4 (123119, 73206), P3 (73206, 123119) и P2 (-9982, 39930)
  - Возьмем точку (39930, -9982) в качестве начальной и получим относительные координаты: (83189, 83188),

(-49913, 49913), (-83188, -83189) и (49912, -49912).



6

Когда ПЛК работает и M1 = ON, контроллер начинает построение первого сегмента дуги по часовой стрелке с частотой 50 кГц. Значение операнда D1 увеличивается на 1 по завершении построения сегмента дуги и автоматически начинается построение второго сегмента дуги. Схема повторяется, пока не будет построен четвертый сегмент дуги.

Когда D1=1, при помощи инструкции DCICCR ПЛК выполняет построение первого сегмента дуги от P1 до P4 со сдвигом центра окружности: -110295 и разрешение дуги: 5.

Когда D1=2, при помощи инструкции DCICCR ПЛК выполняет построение второго сегмента дуги от P4 до P3 со сдвигом центра окружности: -18825 и разрешение дуги: 5.

Когда D1=3, при помощи инструкции DCICCR ПЛК выполняет построение третьего сегмента дуги от P3 до P2 со сдвигом центра окружности: -110295 и разрешение дуги: 5.

Когда D1=4, при помощи инструкции DCICCR ПЛК выполняет построение четвертого сегмента дуги от P2 до P1 со сдвигом центра окружности: -18825 и разрешение дуги: 5.

- Для построения окружности используется инструкция DCCMR, которая осуществляет построение окружности в относительных координатах. Подробнее см. пример в инструкции API 2712 DCICR.
- Для построения треугольника используется инструкция DPRMA, осуществляющая 2-осевую синхронизацию в абсолютных координатах. Подробнее см. пример в инструкции API 2712 DCICR.



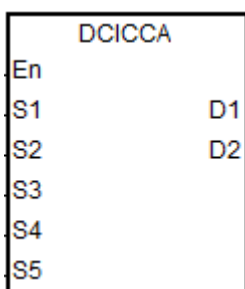
API	Инструкция			Операнд								Описание				
2715	D	CICCA		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, S<sub>5</sub>, D<sub>1</sub>, D<sub>2</sub></b>								2-осевая дуговая интерполяция против часовой стрелки в абсолютных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>							●	●	●		○		○	○		
S <sub>2</sub>							●	●	●		○		○	○		
S <sub>3</sub>							●	●	●		○		○	○		
S <sub>4</sub>							●	●	●		○		○	○		
S <sub>5</sub>							●	●	●		○		○	○		
D <sub>1</sub>		○														
D <sub>2</sub>		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>			●				●						
S <sub>2</sub>			●				●						
S <sub>3</sub>			●				●						
S <sub>4</sub>			●				●						
S <sub>5</sub>			●				●						
D <sub>1</sub>	●												
D <sub>2</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символьное обозначение:**



- S<sub>1</sub> : Заданная координата по оси X (абсолютное позиционирование)
- S<sub>2</sub> : Заданная координата по оси Y (абсолютное позиционирование)
- S<sub>3</sub> : Смещение центра
- S<sub>4</sub> : Заданная опорная частота
- S<sub>5</sub> : Функция
- D<sub>1</sub> : Импульсный выход для оси X
- D<sub>2</sub> : Импульсный выход для оси Y

**Описание:**

- Операнды S<sub>1</sub> и S<sub>2</sub> определяют заданные координаты по осям X и Y соответственно. Подробнее см. описание инструкции DCICCR.
- См. Пример 2 из инструкции DPPMR (API 2710) для программирования на языке ST.

API	Инструкция			Операнд								Описание				
2716	D	CCMR		$S_1, S_2, S_3, S_4, D_1, D_2$								Построение окружности в относительных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
$S_4$							●	●	●		○		○	○		
$D_1$		○														
$D_2$		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

Символьное обозначение:

DCCMR	
$E_n$	
$S_1$	$D_1$
$S_2$	$D_2$
$S_3$	
$S_4$	

- $S_1$  : Координата центра по оси X (относительное позиционирование)
- $S_2$  : Координата центра по оси Y (относительное позиционирование)
- $S_3$  : Выбор функции
- $S_4$  : Заданная опорная частота
- $D_1$  : Импульсный выход для оси X
- $D_2$  : Импульсный выход для оси Y

Описание:

- Операнды  $S_1$  и  $S_2$  определяют координаты центра окружности по осям X и Y соответственно (относительное положение).
- Операнд  $S_3$  – это выбор функции. Если значение 0, разрешение дуги берется равным  $10^\circ$  в качестве стандартного угла для движения по часовой стрелке. Если значение 1, разрешение дуги берется равным  $5^\circ$  в качестве стандартного угла для движения по часовой стрелке. Если значение 2, разрешение дуги берется равным  $10^\circ$  в качестве стандартного угла для движения против часовой стрелки. Если значение 3, разрешение дуги берется равным  $5^\circ$  в качестве стандартного угла для движения против часовой стрелки.

3. Операнды  $D_1$  и  $D_2$  задают импульсные выходы для осей X и Y соответственно. Выбор дискретных выходов и режимов выходов осей см. в инструкции DPPMR.

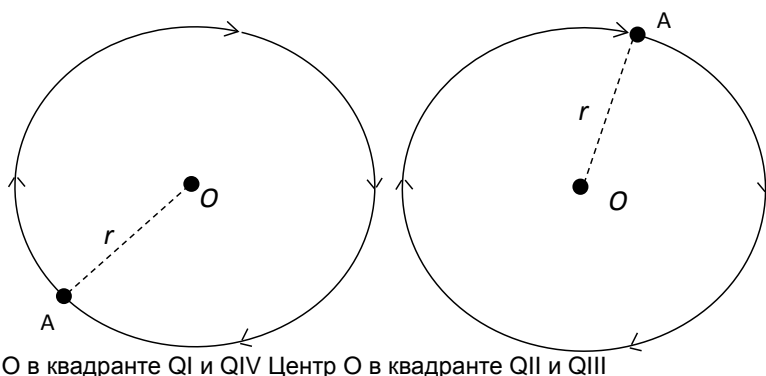
4. Операнд  $S_4$  определяет заданную опорную частоту. Заданная опорная частота используется для предварительного вычисления, когда контроллер планирует траекторию движения по дуге после запуска инструкции. Если по результатам предварительного расчета построить траекторию движения по дуге не удастся, выходная частота автоматически снижается, чтобы построение дуги было синхронизировано.

Если координата O центра окружности по оси X равна 0, точка O считается находящейся в квадранте QI точки A, если заданная координата по оси Y  $\geq 0$ . Если заданная координата по оси Y  $< 0$ , точка B считается находящейся в квадранте QIII точки A.

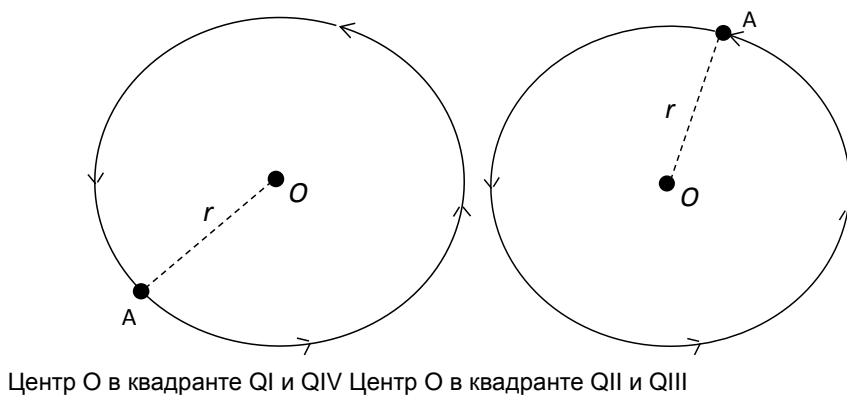
5. См. Пример 2 из инструкции DPPMR (API 2710) для программирования на языке ST.

**Построение окружности по часовой стрелке:**

Точка A представляет текущее положение, точка O является заданным центром окружности, r – это радиус окружности, как показано на рисунке ниже.

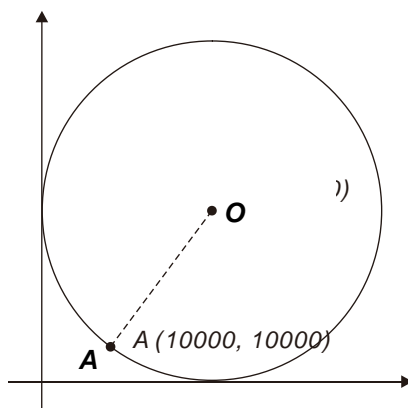


**Построение окружности против часовой стрелки:**



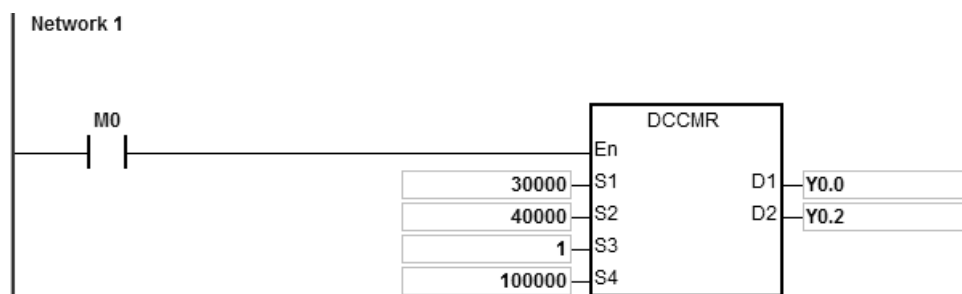
**Пример:**

1. Возьмем координаты (40000, 50000) за центр O и координаты (10000, 10000) за точки A и построим окружность, как показано ниже.



2. Шаги:

- Полученные относительные координаты точки O: (30000, 40000)
- Координаты точки A – это текущее положение; SR460=10000 и SR480=10000
- S<sub>3</sub>=1 и выполняется построение окружности по часовой стрелке с углом 5° в качестве стандартного угла для движения.
- Когда ПЛК работает и M0 = ON, начинается построение окружности в относительных координатах.



3. Работа:

- Когда ПЛК включен и M0 = ON, контроллер начинает построение окружности по часовой стрелке с частотой 100 кГц.
- По завершении построения окружности SM461 и SM481 переходят в состояние ON.

API	Инструкция			Операнд								Описание				
2717	D	CCMA		$S_1, S_2, S_3, S_4, D_1, D_2$								Построение окружности в абсолютных координатах				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$							●	●	●		○		○	○		
$S_2$							●	●	●		○		○	○		
$S_3$							●	●	●		○		○	○		
$S_4$							●	●	●		○		○	○		
$D_1$		○														
$D_2$		○														

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$			●				●						
$D_1$	●												
$D_2$	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

Символьное обозначение:

DCCMA	
$E_n$	
$S_1$	$D_1$
$S_2$	$D_2$
$S_3$	
$S_4$	

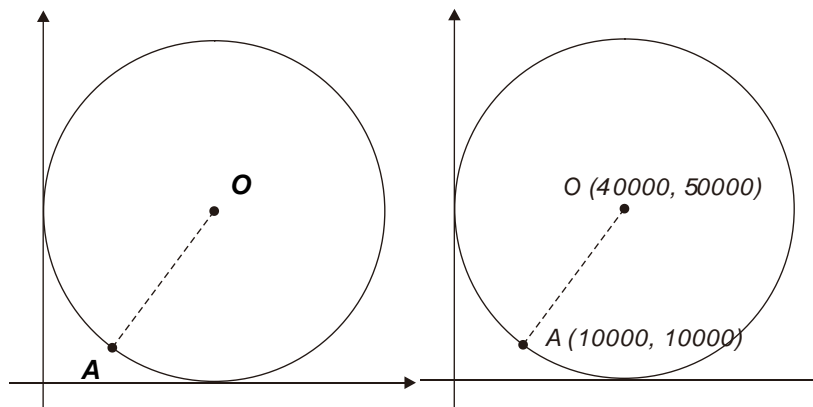
- $S_1$  : Координата центра по оси X (абсолютное позиционирование)
- $S_2$  : Координата центра по оси Y (абсолютное позиционирование)
- $S_3$  : Выбор функции
- $S_4$  : Заданная частота
- $D_1$  : Импульсный выход для X
- $D_2$  : Импульсный выход для Y

Описание:

- Операнды  $S_1$  и  $S_2$  определяют координаты центра окружности по осям X и Y соответственно (абсолютное положение). Подробнее см. описание инструкции DCCMR.
- См. Пример 2 из инструкции DPPMR (API 2710) для программирования на языке ST.

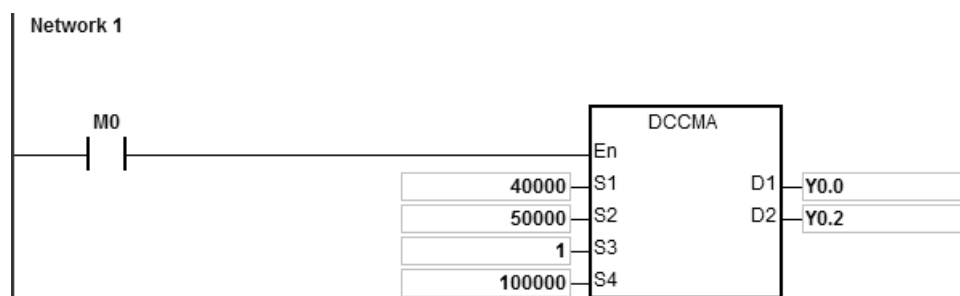
**Пример:**

1. Возьмем координаты (40000, 50000) за центр O и координаты (10000, 10000) за точки A и построим окружность, как показано ниже.



2. Шаги:

- Введем координаты по оси X и Y (40000, 50000) в инструкции
- Точка A – это текущее положение, SR460=10000 и SR480=10000
- S<sub>3</sub>=1, и выполняется построение окружности по часовой стрелке с углом 5° в качестве стандартного угла для движения.
- Когда ПЛК работает и M0 = ON, начинается построение окружности в абсолютных координатах.



3. Работа:

- Когда ПЛК включен и M0 = ON, контроллер начинает построение окружности по часовой стрелке с частотой 100 кГц.
- По завершении построения окружности SM461 и SM481 переходят в состояние ON.

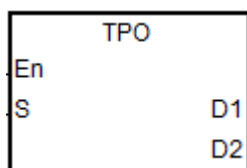
API	Инструкция		Операнд				Описание				
2718		TPO	S, D <sub>1</sub> , D <sub>2</sub>				Управление выходом при помощи таблицы позиционирования				

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S								●				○	○	○		
D <sub>1</sub>								●								
D <sub>2</sub>		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S		●			●	●							
D <sub>1</sub>		●			●	●							
D <sub>2</sub>	●												

Импульсная инструкция	16-битная	32-битная
—	AS	-

Символьное обозначение:



- S** : Номер первого выхода в таблице позиционирования
- D<sub>1</sub>** : Номер выхода, который выдает импульсы
- D<sub>2</sub>** : Флаг выключателя по завершении выдачи импульсов несколькими выходами с номерами по порядку

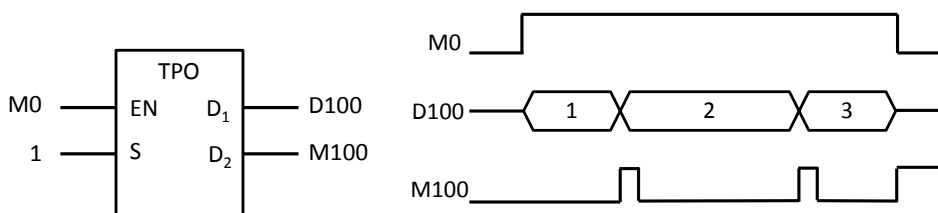
Описание:

- Операнд **S** представляет номер, указанный в таблице позиционирования. Если номер в таблице отсутствует, инструкция не выполняется, SM0 = ON и регистр SR0=16#2027.

No.	Axis No.	Output	ABS/REL mode	Target position	Target speed	Bias speed(Vbi...	Acceleratio
1	Axis 1	Pulse	Reletive	1000000	100000	1000	3000

- Операнд **S** показывает номер первого выхода после включения ПЛК. Если выходов несколько, см. операнд **D<sub>1</sub>**.
- Когда начинается выдача импульсов выходом с одним номером или несколькими выходами с номерами по порядку, флаг выключателя **D<sub>2</sub>** переходит в состояние OFF. Когда очередь доходит до выхода со следующим порядковым номером, операнд **D<sub>2</sub>** переходит в состояние ON, начинается цикл сканирования, и он остается в состоянии ON до завершения выдачи импульсов последним выходом.

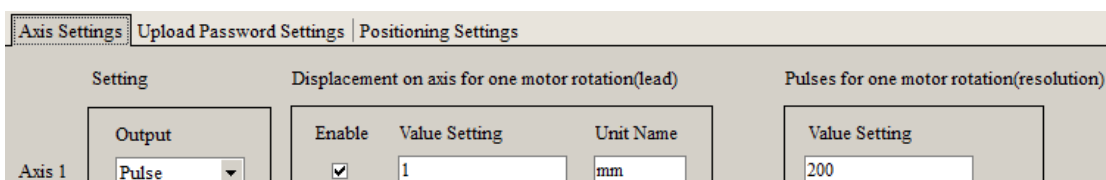
Пример и временный график включения/выключения флага см. ниже.



- На флаг **D<sub>2</sub>** влияет время сканирования ПЛК. Если время для включения следующего по номеру выхода короче времени сканирования, флаг может не успеть вернуться в состояние OFF.

**Пример 1:**

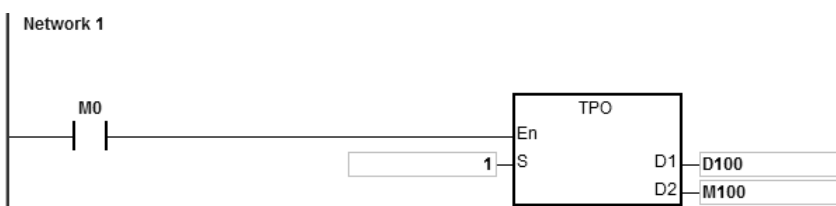
- Включение функции преобразования механического блока



- Задание таблицы планирования позиции и загрузка параметров. Заданная позиция – 2000 (единица измерения: мм). После преобразования в таблицу планирования позиции 400 000 выходных импульсов загружаются и сохраняются в ПЛК.

No.	Axis ...	ABS/REL mode	Target position	Target speed(Hz)	Bias speed(Vbi...	Acceleration ti...
1	Axis 1	Relative	2000	100000	200	200

- Когда M0 включен, Y0.0 выдает 400 000 импульсов на частоте 100 кГц. По завершении вывода M100 включается.



**Пример 2:**

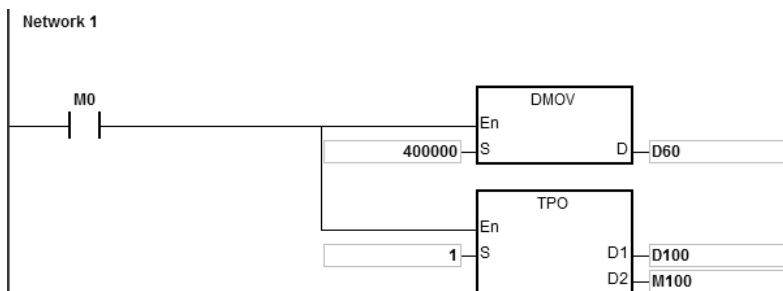
Использование глобального символа или объекта D в задании позиции.

- Использование объекта D для задания позиции в таблице планирования позиции и загрузка параметров. D60 в ПЛК по умолчанию импульсный объект. Если необходимо использовать D для задания позиции и в механическом блоке, необходимо преобразовать механический блок в импульсный.

No.	Axis ...	ABS/REL mode	Target position	Target speed(Hz)	Bias speed(Vbi...	Acceleration ti...
1	Axis 1	Relative(Symbol/Address)	D60	100000	200	200



2. Когда M0 включен, а значение в D60 равно 400 000, Y0.0 выводит 400 000 импульсов на частоте 100 кГц. По завершении вывода M100 включается.



Примечание: устройство является импульсным, когда используются глобальные символы, объекты D или инструкции, включая DTPWS, DTPWL и DTPWC для ввода значений в задание позиции.

3. Если для задания позиции используются глобальные символы

A: Необходимо назначить адрес глобального символа:

Global Symbols				
Global Symbols				
Class	Identifiers	Address	Type...	Initial Value (Activ...
VAR	Target_Axis1	D60	DWORD	N/A

B: Использование глобальных символов для задания позиции в таблице планирования:

Single-axis point-to-point motion						
Single-axis multi-segment motion						
2-axis linear interpolation						
2-axis arc interpolation						
No.	Axis ...	ABS/REL mode	Target position	Target speed(Hz)	Bias speed(Vbi...	Acceleration ti...
1	Axis 1	Relative(Symbol/Address)	Target_axis1	100000	200	200

API	Инструкция			Операнд				Описание			
2719	D	TPWS	P	$S_1, S_2, S_3$				Установка выходных параметров для одной оси в таблице позиционирования			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$								●	●				○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

DTPWS		DTPWSP	
En		En	
S1		S1	
S2		S2	
S3		S3	

- $S_1$  : Номер, указанный в таблице позиционирования
- $S_2$  : Заданное положение
- $S_3$  : Заданная скорость

**Описание:**

- Операнд  $S_1$  представляет номер, указанный в таблице позиционирования. Если номер в таблице отсутствует или номер не относится к перемещению от точки к точке по одной оси или перемещению по нескольким сегментам по одной оси, инструкция не выполняется, SM0 = ON и регистр SR0=16#2027.

No.	Axis No.	Output	ABS/REL mode	Target position	Target speed	Bias speed(Vbi...	Acceleratio
1	Axis 1	Pulse	Reletive	1000000	100000	1000	3000

- Операнд  $S_2$  определяет заданное положение и является 32-битным целым числом. Если во время редактирования таблицы позиционирования в программе используется преобразование механических единиц, см. инструкции по преобразованию.
- Операнд  $S_3$  определяет заданную скорость. Диапазон: 1 Гц ~ 200,000 Гц. (Прим.: максимальная частота выходов осей Y0 и Y2 может быть 4 МГц для AS324MT.)
- При запуске инструкции для изменения параметров одной оси, выход которой передает импульсы, измененные параметры сохраняются в таблице и станут действительными, только когда выход в следующий раз начнет передавать импульсы.

5. Инструкция может изменять параметры только во время работы ПЛК. Последний записанный параметр не сохраняется после выключения питания контроллера. Таблица, отредактированная в программе и загруженная в память ПЛК, считается таблицей позиционирования по умолчанию при последующем включении питания контроллера.

API	Инструкция			Операнд								Описание			
2720	D	TPWL	P	<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub></b>								Установка параметров линейной интерполяции в таблице позиционирования			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
<b>S<sub>1</sub></b>								●	●				○	○		
<b>S<sub>2</sub></b>								●	●				○	○		
<b>S<sub>3</sub></b>								●	●				○	○		
<b>S<sub>4</sub></b>								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>			●				●						
<b>S<sub>2</sub></b>			●				●						
<b>S<sub>3</sub></b>			●				●						
<b>S<sub>4</sub></b>			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

DTPWL	DTPWLP
En	En
S1	S1
S2	S2
S3	S3
S4	S4

- S<sub>1</sub>** : Номер, указанный в таблице позиционирования
- S<sub>2</sub>** : Заданное положение оси X
- S<sub>3</sub>** : Заданное положение оси Y
- S<sub>4</sub>** : Заданная скорость

**Описание:**

- Операнд **S<sub>1</sub>** представляет номер, указанный в таблице позиционирования. Если номер в таблице отсутствует или номер выхода не относится к линейной интерполяции по двум осям, инструкция не выполняется, SM0 = ON и регистр SR0=16#2027.

No.	Coord.	Axis No.	Output	ABS/REL mode	Target position	Target speed	ACC/DEC time
1	X	Axis 1	Pulse/Dir	Absolute	1000	50000	100
T1	Y	Axis 2	Pulse/Dir		-1000		

- Операнды **S<sub>2</sub>** и **S<sub>3</sub>** определяют заданные положения осей X и Y соответственно и могут быть только 32-битными целыми числами. Если во время редактирования таблицы позиционирования в программе используется преобразование механических единиц, см. инструкции по преобразованию.

3. Операнд **S<sub>4</sub>** определяет заданную скорость. Диапазон: 1 Гц ~ 200,000 Гц.
4. При выполнении линейной интерполяции частота, заданная операндом **S<sub>4</sub>**, автоматически соответствует выходу оси, которая находится дальше всего от заданного положения. Если оси X и Y не могут одновременно выйти на заданные положения, ПЛК автоматически понижает частоту, чтобы две оси могли выйти на заданные положения одновременно.
6. При запуске инструкции для изменения параметров двух осей, выход одной из которых передает импульсы, измененные параметры двух осей сохраняются в таблице и станут действительными, только когда выход двух осей в следующий раз начнет передавать импульсы.
5. Инструкция может изменять параметры только во время работы ПЛК. Последний записанный параметр не сохраняется после выключения питания контроллера. Таблица, отредактированная в программе и загруженная в память ПЛК, считается таблицей позиционирования по умолчанию при последующем включении питания контроллера.

API	Инструкция			Операнд								Описание			
2721	D	TPWC	P	$S_1, S_2, S_3, S_4, S_5$								Установка параметров дуговой интерполяции в таблице позиционирования			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
$S_1$								●	●				○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		
$S_4$								●	●				○	○		
$S_5$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						
$S_4$			●				●						
$S_5$			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
AS	-	AS

**Символьное обозначение:**

DTPWC		DTPWCP		
En	En			$S_1$ : Номер, указанный в таблице позиционирования
S1	S1			$S_2$ : Координата центра по оси X
S2	S2			$S_3$ : Координата центра по оси Y
S3	S3			$S_4$ : Смещение центра
S4	S4			$S_5$ : Заданная скорость
S5	S5			

**Описание:**

- Операнд  $S_1$  представляет номер, указанный в таблице позиционирования. Если номер в таблице отсутствует или номер выхода не относится к дуговой интерполяции по двум осям, инструкция не выполняется, SM0 = ON и регистр SR0=16#2027.

Single-axis point-to-point motion								Single-axis multi-segment motion		2-axis linear interpolation		2-axis arc interpolation	
No.	Coord.	Axis No.	Output	ABS/REL mode	Target position	Target speed	Center shift value						
1	X	Axis 1	Pulse/Dir	Absolute	100000	15000	-50000						
T1	Y	Axis 2	Pulse/Dir		100000								

2. Операнды **S<sub>2</sub>** и **S<sub>3</sub>** определяют координаты центра по осям X и Y соответственно. Операнд **S<sub>4</sub>** задает сдвиг центра. Три параметра могут быть только 32-битными целыми числами. Если во время редактирования таблицы позиционирования в программе используется преобразование механических единиц, см. инструкции по преобразованию.
3. Операнд **S<sub>5</sub>** определяет заданную скорость. Диапазон: 1 Гц ~ 200,000 Гц. Это опорная частота для выдачи импульсов. ПЛК автоматически понижает частоту, если не удастся выполнять синхронно по двум осям.
4. При запуске инструкции для изменения параметров двух осей, выход одной из которых передает импульсы, измененные параметры двух осей сохраняются в таблице и станут действительными, только когда выход двух осей в следующий раз начнет передавать импульсы.
5. Инструкция может изменять параметры только во время работы ПЛК. Последний записанный параметр не сохраняется после выключения питания контроллера. Таблица, отредактированная в программе и загруженная в память ПЛК, считается таблицей позиционирования по умолчанию при последующем включении питания контроллера.

Инструкция не поддерживает изменение направления движения (по часовой и против часовой стрелки). Для смены направления необходимо добавить или изменить параметры выхода для дуговой интерполяции в таблице позиционирования в программе.

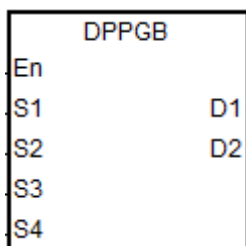
API	Инструкция			Операнд								Описание			
2723	D	PPGB		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, D<sub>1</sub>, D<sub>2</sub></b>								Переход от точки к точке вперед и назад			

Регистр	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	«\$»	F
S <sub>1</sub>								●								
S <sub>2</sub>								●								
S <sub>3</sub>								●								
S <sub>4</sub>								●								
D <sub>1</sub>		○														
D <sub>2</sub>			●													

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>			●				●						
S <sub>2</sub>			●				●						
S <sub>3</sub>			●				●						
S <sub>4</sub>									●				
D <sub>1</sub>	●												
D <sub>2</sub>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	-	AS

**Символьное обозначение:**



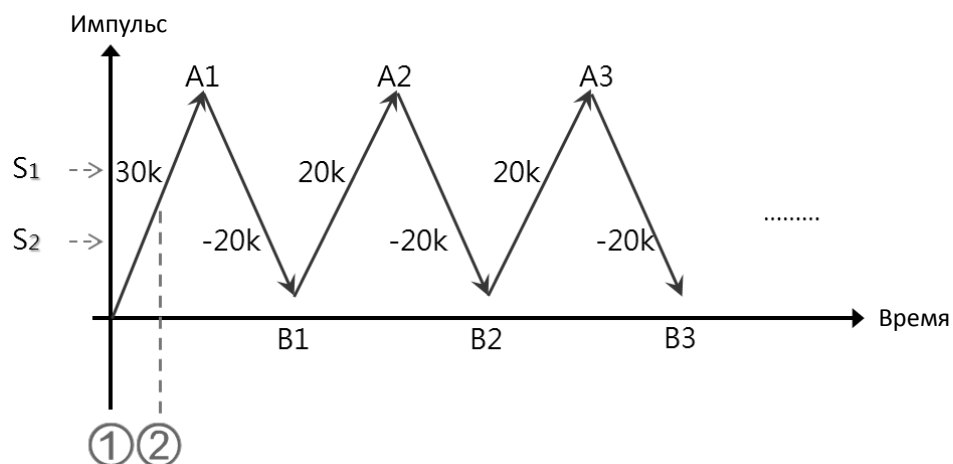
- S<sub>1</sub>** : Относительное заданное положение A
- S<sub>2</sub>** : Относительное заданное положение B
- S<sub>3</sub>** : Заданная частота
- S<sub>4</sub>** : Настраиваемый коэффициент (передаточное число) заданной частоты (точка с плавающей запятой)
- D<sub>1</sub>** : Импульсный выход
- D<sub>2</sub>** : Индикатор возможности изменения заданной частоты

**Описание:**

- Инструкция высокоскоростного выхода предназначена для перемещения вперед и назад между двумя заданными в относительных координатах положениями, которые задаются в операндах **S<sub>1</sub>** и **S<sub>2</sub>**. Данная инструкция подходит для сновальных машин в текстильной промышленности, наматывающих машин на кабельном производстве и т. д.
- Перед запуском выхода инструкцией необходимо задать относительные положения в операндах **S<sub>1</sub>** и **S<sub>2</sub>**, чтобы инструкция могла выполнить предварительный расчет следующего перехода. После включения



инструкции можно изменить заданные положения, которые будут использоваться впоследствии, но текущие заданные положения, по которым уже идет выдача импульсов, изменить уже нельзя.



① Инструкция DPPGB включена. Если  $S_1 = 30,000$  и  $S_2 = -30,000$ , выполняется движение вперед в заданную точку  $A1=30,000$ .

② Если импульсов  $>15,000$ , относительные положения превращаются в  $S_1: 20,000$  и  $S_2: -20,000$ , а заданное положение будет  $A1=30,000$ . Заданные положения, полученные вычислениями, следующие:  $B1=10,000$  ( $A1 + S_2$ ) и  $A2=30,000$  ( $B1 + S_1$ ), потому что текущие заданные положения, по которым уже идет выдача импульсов, изменить нельзя.

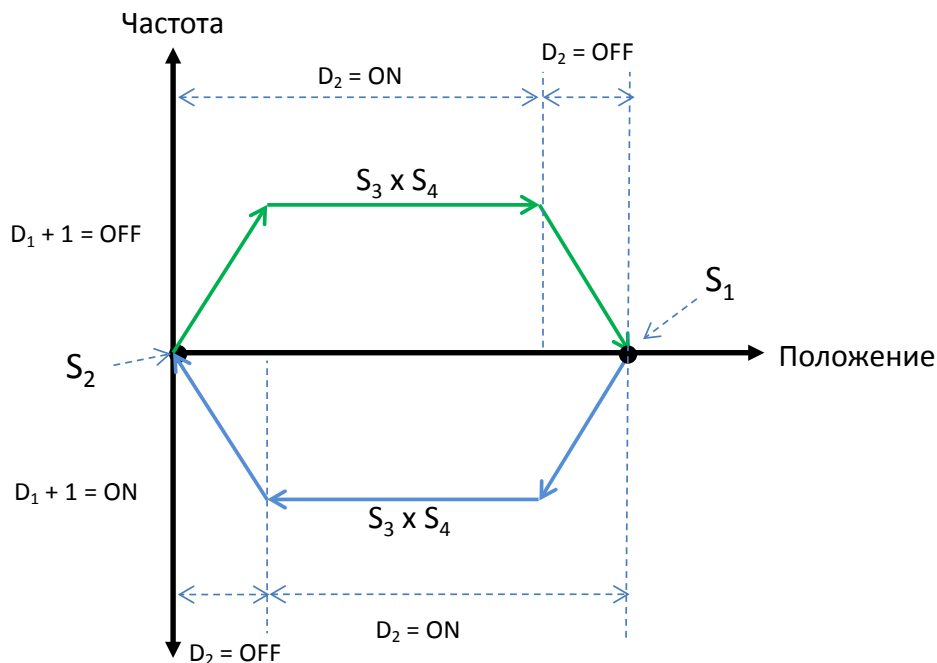
3. Операнд  $S_3$  определяет заданную частоту (32-битное целое число), а операнд  $S_4$  – настраиваемый коэффициент заданной частоты (точка с плавающей запятой). Фактическая частота – это результат перемножения значений операндов  $S_3$  и  $S_4$  с последующим округлением результата до 32-битного целого числа. Допустимый диапазон значений входной частоты от 1 до 200 кГц. Если значение вне диапазона, ПЛК автоматически берет минимальное или максимальное значение диапазона. Формула отношения фактической заданной частоты =  $S_3 \times S_4$ . Например, заданная частота 1 кГц, настраиваемое значение с плавающей запятой 1.2345, тогда фактическая частота будет 1234 Гц.
4. Если идет выдача импульсов, заданную частоту и ее коэффициент можно изменять, и тогда результат будет обновлен после сканирования инструкции. При этом предполагается, что заданная частота не будет меняться слишком сильно, иначе это повлияет на вычисленное снижение частоты.
5. В операнде  $D_1$  можно указывать только Y0.0, Y0.2, Y0.4, Y0.6, Y0.8 и Y0.10. Операнд  $D_1$  занимает 2 дискретных выхода подряд. Наборы выходов и режимов специальных регистров см. в таблице ниже.

Номер оси	1	2	3	4	5	6
Выход $D_1+0$	Y0.0	Y0.2	Y0.4	Y0.6	Y0.8	Y0.10
Выход $D_1+1$	Y0.1	Y0.3	Y0.5	Y0.7	Y0.9	Y0.11
Режимы выходов	SR462	SR482	SR502	SR522	SR542	SR562

6. Операнд  $D_2$  – это флаг, показывающий возможность изменения заданной частоты и ее коэффициента. Если флаг стоит, значит заданная частота может быть изменена. Если флаг снят, значит сейчас идет понижение частоты и изменение заданной частоты вступит в силу при следующем переходе.

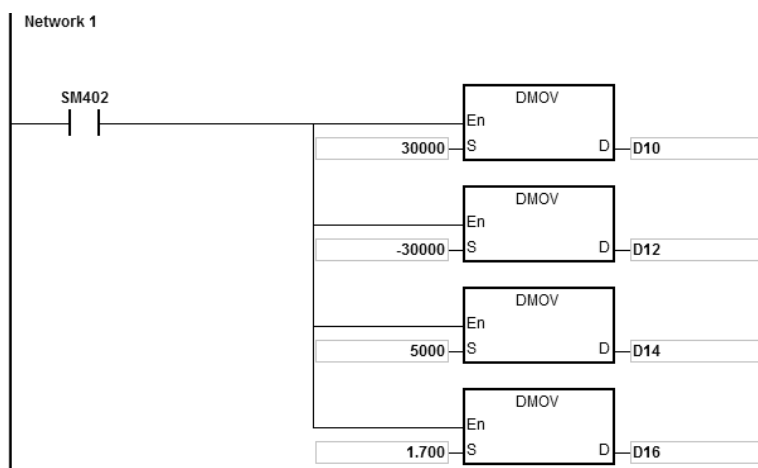
7. Ограничений по количеству исполнений инструкции не существует. Но во время выполнения задействованная инструкцией высокоскоростная ось не может использоваться другими инструкциями. Иначе инструкция не будет выполняться.

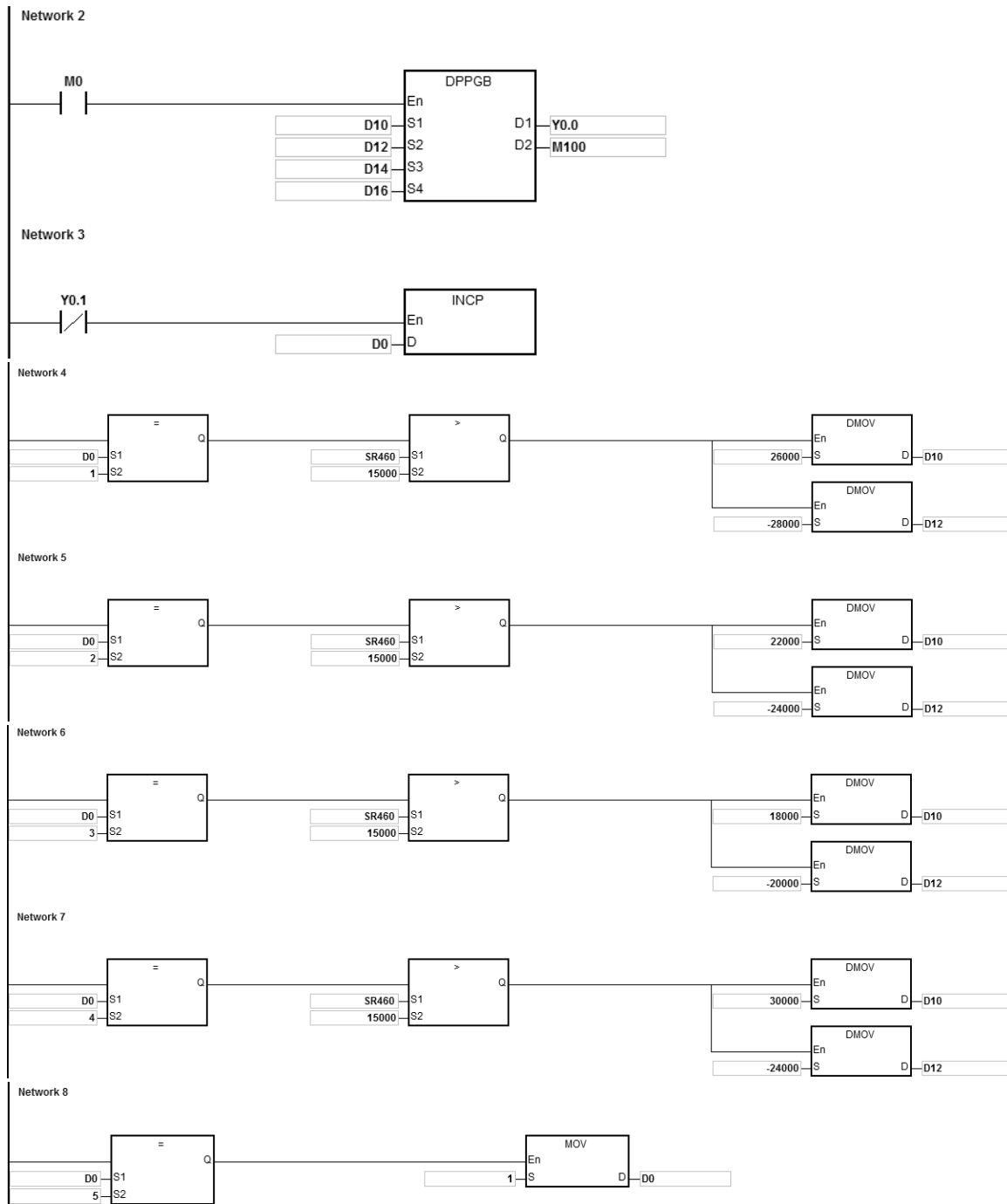
Временная диаграмма работы выхода:



Пример:

Если M0 = ON, инструкция DPPGB начинает движение назад и вперед между двумя заданными положениями, вычисленными по двум относительным положениям, заданным операндами S<sub>1</sub> и S<sub>2</sub>, с фактической частотой 8500 Гц (5000\*1.7). Y0.0 – это выход. Y0.1 = OFF - это значит положительное направление.

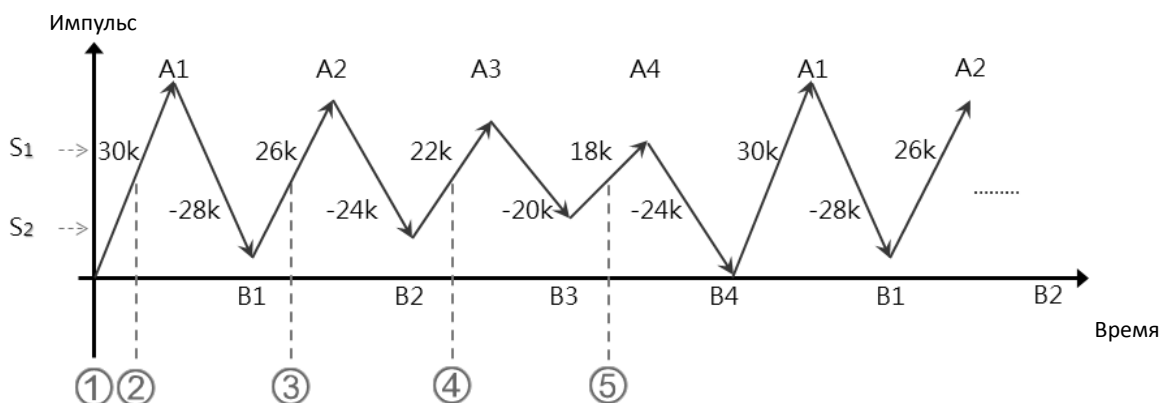




**Описание:**

1. Когда по команде инструкции выход начал выдавать импульсы, измените относительное заданное положение A для операнда **S<sub>1</sub>** и B для операнда **S<sub>2</sub>**.

а) График выхода Y0.0:



б) Описание:

① Если **S<sub>1</sub>**: 30,000, **S<sub>2</sub>**: -30,000 и контакт M0 запустит инструкцию DPPGB, начинается движение в положение A1=30,000.

② Если D0=1 и положение: количество импульсов в регистре SR460 >15,000, относительные положения меняются на **S<sub>1</sub>**: 26,000 и **S<sub>2</sub>**: -28,000.

В момент, когда заданное положение A1=30,000, заданное положение уже изменить нельзя. Заданные положения, полученные вычислениями: B1=2,000 (A1-28,000), A2=28,000 (B1+26,000).

③ Если D0=2 и положение: количество импульсов в регистре SR460 >15,000, относительные положения меняются на **S<sub>1</sub>**: 22,000 и **S<sub>2</sub>**: -24,000.

В момент, когда заданное положение A2=28,000, заданное положение уже изменить нельзя.

Заданные положения, полученные вычислениями: B2=4,000 (A2-24,000), A3=26,000 (B2+22,000).

④ Если D0=3 и положение: количество импульсов в регистре SR460 >15,000, относительные положения меняются на **S<sub>1</sub>**: 18,000 и **S<sub>2</sub>**: -20,000.

Заданные положения, полученные вычислениями: B3=6,000 (A3-20,000), A4=24,000 (B3+18,000).

⑤ Если D0=4 и положение: количество импульсов в регистре SR460 >15,000, относительные положения меняются на **S<sub>1</sub>**: 30,000 и **S<sub>2</sub>**: -24,000.

Заданные положения, полученные вычислениями: B4=0 (A4-24,000) и A1=30,000 (B4+30,000).

Шаг	S <sub>1</sub> : относительное положение	S <sub>2</sub> : относительное положение	Заданное положение B	Заданное положение A
①	30,000	-30,000	-	A1=30,000
②	26,000	-28,000	B1=2,000	A2=28,000
③	22,000	-24,000	B2=4,000	A3=26,000
④	18,000	-20,000	B3=6,000	A4=24,000
⑤	30,000	-24,000	B4=0	A1=30,000

## 6.27 Инструкции коммуникации CANopen

### 6.27.1 Описание инструкций коммуникации CANopen

API	Код инструкции			Операнды								Функция				
2800		INITC		<b>S</b>								Инициализация приводов для связи по CANopen				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
<b>S</b>													○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S</b>		●											

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

Символ:

INITC
En
S

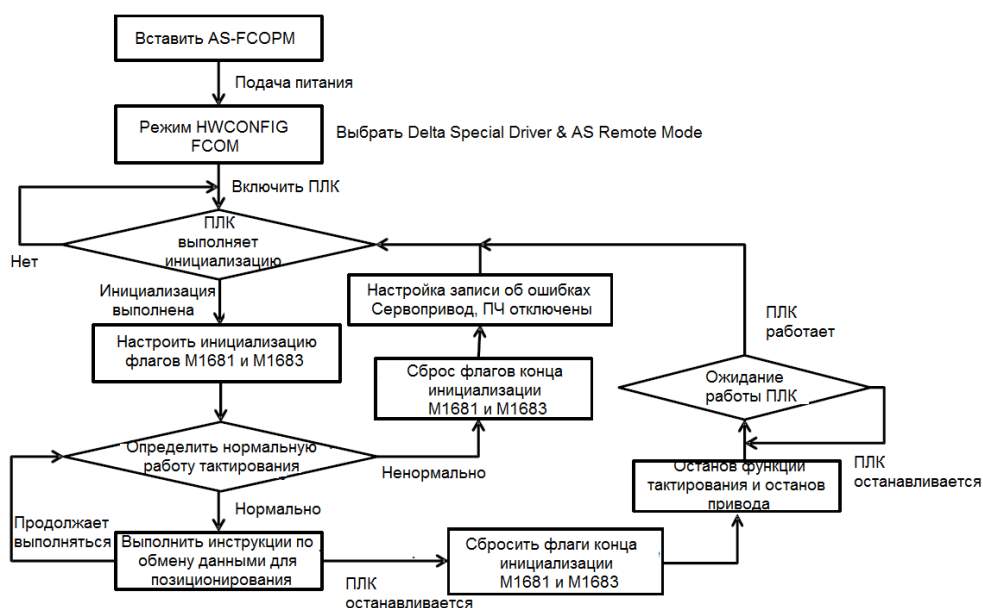
**S** : Число сервоприводов для инициализации

Описание:

1. Перед выполнением инструкции убедитесь, что в ПО аппаратной конфигурации HWCONFIG как рабочий режим функциональной платы 2 выбран «Delta ASD-A2 control» (Управление сервоприводом Delta ASD-A2).
2. Диапазон **S**: 1~8. Если значение больше 8, ПЛК автоматически настроит значение для инициализации **S** на 8. Заданные адреса станций могут начинаться только с 1, а номера между 1 и 8 не могут быть пропущены или зарезервированы.
3. Для версии прошивки V1.02.00 или выше, для ПЧ добавляются значения **S** от 21 до 28. Для значения, которое не находится в диапазоне от 21 до 28, эта инструкция не будет выполнена. То же правило применяется к ПЧ; адрес станции должен начинаться с 21 и следующий адрес должен быть последовательно по порядку.
4. При установке номера станции до 5 эта инструкция инициализирует адрес станции от 1 до 5. И при установке номера станции до 23, эта инструкция инициализирует адрес станции от 21 до 28. Когда есть 5 сервоприводов и 3 ПЧ, которые необходимо инициализировать, необходимо выполнить эту инструкцию дважды.

5. При выполнении этой команды для сервоприводов флаг SM1681 выключается (S= 1-8). При инициализации SM1681 включается. SM1683 отключается, инструкцию выполняется для ПЧ (S = 21-28). SM1683 включается при завершении инициализации.
6. Флаг SM1682 включается при возникновении ошибки в процессе коммуникации. При этом в регистре SR658 будет сохранен номер привода, в котором произошла ошибка, а в регистре SR659 будет сохранен код ошибки.
7. Если после завершения инициализации происходит ошибка тактирования (SR0 = 0x1900 ~ 0x191C, последние 2 кода – идентификатор Slave устройства), а SM1684 выключен (по умолчанию, при отключении одного, отключаются все), флаги завершения инициализации SM1681 и SM1683 будут сброшены и связь с другими Slave устройствами также будет приостановлена. После устранения всех проблем необходимо инициализировать каждое Slave устройство, чтобы перезапустить операцию. ПЛК выключит индикатор ошибки после того, как подтвердит завершение инициализации. Эта ошибка будет записана в журнале ошибок.
8. Если оси работают независимо и связь работает правильно, можно включить SM11684 (указывая, что отключается только неисправный привод), чтобы ПЛК записал конкретную ошибку в журнале ошибок, а другие Slave устройства могли продолжать работать.
9. Необходимо настроить станцию связи и скорость приводов (серво или ПЧ) вручную, а затем выключить и снова включить приводы, чтобы активировать новые параметры.
10. Эта инструкция не должна использоваться на языке программирования ST, программах прерывания или однократно вызываемых функциональных блоках.

#### Блок-схема инициализации и работы:



**Пример:** (Коммуникация с сервоприводом Delta ASD-A2M)

1. Перед применением инструкций связи по CANopen поместите плату AS-FCOPM в ПЛК (только слот 2) и настройте в ПО аппаратной конфигурации HWCONFIG «Delta ASD-A2 control» как рабочий режим функциональной платы 2. Скорость обмена данными задается как 125, 250, 500 или 1000 кбит/с.

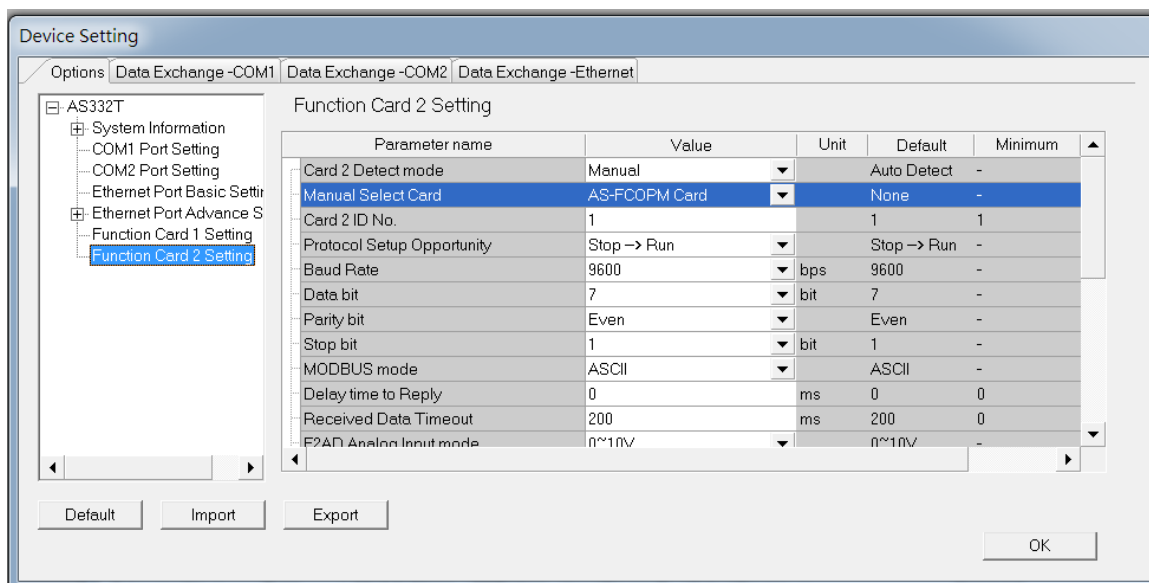


Рис. 1 настройка функциональной платы 2

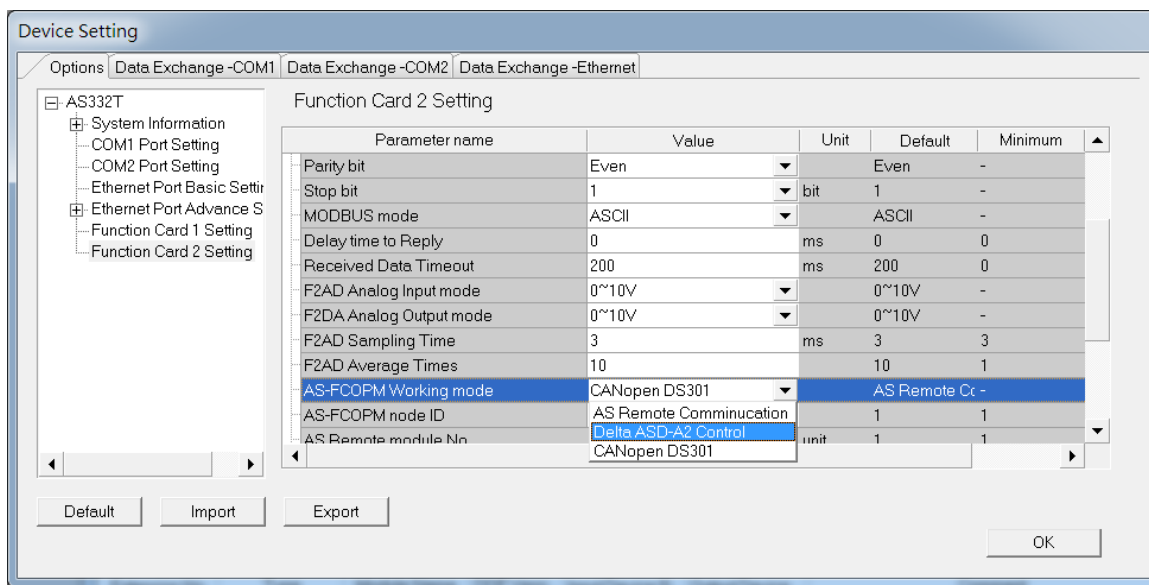


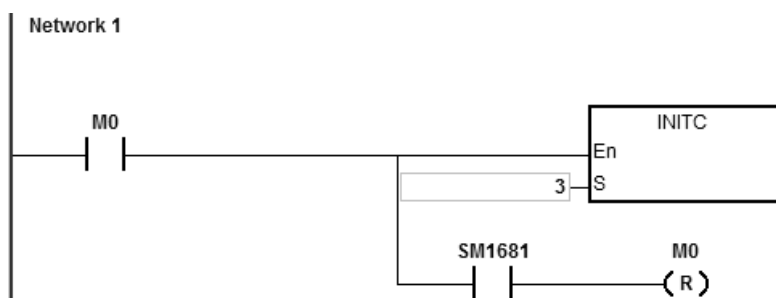
Рис. 2 Выбор рабочего режима

2. Подключите ПЛК серии AS к сервоприводу ASDA с помощью кабеля связи CANopen как показано на рисунке ниже (**необходимо включить терминальный резистор 120 Ом на плате AS-FCOPM**).





3. Выполните следующие действия для базовой настройки на пульте ASD-A2.
  - a. Задайте параметр сервопривода P2-08 как 10 для сброса на заводские значения.
  - b. Отключите и вновь подайте питание на сервопривод.
  - c. Задайте значение параметра P1-01 как 0001 (PR режим).
  - d. Задайте значение параметра P3-01 как 0400 и скорость обмена данными по коммуникации CAN 1.0 Мбит/с (такое же значение должно быть задано в ПЛК).
  - e. Установите адрес станции для каждого сервопривода на основе количества сервоприводов. P3-00 для каждого сервопривода устанавливается в порядке 1, 2 и 3. Могут быть установлены максимально 8 сервоприводов.
  - f. Отключите и вновь подайте питание на сервопривод.
  - g. Работа сервоприводов начнется после завершения базовой настройки.
4. Загрузите программу из примера и включите M0. Сервоприводы с адресами станций 1 ~ 3 будут инициализированы. SM1681 включается, это означает, что инициализация завершена (когда сервопривод успешно войдет в режим CANOpen, на дисплее отобразится информация CO-LD).



5. Настройки сервопривода в процессе инициализации их ПЛК серии AS показаны ниже.
  - a. Задайте P2-30 (вспомогательная функция) как 5. Это означает, что сервоприводу не нужно сохранять заданные значения в EEPROM, что продлевает срок службы сервопривода.
  - b. Сбросьте P6-02 (PATH#1) и P6-06(PATH#3) на 0 (очистка PATH#1 и #3 в режиме PR).
  - c. Задайте P3-06 (источник SDI) как 16#0100 (DI1 ~ DI8 управляются аппаратно, EDI9 управляется

- программно и EDI10 ~ EDI14 управляются аппаратно).
- d. Сбросьте P4-07 (ручное управление SDI) на 0.
  - e. Задайте P2-36 (EDI9) как 16#0101 (функция EDI9 выбрана как Servo ON).
  - f. Задайте P0-17 (CM1A) как 1 (параметр отображения является выходным регистром команд CMD\_O).
  - g. Задайте P0-18 (CM2A) как 64 (параметр отображения является регистром импульсных команд CMD\_E).
  - h. Задайте P5-20~P5-35 (время разгона) как 1 (время разгона 1 мс).
  - i. Задайте P5-60~P5-75 (заданная скорость) как 1 (заданная скорость будет 0,1 об/мин).
  - j. Создайте PDO1 соответственно с P5-07 (команда PR), P0-01(код ошибки) и P0-46 (состояние дискретного выхода).
  - k. Создайте PDO2 соответственно с P0-09 (состояние CM1: CMD\_O) и P0-10(состояние CM2: CMD\_E).
6. Для параметров сервопривода, установленных в шести пунктах: a, b, f, g, j и k, не используйте для их изменения инструкцию COPRW.
7. Когда используется сервопривод с абсолютным энкодером, необходимо использовать инструкцию коммуникации COPRW для записи 16#0100 в P3-12, чтобы соответствующие параметры сервопривода записывались в EEPROM в момент отключения питания.
8. Чтобы изменить настройку аппаратного сигнала дискретного входа сервопривода ASD-A2, можно задать соответствующие параметры конфигурации сигнала дискретного входа вручную или с помощью инструкции COPRW. COPRW может использоваться для изменения конфигурации после завершения выполнения инструкции INITC и до включения сервопривода.

#### Программирование на языке ST:

```
0001 IF MO = TRUE AND SM1681 = FALSE THEN
0002     INITC (3) ;
0003 END_IF ;
0004 IF MO = TRUE AND SM1681 = TRUE THEN
0005     MO := FALSE ;
0006 END_IF ;
```

9. В следующем списке показаны настройки для инициализации сервопривода серии AS.
- A. Установите P2-30 (вспомогательная функция) на 5, чтобы указать, что сервоприводу не нужно сохранять настройки в EEPROM постоянно. Это может продлить срок службы сервопривода.
  - B. Сбросьте P6-02 (PATH#1) и P6-06 (PATH#3) на 0. Это означает, что PATH#1 и #3 в режиме PR сброшены.
  - C. Установите P3-06 (источник SDI) на 16#0100. Это означает, что DI1-DI8 управляется аппаратным

обеспечением, EDI9 управляется программным обеспечением, а EDI10-EDI14 управляется аппаратным обеспечением.

D. Сбросьте P4-07 (состояние SDI контролируется вручную) на 0.

E. Установите P2-36 (EDI9) на 16#0101. Это означает, что функция EDI9 установлена на Servo-ON.

F. Установите P0-17 (CM1A) на 1. Это означает, что параметр отображения является выходным регистром команды CMD\_O.

G. Установите P0-18 (CM2A) на 64. Это означает, что параметром отображения является регистр команд импульса CMD\_E.

H. Установите P5-20-P5-35 (время разгона) на 1. Это означает, что время разгона составляет 1 мс.

I. Установите P5-60-P5-75 (заданная скорость) на 1. Это означает, что заданная скорость составляет 0,1 об/мин.

J. Установите PDO1 как соответственно P5-07 (команда PR), P0-01 (код ошибки), P0-46 (состояние точки DO) и P4-07 (состояние точки DI)

K. Установите PDO2 как соответственно P0-09 (состояние CM1: CMD\_O) и P0-10 (состояние CM2: CMD\_E).

L. Установите PDO3 как соответственно P0-11 (состояние CM3: состояние момента)

M. Установите время тактирования и время сканирования ПЛК как 200 мс (по умолчанию). Когда связь прерывается, ПЛК отправит сообщение об ошибке (индикатор ошибки медленно мигает, см. код ошибки для получения дополнительной информации) и сервопривод автоматически выключается.

Примечание. Состояние текущего момента и точки DI поддерживает версия прошивки V1.04.10 или выше.

10. Не используйте инструкцию COPRW (API 2807) для изменения параметров сервопривода для шести элементов A, B, F, G, J, K и M.
11. Когда вы используете сервопривод абсолютного типа, используйте инструкцию связи COPRW для записи 16#0100 в P3-12, которая записывает соответствующие сервопараметры абсолютного типа в EEPROM в момент отключения сервопривода.
12. Установите соответствующие параметры конфигурации сигнала дискретного входа DI вручную или с помощью инструкции COPRW, чтобы изменить настройку аппаратного сигнала ASD-A2. Используйте COPRW для изменения конфигурации после завершения выполнения инструкции INITC и до включения сервопривода.
13. Когда инициализация завершена, сервопривод находится в режиме PR. Не выполняйте управление связью с сервоприводом.

14. Детальные сведения см. Руководство по эксплуатации сервопривода Delta ASD-A2.

#### Пример связи с ПЧ Delta:

1. Вручную установите параметры преобразователя P09-36 (в диапазоне от 21 до 28), P09-37 (скорость обмена данными по умолчанию - 1 МГц) и P09-40 – по стандартному протоколу CANopen (по умолчанию).

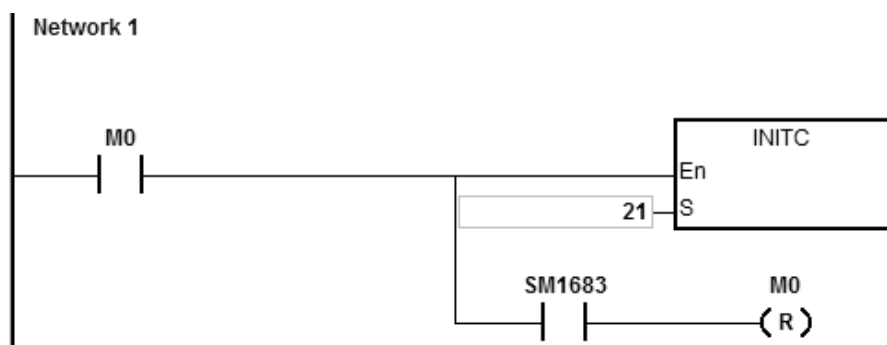
2. Вручную установите источник задания частоты от P00-20 на 6 и источник команды управления P00-21 на 3 и обеспечьте их управление по протоколу CANopen.

3. В следующем списке показаны настройки для инициализации ПЧ с ПЛК AS.

A. Установите область отображения PDO: рабочие частоты, текущее состояние и состояния момента и точка дискретного входа DI соответствуют SRS ПЛК.

B. Установите время тактирования и время ожидания сканирования ПЛК на 200 мс (по умолчанию). Когда происходит превышение времени связи, ПЛК отправляет сообщение об ошибке (светодиод ошибки мигает медленно, см. Код ошибки для получения дополнительной информации), и сервопривод отключается автоматически.

4. Более подробную информацию о параметрах сервомеханизма см. В руководстве по эксплуатации ПЧ Delta.



Когда M0 включен, инструкция INITC начинает инициализировать ПЧ по адресу станции 21, пока не будет включен SM1683. Когда SM1683 включен, это означает, что инициализация завершена.

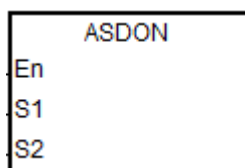
API	Код инструкции		Операнды										Функция			
2801		ASDON	$S_1, S_2$										Запуск и отключение приводов			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●		○	○	○	○		
$S_2$													○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

**Символ:**



- $S_1$  : Адрес станции привода
- $S_2$  : Сигналы привода ВКЛ и ВЫКЛ

**Описание:**

1. Выполнение инструкции INITS должно быть закончено до выполнения данной инструкции.
2. Диапазон  $S_1$  составляет 1-8 (для сервопривода) и 21-28 (для ПЧ). Когда значение выходит за пределы диапазона, инструкция не будет выполняться.
3. Если значение в  $S_1$  составляет 1-8 (для сервопривода) и в  $S_2$  не равно 0, сервопривод готов к пуску (Servo-ON). Если значение в  $S_2$  равно 0, сервопривод отключен (Servo-OFF).
4. Диапазон  $S_1$  составляет 21-28 (для ПЧ). Когда  $S_2$  равно 1, ПЧ находится в режиме скорости а. Когда  $S_2$  равно 2, ПЧ находится в режиме момента. Когда  $S_2$  равно 0, ПЧ перестает работать. Поскольку режимы управления (режим скорости и режим момента) для ПЧ могут переключаться напрямую, инструкция может продолжать выполняться. Перед переключением режимов управления убедитесь, что все параметры ПЧ установлены.
5. Таблица режимов:

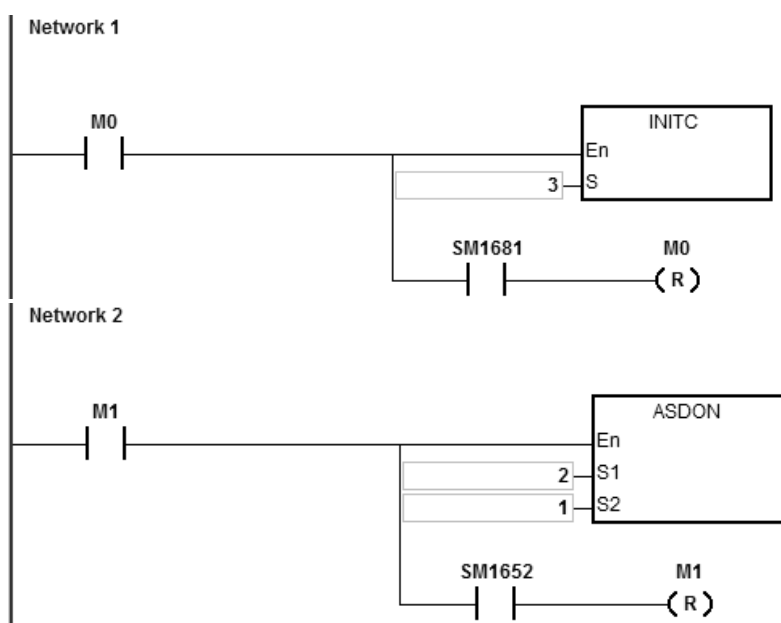
Привод	Диапазон $S_1$	Диапазон $S_2$
Сервопривод	1 – 8	1 = SERVO-ON 0 = SERVO-OFF
Преобразователь частоты	21 – 28	0 = Останов

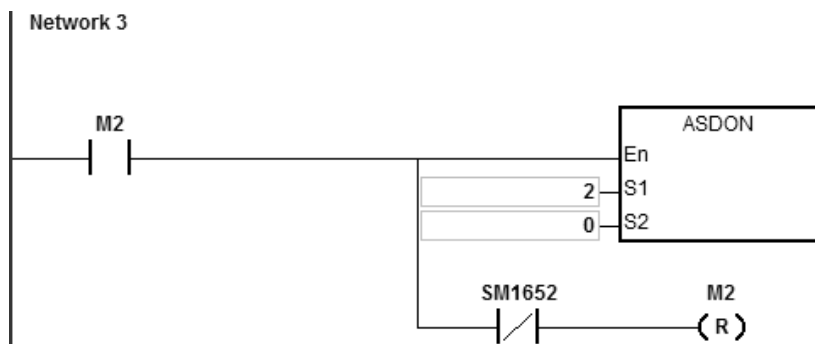
		1 = Режим скорости
		2 = Режим момента

- Каждый сервопривод имеет флаги (SM1651 ~ M1658) для отображения собственного состояния. Фактическое состояние сервопривода можно узнать из состояния флага. Когда флаг включен, сервопривод получает сигнал Servo-ON. Когда флаг выключен, сервопривод отключен.
- Каждый ПЧ (21-28) имеет флаг (SM1621-M1628) для отображения своего состояния. Вы можете прочитать фактическое состояние ПЧ из флага. Когда флаг включен, ПЧ также включен. Когда флаг выключен, ПЧ выключен.
- Если во время коммуникации возникает ошибка, включается SM1682. В регистре SR658 будет сохранен номер сервопривода, в котором возникла ошибка, а в регистре SR659 будет сохранен код ошибки.

**Пример 1 (LD):**

- Когда M0 включен, инструкция INITC начинает инициализацию сервоприводов с адресами 1~3, по завершении инициализации включается SM1681.
- Когда M1 включен, инструкция ASDON начинает активировать сервопривод с адресом станции 2.  
Когда SM1652 включен, сигнал Servo-ON активирован.
- Когда M2 включен, инструкция ASDON начинает отключать сервопривод с адресом станции 2. Когда SM1652 выключен, сигнал Servo-OFF активирован.





**Пример 2 (программирование на языке ST):**

```

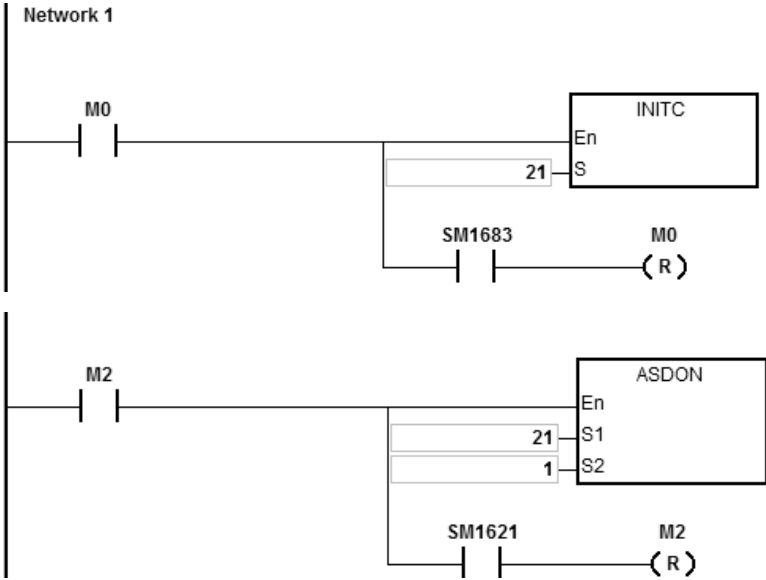
0001 IF M1 THEN
0002   ASDON(2,1);
0003 END_IF;
0004 IF M1 AND SM1652 THEN
0005   SM1582 := TRUE;
0006   M1 := FALSE;
0007 END_IF;
0008
0009 IF M2 THEN
0010   ASDON(2,0);
0011 END_IF;
0012 IF M2 AND (not SM1652) THEN
0013   SM1582 := TRUE;
0014   M2 := FALSE;
0015 END_IF;

```

1. Когда M1 включается, инструкция начинает включать сервопривод с адресом станции 2. Когда SM1652 включен, сигнал Servo-ON активирован.
2. Убедитесь, что M1 и SM1652 включены, а затем включите флаг SM1582 управления прямой связью для адреса станции 2. Это освободит право управления связью, и адрес станции 2 может принимать и выполнять другие инструкции связи.
3. Когда M2 включается, инструкция начинает отключать сервопривод по адресу станции 2. Когда SM1652 выключен, сигнал Servo-OFF активирован.
4. Убедитесь, что M2 включен, а SM1652 выключен, а затем включите флаг SM1582 управления прямой обратной связью для адреса станции 2. Это освободит право управления связью, и адрес 2 станции может принимать и выполнять другие инструкции связи.

**Пример 3 (связь с ПЧ Delta):**

1. Когда M0 включается, инструкция начинает инициализировать адрес станции 21. Когда SM1683 включен, он указывает, что инициализация завершена.
2. Когда M2 выключается, эта инструкция начинает устанавливать режим управления на адрес станции 21. Когда SM1621 включен, это означает, что ПЧ работает.





API	Код инструкции			Операнды								Функция				
2802	CASD			$S_1$ , $S_2$ , $S_3$								Настройка времени разгона и торможения для привода				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$	●	●			●	●		●	●		○	○	○	○		
$S_2$	●	●			●	●		●	●		○	○	○	○		
$S_3$	●	●			●	●		●	●		○	○	○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

Символ:

CASD	
En	
S1	
S2	
S3	

 $S_1$  : Адрес станции привода $S_2$  : Время разгона (мс) $S_3$  : Время торможения (мс)

Описание:

1. Выполнение инструкции INITS должно быть закончено до выполнения данной инструкции.
2. Диапазон  $S_1$  составляет 1-8 (для сервоприводов) и 21-28 (для ПЧ). Когда значение выходит за пределы диапазона, инструкция не будет выполняться.
3. Диапазоны и описание  $S_2$  и  $S_3$ :

Привод	Диапазон $S_1$	Диапазоны $S_2$ и $S_3$	Описание $S_2$ и $S_3$
Сервопривод	1 – 8	1 – 32767 (мс)	$S_2$ : Время ускорения – это период времени, в течение которого сервопривод ускоряется от 0 до 3000,0 об/мин. $S_3$ : Время замедления – это период времени, в течение которого сервопривод замедляется от 3000,0 об/мин до 0.
ПЧ	21 – 28	100 – 32700 (мс) Входное значение следует умножить на 100	$S_2$ : Время ускорения – это период времени, в течение которого ПЧ ускоряется от 0 до максимальной рабочей частоты. $S_3$ : Время замедления – это период времени, в течение которого ПЧ замедляется от максимальной рабочей частоты до 0.

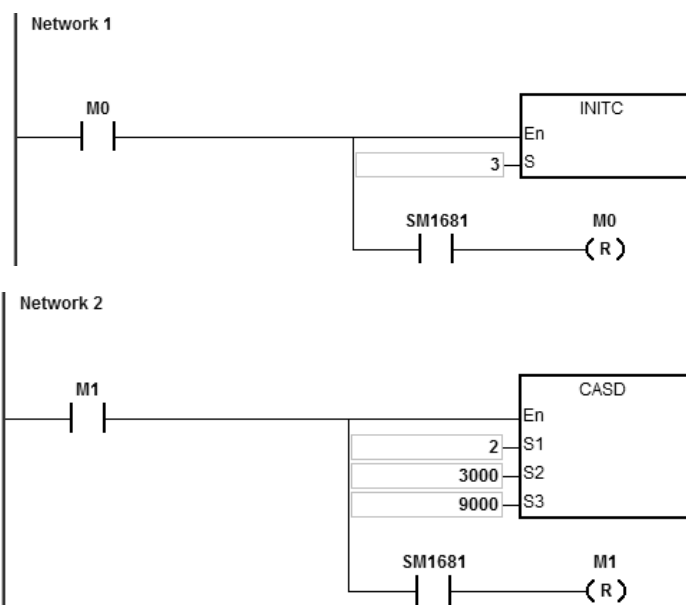
- При выполнении этой инструкции, если диапазон **S<sub>1</sub>** равен 1-8 (для сервоприводов), SM1681 выключается (параметры редактируются). Когда сервопривод отвечает на полученную команду, SM1681 включается (параметры установлены успешно). При выполнении этой инструкции, если диапазон **S<sub>1</sub>** равен 21-28 (для ПЧ), SM1683 выключается (параметры редактируются). Когда ПЧ отвечает на полученную команду, SM1683 включается (параметры установлены успешно).
- Если во время коммуникации возникает ошибка, включается SM1682. В регистре SR658 будет сохранен номер привода, в котором возникла ошибка, а в регистре SR659 будет сохранен код ошибки.

**Пример 1 (коммуникация с сервоприводом):**

- Когда M0 включен, инструкция INITC начинает инициализацию сервоприводов с адресами 1~3, по завершении инициализации включается SM1681.
- Когда включен M1 и заданная скорость сервопривода с адресом 2 равна 3000 об/мин, инструкция CASD задает для сервопривода с адресом станции 2 скорость разгона 3000 мс и время торможения 9000 мс.
- Если заданная скорость сервопривода 2 составляет 1000 об/мин, время разгона и торможения будет следующим:

Время разгона:  $[3000 \text{ мс} / 3000 \text{ об/мин}] * 1000 \text{ об/мин} = 1000 \text{ мс}$

Время торможения:  $[9000 \text{ мс} / 3000 \text{ об/мин}] * 1000 \text{ об/мин} = 3000 \text{ мс}$



**Пример 2 (коммуникация с ПЧ)**

- Когда M0 включен, инструкция начинает инициализировать адрес станции 21. Когда SM1683 включен, он указывает, что инициализация завершена.
- Когда M1 включен, эта инструкция начинает устанавливать время ускорения до 2000 мс и время торможения до 3000 мс на адрес станции 21.

API	Код инструкции			Операнды								Функция			
2803	D	DRVIC		$S_1$ , $S_2$ , $S_3$								Управление относительным позиционированием для привода			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$								●					○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

## Символ:

DDRVIC	
En	
S1	
S2	
S3	

- $S_1$  : Адрес станции привода
- $S_2$  : Относительная заданная позиция
- $S_3$  : Заданная скорость

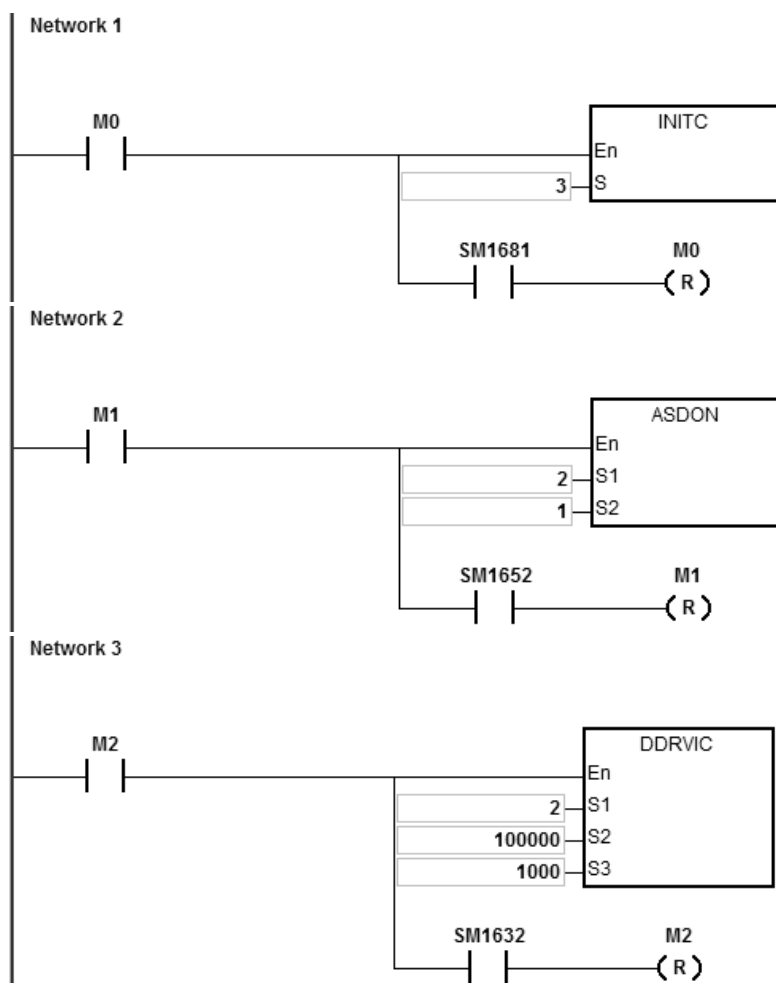
## Описание:

- Перед выполнением данной инструкции необходимо выполнить инструкции INITC и ASDON (Servo-ON).
- Диапазон  $S_1$ : 1~8.
- Диапазон  $S_2$ : -2147483648 ~ 2147483647. Знак +/- определяет прямое/реверсное вращение. Заданная позиция является относительной позицией.
- Единица значения  $S_3$ : 0.1 об/мин. Диапазон: 1~60000, что означает 0.1~6000.0 об/мин.
- При достижении заданной позиции включается соответствующий сервоприводу флаг завершения (SM1631~SM1638).
- См. таблицу SM и SR, соответствующих сервоприводам.
- Если во время коммуникации возникает ошибка, включается SM1682. В регистре SR658 будет сохранен номер сервопривода, в котором возникла ошибка, а в регистре SR659 будет сохранен код ошибки.

## Пример 1 (программирование на LD):

1. Когда M0 включен, инструкция INITC начинает инициализацию сервоприводов с адресами 1~3, по завершении инициализации включается SM1681.
2. Когда M1 включен, инструкция ASDON начинает активировать сервопривод с адресом станции 2. Когда SM1652 включен, сигнал Servo-ON активирован.
3. Когда M2 включен, сервопривод с адресом станции 2 начинает движение для достижения заданной относительной позиции 100000 PУU со скоростью 100.0 об/мин.

После достижения заданной позиции включается SM1632.



**Пример 2 (программирование на ST):**

```

0001 IF M2 THEN
0002     DDRVIC (2, 100000, 1000);
0003 END_IF;
0004 IF M2 AND SM1632 THEN
0005     SM1582 := TRUE;
0006     M2 := FALSE;
0007 END_IF;

```

1. Когда M2 включен, сервопривод 2 перемещается в относительное положение 100000 PUU при 100,0 об/мин. Флаг SM1632 включен, когда достигнуто заданное положение.

2. Убедитесь, что M2 и SM1632 включены, а затем включите флаг SM1582 управления прямой связью для адреса станции 2. Это освободит право управления связью, и адрес 2 станции может принимать и выполнять другие инструкции связи.

### Описание специальных флагов (SM) и регистров (SR), соответствующих инструкциям коммуникации по CAN с сервоприводами Delta серии ASD-A2

Специальные флаги (SM):

Флаг	R/W	ID. 1	ID. 2	ID. 3	ID. 4	ID. 5	ID. 6	ID. 7	ID. 8
Включение определенной функции	R/W	Флаг задается в HWCONFIG и включается, когда ПЛК запускается							
Инициализация и коммуникация выполнены	R	SM1681							
Флаг ошибки связи CANopen	R	SM1682							
Позиционирование завершено	R	SM1631	SM1632	SM1633	SM1634	SM1635	SM1636	SM1637	SM1638
Флаг останова	R/W	SM1641	SM1642	SM1643	SM1644	SM1645	SM1646	SM1647	SM1648
Флаг Servo-ON	R	SM1651	SM1652	SM1653	SM1654	SM1655	SM1656	SM1657	SM1658
Функция перехода вперед/назад включена	R/W	SM1661	SM1662	SM1663	SM1664	SM1665	SM1666	SM1667	SM1668
Поддерживается только DDRVAC	R	SM1671	SM1672	SM1673	SM1674	SM1675	SM1676	SM1677	SM1678
Флаг индикации направления вперед/назад	R	SR658							
Поддерживается только DDRVAC	R	SR659							

Специальные регистры (SR):

Номер и наименование параметра сервопривода		ID. 1	ID. 2	ID. 3	ID. 4	ID. 5	ID. 6	ID. 7	ID. 8
Команда PR (P5_07)	→	SR661	SR662	SR663	SR664	SR665	SR666	SR667	SR668
Код тревоги (P0_01) (шестнадцатеричный)	→	SR671	SR672	SR673	SR674	SR675	SR676	SR677	SR678
Состояние дискретного выхода (P0_46)	→	SR681	SR682	SR683	SR684	SR685	SR686	SR687	SR688
Текущая позиция сервопривода (P0_09)	→	SR691	SR693	SR695	SR697	SR699	SR701	SR703	SR705
		SR692	SR694	SR696	SR698	SR700	SR702	SR704	SR706
Команда задания позиции (P0-10)	→	SR711	SR713	SR715	SR717	SR719	SR721	SR723	SR725
		SR712	SR714	SR716	SR718	SR720	SR722	SR724	SR726

Коды ошибок:

Код ошибки	Причина возникновения ошибки
0x0002	Slave устройство не отвечает на сообщение SDO
0x0003	Ошибка в сообщении, которое получает Slave устройство. Ошибка часто возникает из-за настройки инструкции COPRW, когда Slave устройство не может получить полное сообщение
0x0004	Не принимается сообщение от Slave устройства PDO
0x0005	Ошибка возникает при использовании операнда инструкции
0x0006	При выполнении команды INITC один из сервоприводов занят

### Описание специальных флагов (SM) и регистров (SR), соответствующих инструкциям коммуникации по CAN с ПЧ Delta

Специальные флаги (SM):

Флаг	R/W	ID. 21	ID. 22	ID. 23	ID. 24	ID. 25	ID. 26	ID. 27	ID. 28
Флаг завершения инициализации	R	SM1683							
Флаг запуска ПЧ	R	SM1621	SM1622	SM1623	SM1624	SM1625	SM1626	SM1627	SM1628
Ошибка тактирования ПЧ	R	SM1611	SM1612	SM1613	SM1614	SM1615	SM1616	SM1617	SM1618
Автоматическое управление передачей сообщений	R/W	SM1601	SM1602	SM1603	SM1604	SM1605	SM1606	SM1607	SM1608

Специальные регистры (SR):

Номер и наименование параметра сервопривода		ID. 1	ID. 2	ID. 3	ID. 4	ID. 5	ID. 6	ID. 7	ID. 8
Текущее состояние ПЧ	→	SR751	SR752	SR753	SR754	SR755	SR756	SR757	SR758
Текущая скорость ПЧ (ед.: об/мин)	→	SR761	SR762	SR763	SR764	SR765	SR766	SR767	SR768
Текущий момент (ед.: 0,1%)	→	SR771	SR772	SR773	SR774	SR775	SR776	SR777	SR778
Состояние дискретных входов	→	SR781	SR782	SR783	SR784	SR785	SR786	SR787	SR788

Коды ошибок:

Код ошибки	Причина возникновения ошибки
0x0002	Slave устройство не отвечает на сообщение SDO
0x0003	Ошибка в сообщении, которое получает Slave устройство. Ошибка часто возникает из-за настройки инструкции COPRW, когда Slave устройство не может получить полное сообщение

---

0x0004	Не принимается сообщение от Slave устройства PDO
0x0005	Ошибка возникает при использовании операнда инструкции
0x0006	При выполнении команды INITS один из сервоприводов занят
0x0007	Адрес Slave станции превышает допустимый диапазон инструкции INITS
0x0008	Slave устройство не может быть сброшено; проверьте, подключен ли коммуникационный кабель, и работает ли связь с Slave устройством должным образом

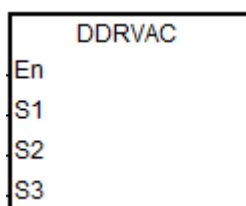
API	Код инструкции			Операнды								Функция				
2804	D	DRVAC		$S_1, S_2, S_3$								Управление абсолютным позиционированием для сервопривода				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$								●					○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						
$S_3$			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

**Символ:**



- $S_1$  : Адрес станции сервопривода
- $S_2$  : Абсолютная заданная позиция
- $S_3$  : Заданная скорость

**Описание:**

1. Перед выполнением данной инструкции необходимо выполнить инструкции INITC и ASDON (Servo-ON).
2. Диапазон  $S_1$ : 1~8.
3. Диапазон  $S_2$ : -2147483648 ~ 2147483647. Знак +/- определяет прямое/реверсное вращение. Заданная позиция является абсолютной позицией.
4. Другие сведения и пример см. описание инструкции DRVIC.

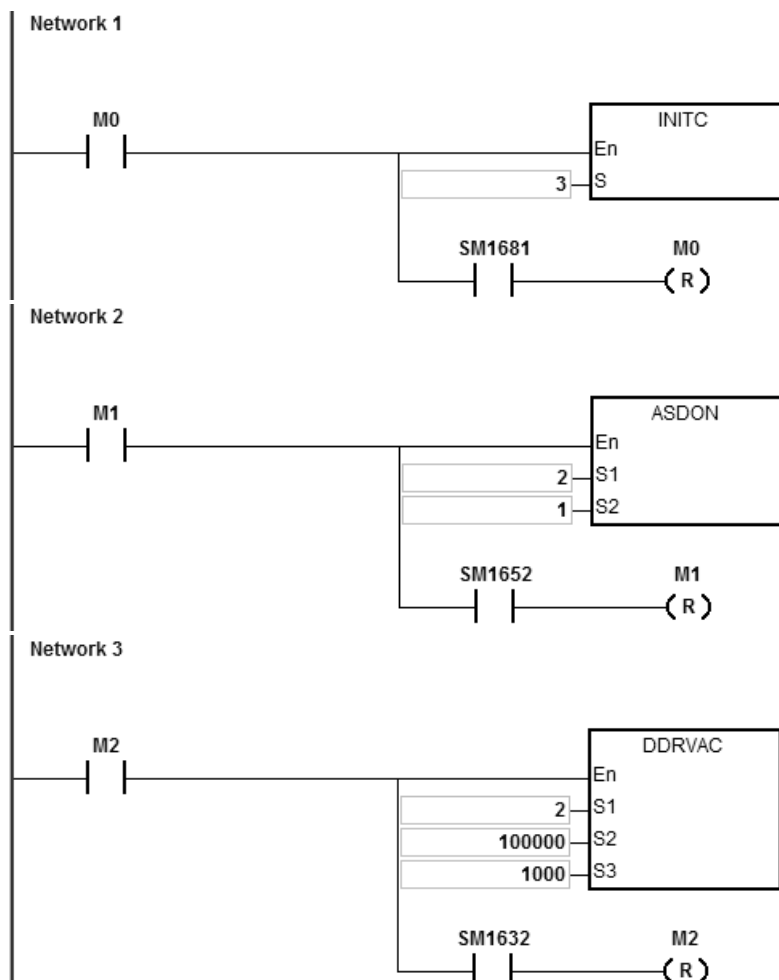
**Пример 1:**

1. Когда M0 включен, инструкция INITC начинает инициализацию сервоприводов с адресами 1~3, по завершении инициализации включается SM1681.
2. Когда M1 включен, инструкция ASDON начинает активировать сервопривод с адресом станции 2. Когда SM1652 включен, сигнал Servo-ON активирован.



3. Когда M2 включен, сервопривод с адресом станции 2 начинает движение для достижения заданной абсолютной позиции 100000 PУU со скоростью 100.0 об/мин.

После достижения заданной позиции включается SM1632

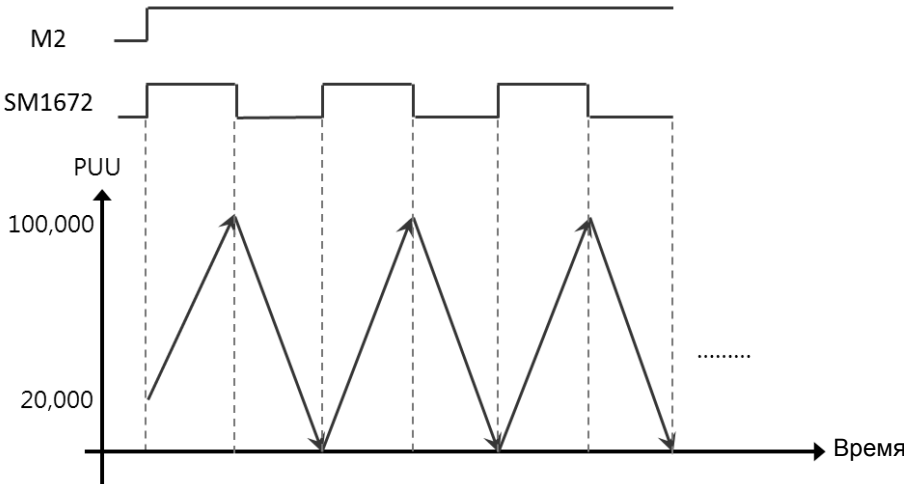


Пример 2:

1. Добавьте одну строку программы в программу в примере 1. Когда ПЛК запускается и SM1662 включен, функция направления вперед /назад для сервопривода 2 включена.



2. Как показано на рисунке ниже, после включения M2 сервопривод перемещается из текущего положения 20 000 в заданное абсолютное положение 100 000, а затем он перемещается от абсолютного положения 100 000 к 0.
3. Флаг индикации направления SM1672 включен, когда сервопривод переходит в исходное положение в первый раз после выполнения Servo-ON. После этого флаг будет выключаться.
4. Заданная позиция может быть изменена в любое время в процессе движения. Но вступление в силу нового задания позиции возможно только со следующего цикла после возврата сервопривода в исходную позицию.



API	Код инструкции			Операнды								Функция				
2805	D	PLSVC		$S_1$ , $S_2$								Управление скоростью для привода				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
$S_1$								●					○	○		
$S_2$								●	●				○	○		

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$			●				●						
$S_2$			●				●						

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	—	AS

## Символ:

DPLSVC	
En	
S1	
S2	

- $S_1$  : Адрес станции сервопривода
- $S_2$  : Заданная скорость

## Описание:

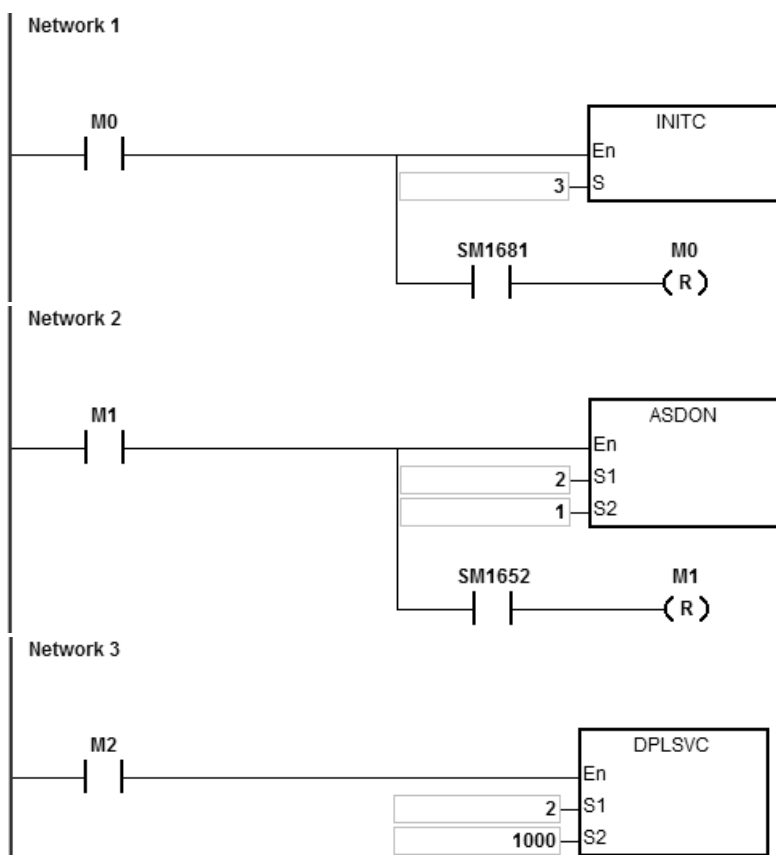
- Перед выполнением данной инструкции необходимо выполнить инструкции INITC и ASDON (Servo-ON).
- Диапазон  $S_1$  составляет 1-8 (для сервоприводов) и 21-28 (для ПЧ). Когда входное значение выходит за пределы диапазона, инструкция не будет выполняться.
- Диапазон  $S_2$ :

Привод	Диапазон $S_1$	Диапазон $S_2$	Описание $S_2$
Сервопривод	1 – 8	-60000 – 60000 (0,1 об/мин)	Знак +/- указывает / обратное направление прямое. Например, при вводе 6005 в $S_2$ значение составляет 600,5 об/мин
ПЧ	21 – 28	-32768 - 32767 (1 об/мин)	Знак +/- указывает / обратное направление прямое. Например, при вводе 605 в $S_2$ значение составляет 605 об/мин

- См. описание инструкции DRVIC (API 2803) для сведений по SM и SR для приводов.
- Если во время коммуникации возникает ошибка, включается SM1682. В регистре SR658 будет сохранен номер привода, в котором возникла ошибка, а в регистре SR659 будет сохранен код ошибки.

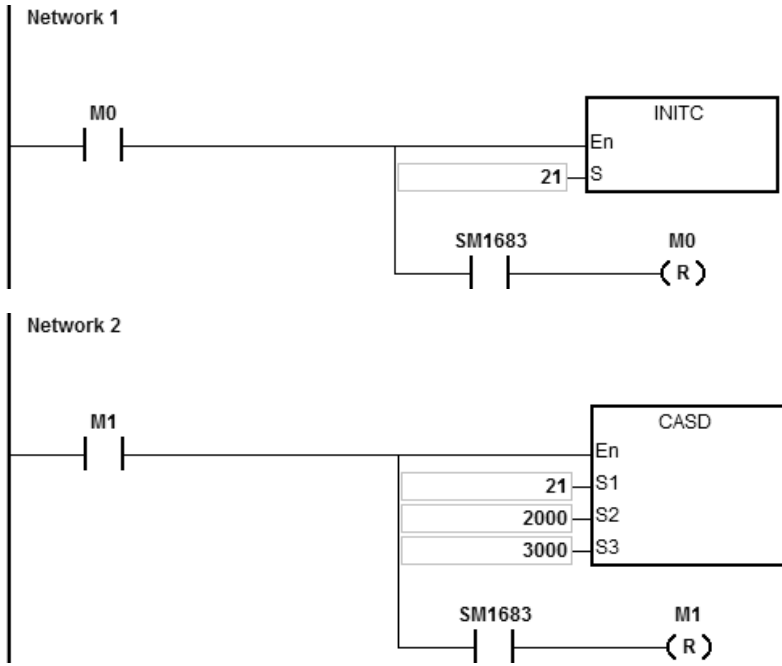
**Пример 1 (коммуникация с сервоприводом Delta):**

1. Когда M0 включен, инструкция INITC начинает инициализацию сервоприводов с адресами 1~3, по завершении инициализации включается SM1681.
2. Когда M1 включен, инструкция ASDON начинает активировать сервопривод с адресом станции 2. Когда SM1652 включен, сигнал Servo-ON активирован.
3. Когда M2 включен, сервопривод с адресом станции 2 двигается со скоростью 100,0 об/мин, пока M2 не будет выключен.



**Пример 2 (коммуникация с ПЧ Delta):**

1. Когда M0 включен, инструкция начинает инициализировать адрес станции 21. Когда SM1683 включен, он указывает, что инициализация завершена.
2. Когда M1 включен, инструкция начинает устанавливать время разгона и торможения для адреса станции 21.
3. Когда M2 включен, инструкция начинает устанавливать режим управления для адреса станции 21.
4. Когда M3 включен, ПЧ работает на скорости 2500 об/мин.



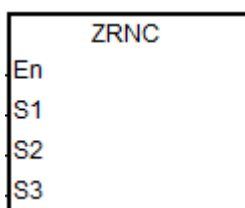
API	Код инструкции			Операнды								Функция			
2806		ZRNC		$S_1, S_2, S_3$								Возврат в начальную позицию (Homing)			

Device	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
$S_1$								●					○	○		
$S_2$								●	●				○	○		
$S_3$								●	●				○	○		

Data type	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
$S_1$		●			●	●							
$S_2$		●			●	●							
$S_3$		●			●	●							

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

**Символ:**



- $S_1$  : Адрес станции сервопривода
- $S_2$  : 1<sup>й</sup> – сегмент скорости
- $S_3$  : 2<sup>й</sup> – сегмент скорости

**Описание:**

1. Перед выполнением данной инструкции необходимо выполнить инструкции INITC и ASDON (Servo-ON).
2. Диапазон  $S_1$ : 1~8.
3. Диапазон  $S_2$ : 1 ~ 20000. Шаг: 0,1 об/мин
4. Диапазон  $S_3$ : 1 ~ 5000. Шаг: 0,1 об/мин
5. Когда сервопривод вернется в исходную точку, будет включен соответствующий сервоприводу флаг завершения. См. описание инструкции DRVIC для информации о специальных флагах (SM) и регистрах (SR).
6. Если во время коммуникации возникает ошибка, включается SM1682. В регистре SR658 будет сохранен номер сервопривода, в котором возникла ошибка, а в регистре SR659 будет сохранен код ошибки.

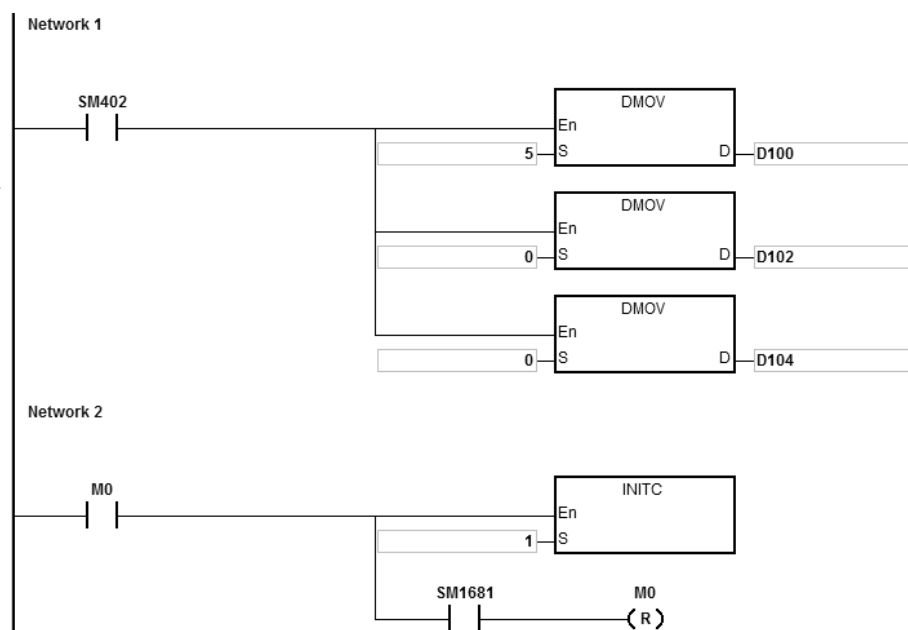
**Пример:**

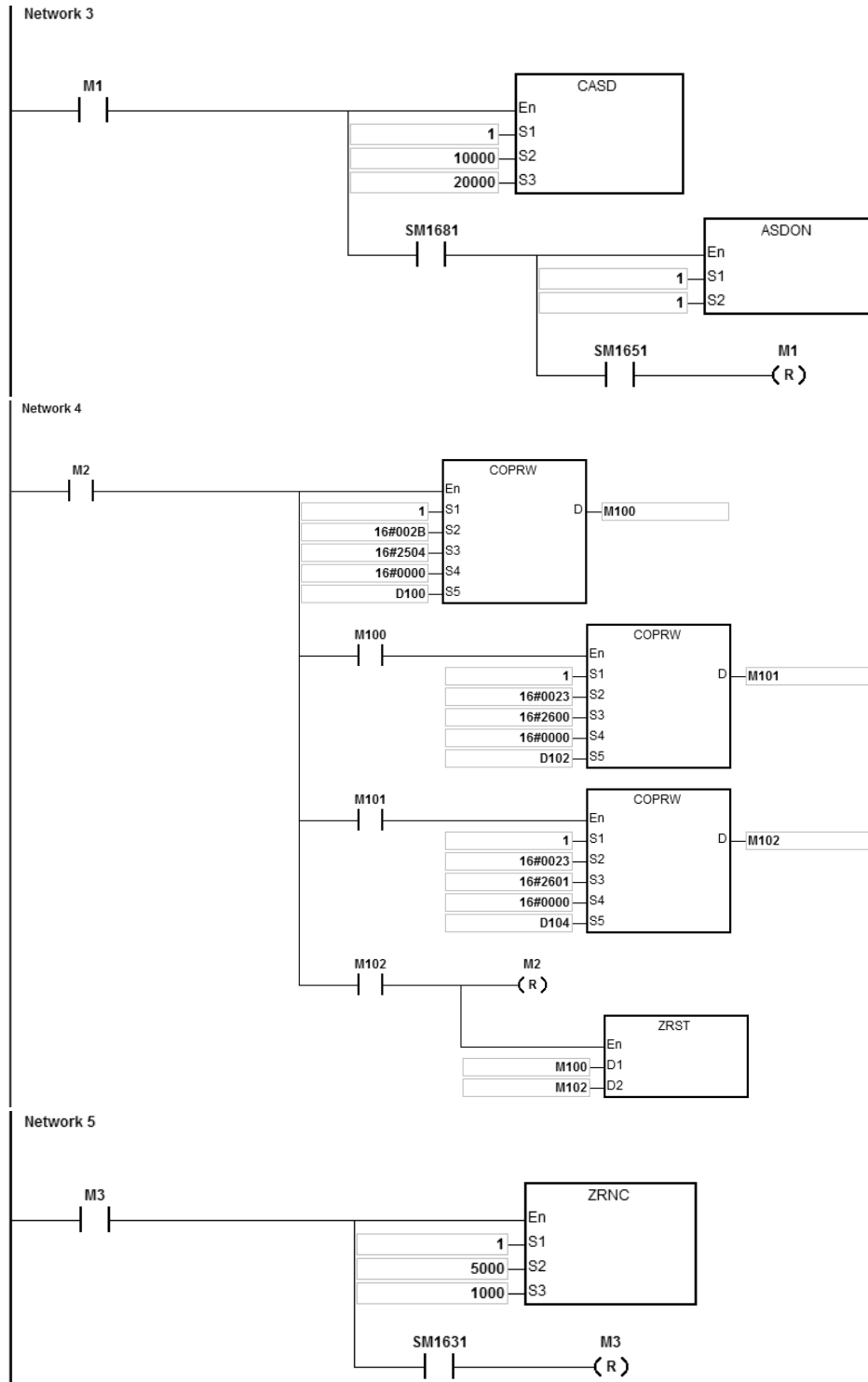
1. Когда M0 включен, инструкция INITC начинает инициализацию сервопривода с адресом 1, по завершении инициализации включается SM1681.

2. Когда M1 включен, инструкция CASD задает время разгона (10000) и время замедления (20000) для сервопривода с адресом станции 1. Когда SM1652 включен, сигнал Servo-ON активирован.
3. Когда M2 включен,
  - 2-байтовые данные записываются в P5-04 сервопривода 1. Когда D100 = 5 и запись данных закончена, включается M100.
  - 4-байтовые данные записываются в P6-00 сервопривода 1. Когда D102 = 0 и запись данных закончена, включается M101.
  - 4-байтовые данные записываются в P6-01 сервопривода 1. Когда D104 = 0 и запись данных завершена, включается M102.

P5-04 используется для настройки режима возврата в начальную позицию (Homing-режим).

P6-00 и P6-01 – параметры для описания режима возврата в начальную позицию (Homing-режим).
4. Когда M3 включен, функция возврата в начальную позицию для сервопривода 1 активирована.







API	Код инструкции			Операнды								Функция					
2807		COPRW			<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, S<sub>4</sub>, S<sub>5</sub>, D</b>								Запись и чтение данных коммуникации CANopen				

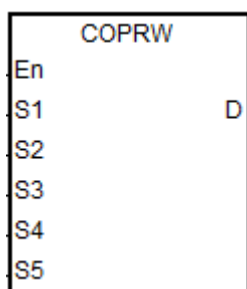
Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
S <sub>1</sub>								●					○	○		
S <sub>2</sub>								●	●				○	○		
S <sub>3</sub>								●	●				○	○		
S <sub>4</sub>								●	●				○	○		
S <sub>5</sub>								●								
D		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
S <sub>1</sub>		●			●	●							
S <sub>2</sub>		●			●	●							
S <sub>3</sub>		●			●	●							
S <sub>4</sub>		●			●	●							
S <sub>5</sub>		●			●	●							
D	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

6

Символ:



- S<sub>1</sub> : Адрес станции сервопривода
- S<sub>2</sub> : Код запроса
- S<sub>3</sub> : Индекс
- S<sub>4</sub> : Субиндекс
- S<sub>5</sub> : Объект для чтения / записи
- D : Флаг завершения связи

Описание:

1. Диапазон S<sub>1</sub>: 1~127. Если значение выходит за допустимые пределы (<1 или >127), в S<sub>1</sub> автоматически задается минимальное или максимальное возможное значение.
2. S<sub>2</sub> может реализовывать только четыре типа кодов запроса, как показано в таблице ниже.

H23	Запись 4-байтных данных
H2B	Запись 2-байтных данных

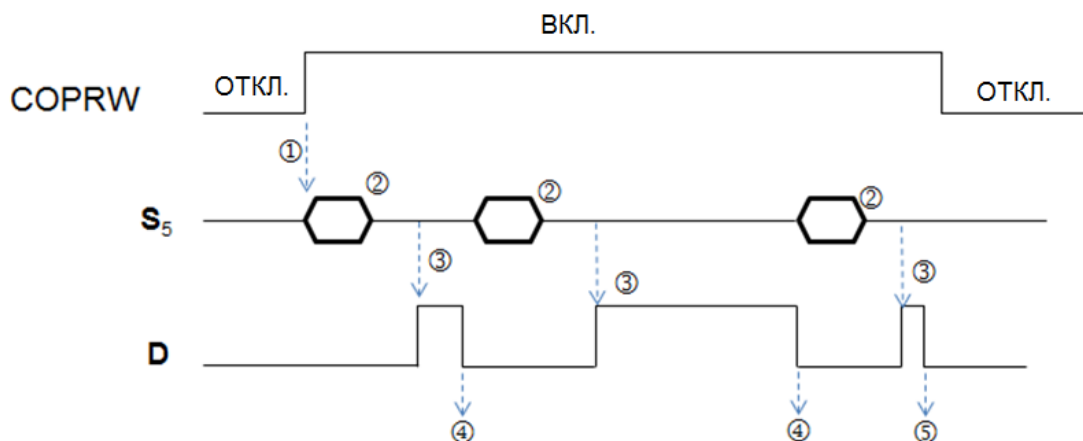
H2F	Запись 1-байтных данных
H40	Чтение данных. Длина данных содержится в сообщении ответа SDO

3. Описание **S<sub>3</sub>** и **S<sub>4</sub>**, см. в словаре объектов в Руководстве по эксплуатации сервоприводов.
4. В случае если записанные значения параметров могут быть перезаписаны инструкцией INITC, инструкция COPRW должна быть выполнена после завершения выполнения инструкции INITC. Инструкция COPRW используется для связи формата SDO CANopen. Если адрес станции Slave устройства выходит за пределы диапазона 1 ~ 8, то инструкцию COPRW можно использовать для настройки, чтения и записи параметров, связанных с Slave устройством.
5. Если во время коммуникации возникает ошибка, включается SM1682. В регистре SR658 будет сохранен номер сервопривода, в котором возникла ошибка, а в регистре SR659 будет сохранен код ошибки.

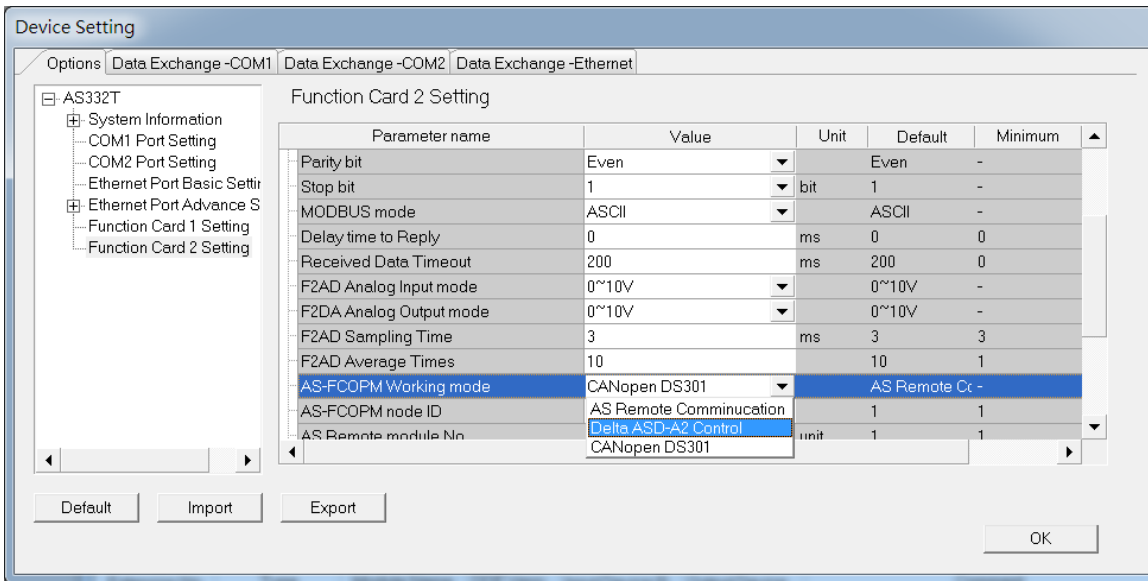
**Примечание:** Когда используется инструкция COPRW, необходимо отредактировать процесс обработки ошибок связи, чтобы избежать неправильного процесса связи, вызванного коммуникационными ошибками.

6. Временная диаграмма инструкции COPRW.

- ① Когда инструкция COPRW выполняется впервые, она немедленно отправит командный код, если его не использует другое сообщение CANopen.
- ② Инструкция отправляет командный код.
- ③ После выполнения передачи включается флаг завершения.
- ④ Далее можно подготовить следующие данные для отправки. Следующий командный код отправляется сразу после того, как выключается флаг завершения.
- ⑤ Передача завершена, и инструкция COPRW отключается.



7. Инструкция COPRW поддерживает только следующий режим управления сервоприводом Delta ASD-A2.  
Настройки аппаратной конфигурации:



Большинство параметров в ASDA-A2 отображаются в десятичном формате. Можно преобразовать параметры в индексные адреса, используя формулу: индексный адрес PX-YY = 0x2000 + (X << 8) + YY

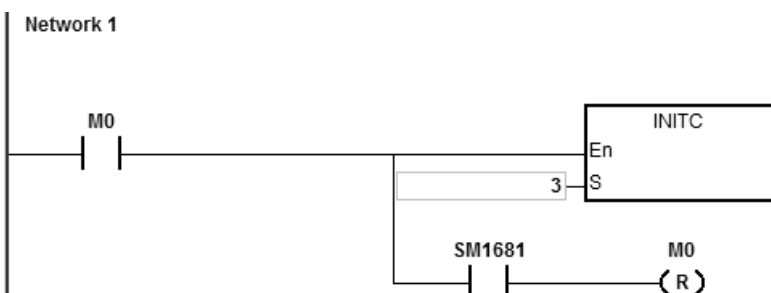
Пример: P2-10 = 0x2000 + (0x0002 << 8) + 0x000A = 0x220A

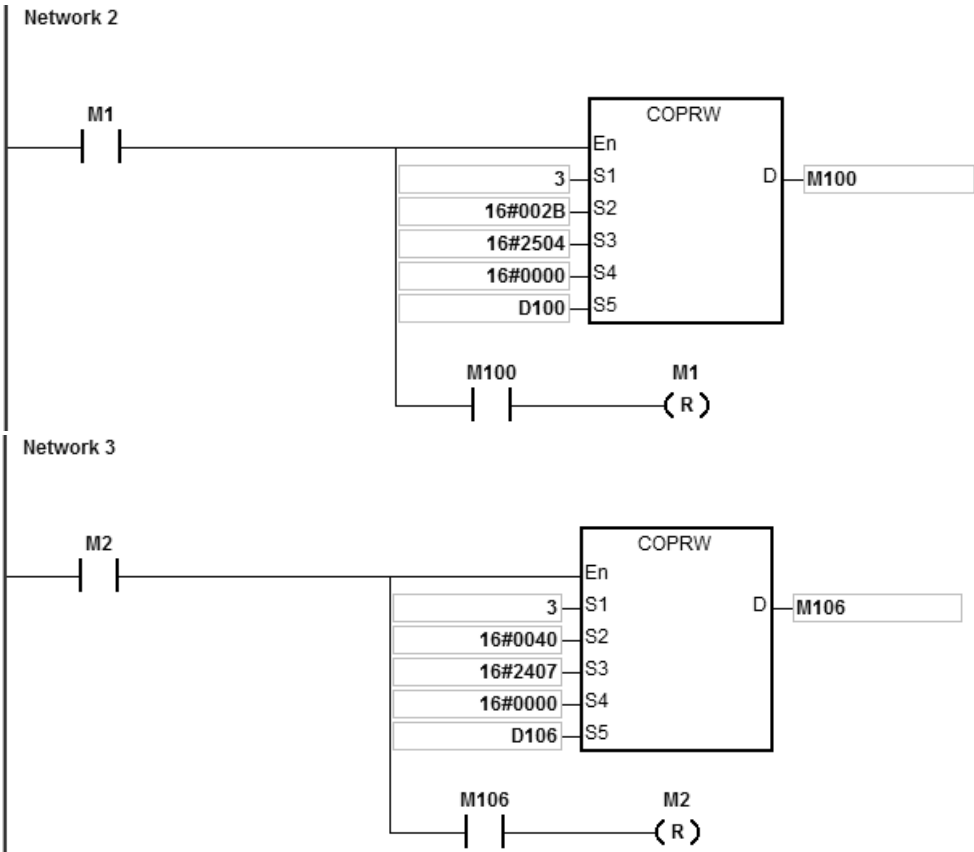
P5-04 = 0x2000 + (0x0005 << 8) + 0x0004 = 0x2504

P1-44 = 0x2000 + (0x0001 << 8) + 0x002C = 0x212C

#### Пример:

1. Когда M0 включен, инструкция INITC начинает инициализацию сервоприводов с адресами 1~3, по завершении инициализации включается SM1681.
2. Когда M1 включен, ПЛК с помощью инструкции COPRW записывает 2-байтные данные в D100 параметра P5-04 сервопривода с адресом 3. Когда запись выполнена, включается M100.
3. Когда M2 включен, ПЛК считывает значение параметра P4-07 сервопривода с адресом 3 и сохраняет эти данные в D106. Когда считывание завершено, включается M106.





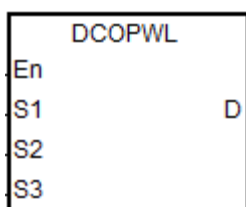
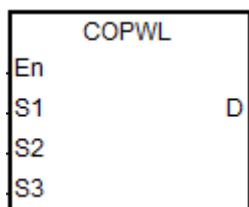
API	Код инструкции			Операнды								Функция				
2808	D	COPWL		<b>S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>, D</b>								Запись нескольких значений параметров по CANopen				

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	"\$"	F
<b>S<sub>1</sub></b>								●					○	○		
<b>S<sub>2</sub></b>								●	●							
<b>S<sub>3</sub></b>								●					○	○		
<b>D</b>		●	●	●				●								

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
<b>S<sub>1</sub></b>		●			●	●							
<b>S<sub>2</sub></b>		●	●		●	●	●						
<b>S<sub>3</sub></b>		●			●	●							
<b>D</b>	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	AS

**Символ:**



- S<sub>1</sub>** : Адрес станции (Mac ID)
- S<sub>2</sub>** : Стартовый операнд, в котором сохраняются записанные данные
- S<sub>3</sub>** : Количество сообщений при последовательной записи данных
- D** : Флаг завершения связи

**Описание:**

- S<sub>1</sub>** устанавливает адрес станции в диапазоне от 1 до 127. Если значение настройки превышает диапазон (<1 или > 127), инструкция автоматически применяет минимальное или максимальное значение соответственно.
- S<sub>2</sub>** – это стартовый операнд, в котором сохраняются записанные данные, а **S<sub>3</sub>** – количество сообщений при последовательной записи данных. Например, **S<sub>2</sub>** задает D10 в качестве стартового операнда, а количество сообщений при последовательной записи данных равно 3. См. таблицу ниже:

Инструкция	№ сообщения	Адрес индекса	Адрес субиндекса	Записанные исходные данные
COPWL (Запись 16-битных значений)	1	D10	D11	D12
	2	D13	D14	D15
	3	D16	D17	D18
DCOPWL (Запись 32-битных значений)	1	D10	D11	D12, D13
	2	D14	D15	D16, D17
	3	D18	D19	D20, D21

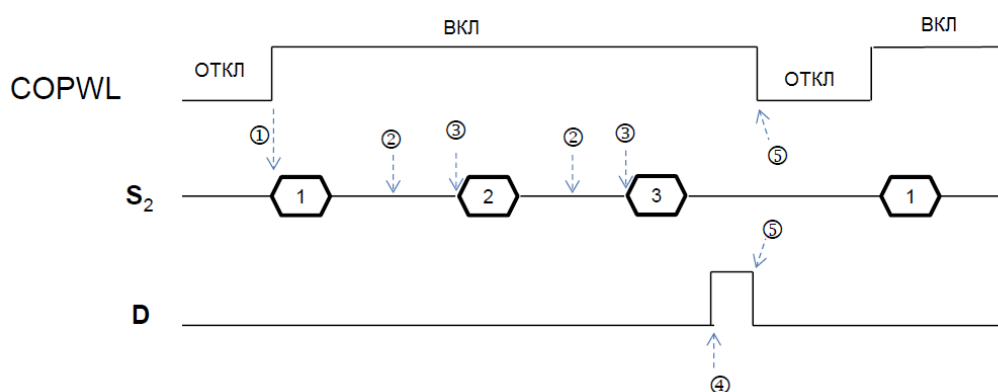
Значение **S<sub>3</sub>** находится в диапазоне 1~100.

- Для адресов индексов и субиндексов сервоприводов и ПЧ Delta см. описание инструкции COPRW. В принципе, значения параметров сервоприводов и ПЧ Delta являются как 16-битными, так и 32-битными значениями, включая числа с плавающей запятой. Если необходимо записать 8-битное значение, также используйте инструкцию COPRW.
- D** – флаг завершения связи. **D** включится после завершения отправки нескольких сообщений.

Диаграмма последовательности процесса показана ниже:

- Инструкция COPWL включена и начинает отправлять данные.
- После того, как инструкция COPWL отправляет одну часть сообщения, продолжает выполняться следующая инструкция ПЛК.
- Когда инструкция COPWL сканируется еще раз, и предыдущее сообщение было получено Slave устройством, инструкция COPWL отправляет следующее сообщение.
- Когда будет выполнена последняя передача записанных данных, инструкция включит флаг завершения.
- Когда флаг завершения включается, инструкцию COPWL необходимо отключить вручную, чтобы смогла выполняться следующая инструкция COPWL или COPRW.

Примечание. При принудительном отключении инструкции, флаг завершения будет автоматически сброшен.



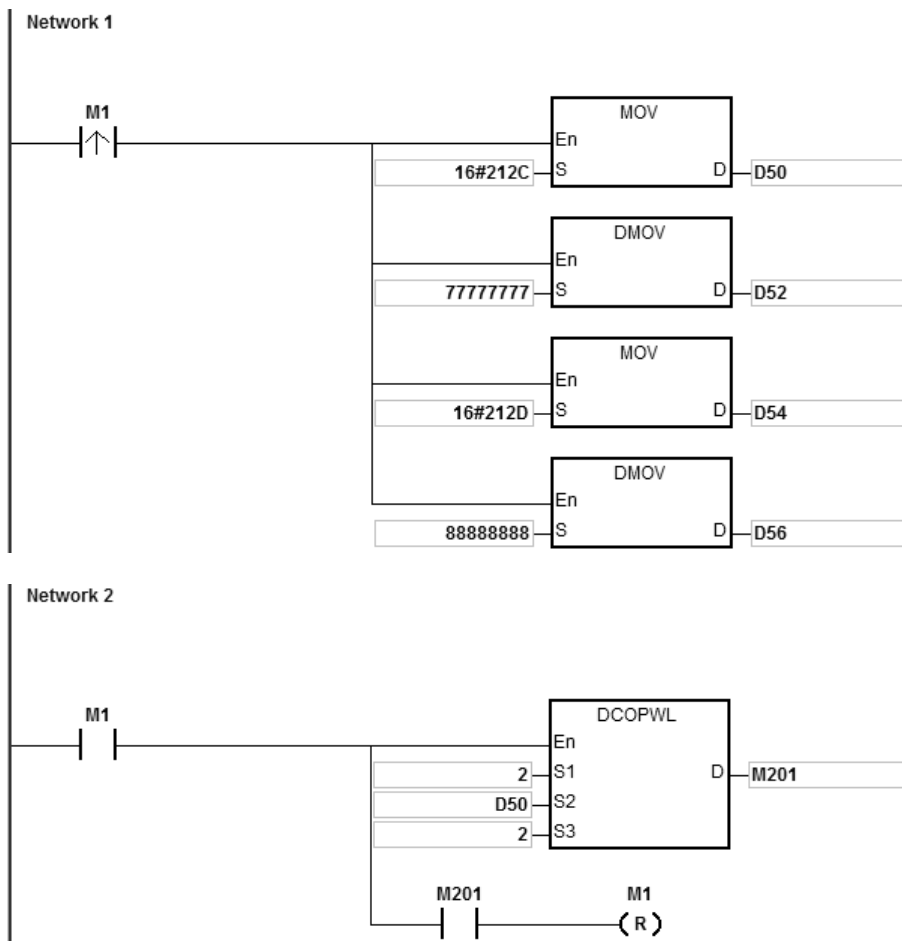
- После того, как инструкция включена, подождите, пока запись не будет завершена, а затем отключите инструкцию. Если при выполнении произошла ошибка связи, устраните проблему, а затем снова запустите инструкцию для записи всех данных.

**Пример:**

- Когда M1 включен, данные записываются в операнд **D**.

Инструкция	№ сообщения	Адрес индекса	Адрес субиндекса	Записанные исходные данные
DCOPWL (Запись 32-битных значений)	1	D50 = 16#212C (числитель передаточного отношения)	D51 = 0	D52, D53 = 77777777
	2	D54 = 16#212D (знаменатель передаточного отношения)	D55 = 0	D56, D57 = 88888888

2. Когда M1 включен, инструкция записывает 32-битное значение для параметра P1-44 сервопривода, чей адрес станции равен 2, а записанное значение 77777777 сохраняется в D52. Инструкция записывает 32-битное значение для параметра P1-45, а записанное значение 88888888 сохраняется в D56. По завершении записи включается M201.



6

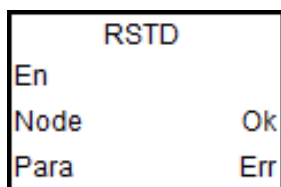
API	Код инструкции			Операнды								Функция			
2809		RSTD		Node, Para, Ok, Err								Отправка команды Сброса или NMT			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Node								●					○	○		
Para								●					○	○		
Ok		●	●	●												
Err		●	●	●												

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Node		●											
Para		●											
Ok	●												
Err	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

**Символ:**



- Node** : Адрес сбрасываемой станции
- Para** : Заданное значение параметра
- Ok** : Флаг выполнения сброса
- Err** : Флаг ошибки сброса

**Описание:**

1. Прежде чем команда RSTD будет использоваться в режиме специальной инструкции Delta, убедитесь, что все приводы Delta были инициализированы с помощью инструкции INITC, и они работают нормально.
2. При использовании в режиме CANopen DS301 инструкция RSTD работает как функция связи NMT и может переключать сетевые состояния через параметр Para.
3. Инструкцию поддерживают только прошивка ПЛК версии V1.04.10 и выше и программное обеспечение ISPSOft версии V3.03.08 и выше.
4. Когда порт связи CAN указан для работы в режиме специального драйвера Delta, значение Node может быть равно 0 (широковещательная функция) и 1 ~ 8, которые предназначены только для адресов сервоприводов и 21 ~ 28 для адресов ПЧ. Когда адрес станции выходит за пределы диапазона, ПЛК не будет выполнять сброс, и включается флаг Err (см. описание SR659).
5. Когда порт связи CAN указан для работы в режиме CANopen DS301, значение Node находится в диапазоне 1 ~ 64 или 0 (широковещательная функция). Когда значение выходит за пределы диапазона, ПЛК не будет



выполнять сброс, и включается флаг **Err** (см. описание SR659).

6. Заданное значение **Para** применимо только к режиму CANopen DS301. Параметр не имеет смысла, если инструкция используется в режиме специального драйвера Delta и в режиме удаленного доступа AS. Настройки **Para** (код службы NMT) приведены в следующей таблице. Если значение настройки не является одним из значений в таблице, включается флаг **Err**.

Служебный код NMT	16#01	16#02	16#80	16#81	16#81
Функция	Пуск Slave устройства	Останов Slave устройства	Ввод предварительного состояния	Сброс прикладных приложений	Сброс коммуникации

7. Инструкция RSTD может одновременно выполнять командное действие только на одном приводе или Slave устройстве. Если одновременно включены несколько инструкций RSTD, ПЛК будет автоматически выполнять инструкцию, которая запущена раньше остальных.
8. Инструкция RSTD выполняется для отправки команды только когда она включена. Если инструкция отключается до включения флага **Ok**, ПЛК не включит флаг **Ok**.
9. Помимо уведомления указанного привода, чтобы сбросить ошибку, инструкция также проверит правильность параметров связи и, при необходимости, повторно настроит правильные значения параметров связи. Например, из-за отключения Slave устройства с адресом станции 2 вся система перестает работать. После того, как проблема решена, Slave устройство с адресом станции 2 может вернуться в управляемое состояние с помощью инструкции RSTD для сброса только Slave устройства с адресом станции 2. Таким образом, нет необходимости в повторной инициализации остальных приводов.
10. Если Slave устройство отправит ПЛК ответ с сообщением об ошибке, инструкция RSTD включит флаг **Err** и остановит свою работу. (см. описание SR659).

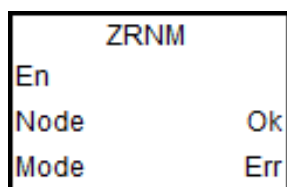
API	Код инструкции			Операнды								Функция					
2810		ZRNM		<b>Node, Mode, Ok, Err</b>								Настройка режима возврата в начальную позицию для сервопривода Delta					

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Node								●					○	○		
Mode								●					○	○		
Ok		●	●	●												
Err		●	●	●												

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Node		●											
Mode		●											
Ok	●												
Err	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

**Символ:**



**Node** : Указанный идентификатор узла (ID)

**Mode** : Код режима возврата в нулевую точку

**Ok** : Флаг выполнения

**Err** : Флаг ошибки

**Описание:**

1. Перед выполнением инструкции убедитесь, что все приводы Delta были инициализированы.
2. Инструкцию поддерживают только прошивка ПЛК версии V1.04.10 и выше и программное обеспечение ISPSOft версии V3.03.08 и выше.
3. Значение узла находится в диапазоне от 1 до 8 (исключая только идентификаторы узлов сервоприводов). Если значение настройки выходит за пределы диапазона, ПЛК не будет выполнять возврат в нулевую точку и включит флаг **Err** в положение ON. (см. описание SR659).
4. Инструкция ZRNM может одновременно выполнять возврат в нулевую точку только на одном приводе. Если одновременно включены несколько инструкций ZRNM, ПЛК будет автоматически выполнять инструкцию, которая запущена раньше остальных.
5. Инструкция ZRNM выполняется для отправки команды только когда она включена. Если инструкция отключается до включения флага **Ok**, ПЛК не включит флаг **Ok**.
6. Инструкция устанавливает режим возврата в нулевую точку. Если значение настройки превышает диапазон,

ПЛК все равно отправит команду, и сервер самостоятельно примет решение о получении команды. Этот режим соответствует параметру сервопривода ASDA P5-04.

Значение настройки режима возврата в нулевую точку для сервоприводов Delta – это шестнадцатеричное значение. Значение определяется как формат 0xWZYX. В таблице ниже показаны соответствующие коды:

Код режима возврата в нулевую точку	Диапазон	Функция и описание кода
W	0 ~ 1	Выбор конечной позиции останова сервопривода: 0 = Сервопривод покидает нулевую точку, замедляется и останавливается, а затем автоматически возвращается к реальной нулевой точке. 1 = После выхода из нулевой точки, замедления и остановки, сервопривод больше не выполняет никаких действий.
Z	0 ~ 1	Действия при достижении предела: 0 = Вывод останавливается. 1 = Вывод осуществляется в обратном направлении
Y	0 ~ 2	Z-импульс (используется для кода X: 0 ~ 8) 0 = Поиск Z-импульса при возврате в нулевую точку. Нет поиска Z-импульса при движении вперед 1 = Поиск Z-импульса при движении вперед. Нет поиска Z-импульса при возврате в нулевую точку 2 = Нет поиска Z-импульса Z-импульс (используется для кода X: 9 ~ A) 0 = Поиск Z-импульса при возврате в нулевую точку 1 = Нет поиска Z-импульса при возврате в нулевую точку и движении вперед
X	0 ~ A	Метод возврата в нулевую точку: 0 ~ 8 0 = Возврат в прямом направлении; <b>PL</b> является нулевой точкой 1 = Возврат в обратном направлении; <b>NL</b> является нулевой точкой 2 = Возврат в прямом направлении; <b>ORGP</b> : OFF > ON, как нулевая точка 3 = Возврат в обратном направлении; <b>ORGP</b> : OFF > ON, как нулевая точка 4 = Возврат в прямом направлении; поиск Z-импульса и признание его нулевой точкой 5 = Возврат в обратном направлении; ищите Z-импульс и считайте его нулевой точкой 6 = Возврат в прямом направлении; <b>ORGP</b> : ON > OFF, как нулевая точка 7 = Возврат в обратном направлении; <b>ORGP</b> : ON > OFF, как нулевая точка 8 = Текущее положение является нулевой точкой Метод возврата в нулевую точку: 9 ~ A 9 = Возврат в прямом направлении; точка столкновения является нулевой точкой A = Возвращение в обратном направлении; точка столкновения является нулевой точкой

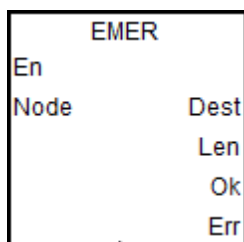
API	Код инструкции			Операнды								Функция			
2811	EMER			<b>Node, Dest, Len, Ok, Err</b>								Чтение аварийного сообщения			

Объекты	X	Y	M	S	T	C	HC	D	FR	SM	SR	E	K	16#	“\$”	F
Node								●					○	○		
Dest								●								
Len								●								
Ok		●	●	●												
Err		●	●	●												

Тип данных	BOOL	WORD	DWORD	LWORD	UINT	INT	DINT	LINT	REAL	LREAL	TMR	CNT	STRING
Node		●											
Dest		●											
Len		●											
Ok	●												
Err	●												

Импульсная инструкция	16-битная инструкция	32-битная инструкция
—	AS	—

**Символ:**



- Node** : Указанный идентификатор узла (ID)
- Dest** : Целевой операнд сохранения данных
- Len** : Общее количество 4 слов прочитанных данных
- Ok** : Флаг выполнения
- Err** : Флаг ошибки

**Описание:**

1. Инструкция EMER применима к режиму CANOpen DS301 и специальному режиму Delta.
2. Поддерживается только прошивка ПЛК от V1.04.10 и выше.
3. После получения аварийного сообщения от Slave узла ПЛК автоматически сохранит данные в заданном операнде сохранения и включит флаг **Ok**.
4. Рекомендуется, чтобы значение узла было указано из уже существующих идентификаторов Slave узлов. Если значение не является одним из существующих идентификаторов узлов или Slave устройство отключено, ПЛК не сможет получить какое-либо сообщение, включится флаг **Err** и будет отображен код ошибки паузы связи (см. описание SR659).
5. Способ, которым инструкция EMER считывает сообщения об аварийных ситуациях, аналогична методу

аварийной связи, описанному в Руководстве по эксплуатации ПЛК серии AS. Выберите один из методов связи при чтении сообщений об ошибках.

6. Инструкция EMER может читать максимум 5 сообщений об авариях. Каждый раз, когда чтение выполняется успешно, включается флаг **Ok**, а **Len** отображает общее количество прочитанных сообщений. Длина (число занятых слов) сохраняется в **Dest**. Каждое сообщение использует 4 слова. Данные хранятся в порядке от младших 8 бит до старших 8 бит. Формат хранения показан ниже. (Например, **Dest** - D10, **Len** - 2, что является количеством сообщений, хранящихся в D5).

№ операнда D	Значение
D5	2

№ операнда D	Младшие 8 бит	Старшие 8 бит
D10	2-й байт 1-го сообщения	1-й байт 1-го сообщения
D11	4-й байт 1-го сообщения	3-й байт 1-го сообщения
D12	6-й байт 1-го сообщения	5-й байт 1-го сообщения
D13	8-й байт 1-го сообщения	7-й байт 1-го сообщения
D14	2-й байт 2-го сообщения	1-й байт 2-го сообщения
D15	4-й байт 2-го сообщения	3-й байт 2-го сообщения
D16	6-й байт 2-го сообщения	5-й байт 2-го сообщения
D17	8-й байт 2-го сообщения	7-й байт 2-го сообщения

# Глава 7 Поиск и устранение неисправностей

## Содержание

<b>7.1</b>	<b>Поиск неисправностей</b> .....	<b>7-2</b>
7.1.1	Основные шаги по поиску неисправностей .....	7-2
7.1.2	Сброс ошибок .....	7-2
7.1.3	Блок-схема устранения неисправностей .....	7-3
7.1.4	Функция System Log .....	7-4
<b>7.2</b>	<b>Поиск неисправностей для модулей ЦПУ</b> .....	<b>7-5</b>
7.2.1	Светодиодный индикатор ERROR включен .....	7-5
7.2.2	Светодиодный индикатор ERROR мигает с частотой 0,5 сек .....	7-5
7.2.3	Светодиодный индикатор ERROR мигает с частотой 0,2 сек .....	7-7
7.2.4	Светодиодный индикатор ERROR медленно мигает каждые 3 сек, загораясь на 1 сек .....	7-7
7.2.5	Светодиодный индикатор BAT. LOW включен .....	7-7
7.2.6	Светодиодный индикатор BAT. LOW мигает с частотой 0,5 сек .....	7-7
7.2.7	Светодиодные индикаторы RUN и ERROR одновременно мигают с частотой 0,5 сек .....	7-8
7.2.8	Светодиодные индикаторы RUN и ERROR попеременно мигают с частотой 0,5 сек .....	7-8
7.2.9	Прочие ошибки (без светодиодной индикации) .....	7-8
<b>7.3</b>	<b>Поиск неисправностей для модулей входов/выходов</b> .....	<b>7-15</b>
7.3.1	Поиск неисправностей для модулей аналоговых входов/выходов (AD/DA/XA) и температурных модулей (RTD/TC) .....	7-15
7.3.2	Поиск неисправностей для весового модуля AS02LC .....	7-16
7.3.3	Поиск неисправностей для модуля коммуникации AS00SCM .....	7-16
7.3.4	Поиск неисправностей для удаленного модуля AS00SCM .....	7-18
<b>7.4</b>	<b>Коды ошибок и светодиодная индикация для модулей ЦПУ</b> .....	<b>7-19</b>
7.4.1	Коды ошибок и светодиодная индикация для модулей ЦПУ .....	7-20
7.4.2	Коды ошибок и светодиодная индикация для модулей аналоговых входов / выходов и температурных модулей .....	7-25
7.4.3	Коды ошибок и светодиодная индикация для весового модуля AS02LC .....	7-25
7.4.4	Коды ошибок и светодиодная индикация для модуля коммуникации AS00SCM .....	7-26
7.4.5	Коды ошибок и светодиодная индикация для удаленного модуля AS00SCM .....	7-26

## 7.1 Поиск неисправностей

### 7.1.1 Основные шаги по поиску неисправностей

В этой главе описаны возможные ошибки и неисправности во время работы и причины их возникновения, а также, какие действия необходимо предпринять для их устранения.

(1) Пункты начальной проверки:

- Проверьте правильность условий эксплуатации ПЛК (окружающая среда, электробезопасность и помехозащищенность, защита от вибраций)
- Проверьте правильность подключения и подачи питания на ПЛК.
- Проверьте правильность установки модулей, клеммных блоков и подключение кабелей.
- Проверьте корректность работы светодиодных индикаторов.
- Проверьте работоспособность всех переключателей.

(2) Проверка работоспособности ПЛК:

- Переключите RUN/STOP
- Проверьте настройки ПЛК при переключении RUN/STOP
- Проверьте и устраните ошибки периферийных устройств
- Используйте функцию System Log в ПО ISPSoft для проверки работы системы и содержимого журналов.

(3) Определение возможных источников ошибок и неисправностей:

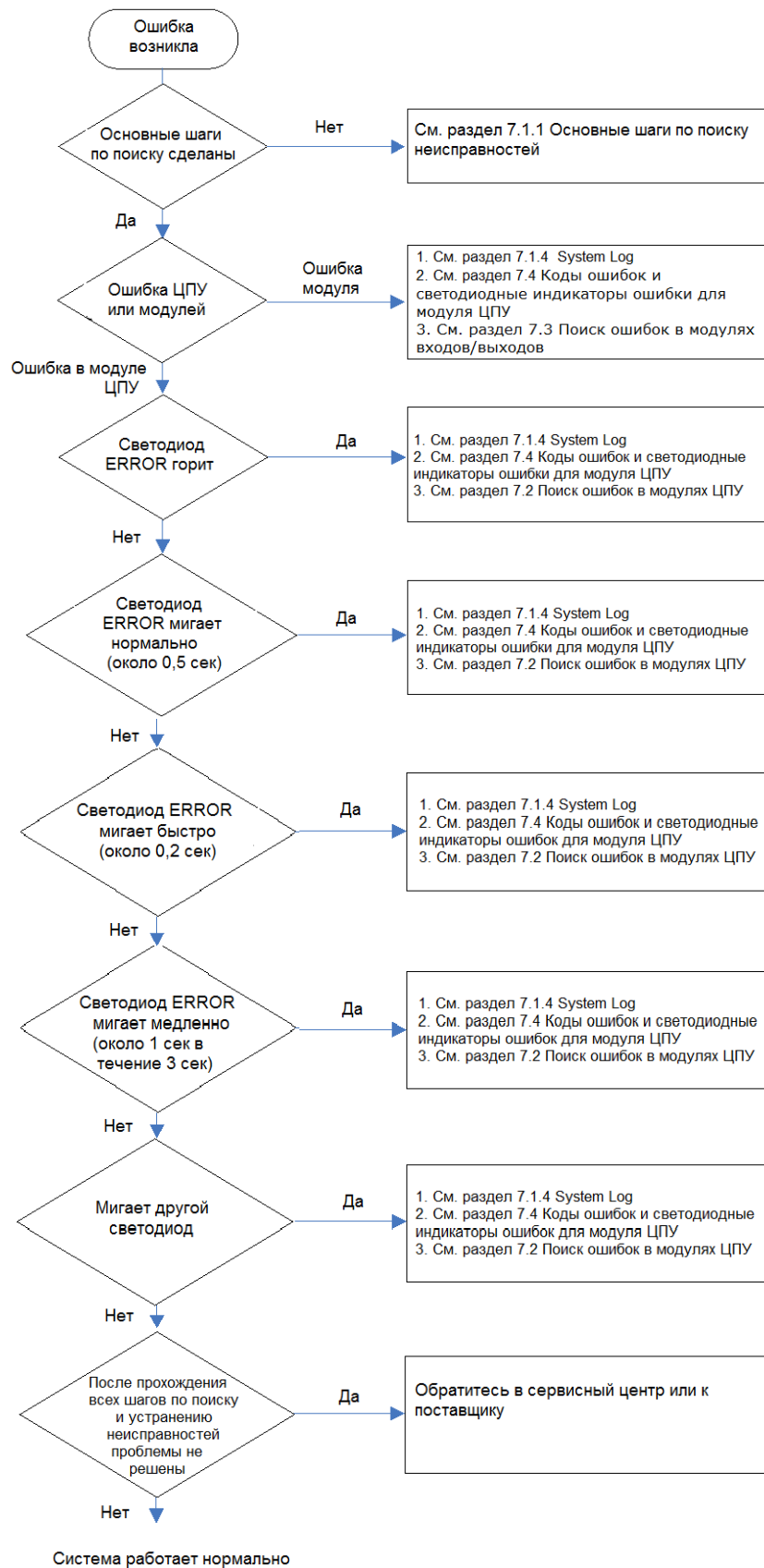
- ПЛК AS или периферийное устройство
- Модуль ЦПУ или модули расширения
- Настройки параметров или программы

### 7.1.2 Сброс ошибок

Используйте следующие методы для сброса состояния ошибки в ПЛК. Необходимо помнить, что если источник ошибки не устранен, система снова будет показывать ошибку.

- (1) Выключите, а затем снова включите ПЛК.
- (2) Выключите, а затем снова включите модуль ЦПУ.
- (3) Очистите журнал ошибок с помощью ПО ISPSoft.
- (4) Перегрузите модуль ЦПУ, сбросьте настройки на заводские и снова загрузите проект.

### 7.1.3 Блок-схема устранения неисправностей

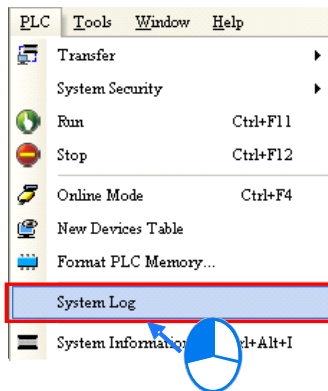




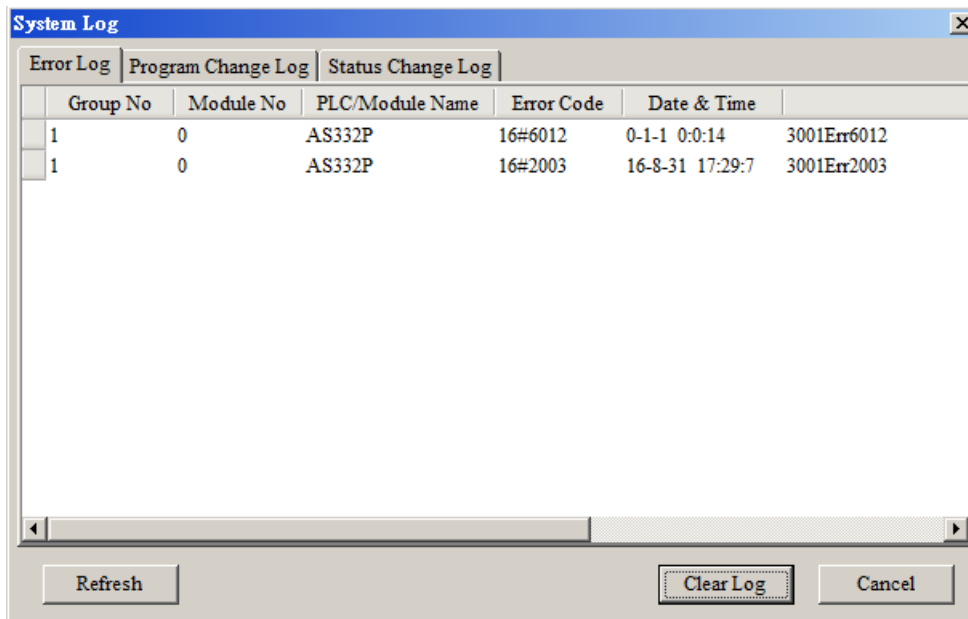
### 7.1.4 Функция System Log

Если ISPSOft работает с ПЛК AS, пользователи могут просматривать действия и ошибки, возникающие в работе ПЛК, нажав кнопку «Системный журнал» в меню PLC. В CPU может храниться до 20 записей ошибок. После сохранения 20 наборов первый журнал будет заменен на 21-й, если появляется новый журнал. Старые журналы будут заменены на новые соответственно. Когда в модуле CPU установлена карта памяти, на карте памяти будут скопированы 20 старых журналов и всего может быть записано до 10000 журналов. Если новый сохраненный журнал превышает ограничение 10000 на карте памяти, самые старые 20 журналов будут заменены новыми 20 журналами.

(1) Нажмите **System Log** в меню **PLC**. PLC> System Log.



(2) Откроется окно **System Log**. После нажатия **Clear Log**, журнал ошибок в окне и на модуле CPU будет сброшен, модуль CPU будет перезагружен.



- **Group No.:** Число 1 указывает на ошибку, возникшую в модуле ЦП или в правом модуле 1. Числа 2 ~ 16 указывают на ошибку, произошедшую в удаленном модуле 1 ~ 15.
- **Module No.:** Число 0 указывает на ошибку, возникшую в модуле CPU или удаленном модуле. Числа 1 ~ 32 указывают на ошибку, произошедшую в модуле с правой стороны от модуля CPU / удаленного модуля. (Число 1 указывает на ближайший модуль к модулю CPU или удаленному модулю, это число увеличивается от самого ближнего к самому дальнему от модуля CPU или удаленного модуля)

Примечание. К правой стороне удаленного модуля могут быть подключены до 8 модулей расширения.

- **PLC/Module name:** Указывает модуль CPU, удаленный модуль или модуль расширения.
- **Error Code:** Код ошибки в журнале.
- **Date & Time:** Время и дата возникновения ошибки. Последняя возникшая ошибка будет наверху.
- В последней колонке приведены относительные описания ошибок.

## 7.2 Поиск неисправностей для модулей ЦПУ

Проверьте светодиодные индикаторы и коды ошибок модуля ЦПУ и обратитесь к таблице ниже для устранения неисправностей. V в столбце «Журнал» указывает, что ошибка будет записана в журнале; X в столбце «Журнал» указывает, что ошибка не будет записана в журнале; H в столбце «Журнал» указывает возможность записи ошибки в HWCONFIG.

### 7.2.1 Светодиодный индикатор ERROR включен

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
000A	Превышение времени ожидания цикла сканирования	1. Проверьте настройки сторожевого таймера в HWCONFIG. 2. Проверьте время сканирования программы	SM8	V

### 7.2.2 Светодиодный индикатор ERROR мигает с частотой 0,5 сек

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
000C	Программа, загруженная в ПЛК, некорректна	Загрузите программу снова	SM9	V
0010	Доступ к памяти в ЦПУ запрещен	Обратитесь к поставщику	SM9	V
002E	Отказ в доступе во внешнюю память ЦПУ.	Обратитесь к поставщику	SM9	V
002F	Программы ПЛК не соответствуют системным журналам	Загрузите программу снова	SM34	V
0070	Фактическое расположение плат расширения не соответствует настройкам	Проверьте, соответствуют ли настройки в HWCONFIG фактическому расположению плат расширения	SM10	V
0102	Номер прерывания превышает допустимый	Проверьте программу, скомпилируйте ее еще раз и снова загрузите	SM5	X
0202	Параметры инструкции MC выходят за допустимые пределы	Проверьте программу, заново скомпилируйте и загрузите ее.	SM5	X
0302	Параметры инструкции MCR выходят за допустимые пределы	Проверьте программу, заново скомпилируйте и загрузите ее.	SM5	X
0D03	Операнды в DHSCS применяются неверно.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM5	X
0E05	Операнды HCXXX, применяемые в DCNT используются неверно.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM5	X

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
1300 ~ 130F	Ошибки возникают в удаленных модулях	См. раздел 12.3.4 для определения кодов ошибок в удаленных модулях.	SM30	V
1402	Фактическое расположение модулей входов/выходов не соответствует таблице модулей.	Проверьте, соответствуют ли настройки в HWCONFIG фактическому расположению модулей входов/выходов.	SM10	V
140B	Модулей коммуникации более 4.	Проверьте количество модулей коммуникации в системе.	SM10	V
140D	Модулей расширения более 32.	Проверьте количество модулей расширения в системе	SM10	V
140E	Число удаленных модулей справа от модуля ЦПУ более 8	Проверьте число удаленных модулей справа от модуля ЦПУ.	SM30	V
1600	Номер ID модуля расширения находится вне допустимого диапазона.	1. Убедитесь, что модуль правильно подключен к модулю ЦПУ и снова включите его. 2. Если ошибка остается, обратитесь к поставщику.	SM10	V
1601	Номер ID модуля расширения не может быть задан.	1. Убедитесь, что модуль правильно подключен к модулю ЦПУ и снова включите его. 2. Если ошибка остается, обратитесь к поставщику.	SM10	V
1602	Номер ID модуля расширения дублируется.	1. Убедитесь, что модуль правильно подключен к модулю ЦПУ и снова включите его. 2. Если ошибка остается, обратитесь к поставщику.	SM10	V
1603	Модуль расширения не работает.	1. Убедитесь, что модуль правильно подключен к модулю ЦПУ и снова включите его. 2. Если ошибка остается, обратитесь к поставщику.	SM10	V
1604	Превышение времени ожидания отклика для модуля коммуникации	1. Убедитесь, что модуль правильно подключен к модулю ЦПУ и снова включите его. 2. Если ошибка остается, обратитесь к поставщику.	SM10	V
1605	Аппаратная неисправность	Обратитесь к поставщику.	SM10	V
1606	Ошибка платы расширения для модуля коммуникации	Убедитесь, что модуль правильно подключен к модулю ЦПУ и снова включите его.	SM10	V
1607	Внешнее напряжение неправильно.	Проверьте правильность подключения внешнего источника питания 24 В к модулю.	SM10	V
1608	Заводская настройка SJC неправильна.	Обратитесь к поставщику.	SM10	V
1609 ~ 160F	Зарезервированы (коды ошибок модулей расширения)			
200A	Неверная инструкция	Проверьте программу, скомпилируйте ее еще раз и снова загрузите.	SM5	V
6010	Превышено количество подключений MODBUS TCP.	Проверьте, чтобы количество подключений не превышало 32.	SM 1092	V
6011	Превышено количество подключений EtherNet/IP.	Проверьте, чтобы количество подключений не превышало 16.	SM 1093	V

### 7.2.3 Светодиодный индикатор ERROR мигает с частотой 0,2 сек

Ошибка возникает при отключении питания 24 В DC на модуле ЦПУ, или если питание ЦПУ является недостаточным или нестабильным.

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
002A	Внешнее питание является некорректным.	Проверьте питание 24 В DC для ЦПУ.	SM7	V

### 7.2.4 Светодиодный индикатор ERROR медленно мигает каждые 3 сек, загораясь на 1 сек

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
1500	Потеря соединения с удаленным модулем	Проверьте соединительный кабель.	SM30	V
1502 ~ 150F	Ошибка в удаленном модуле	См. раздел 12.3.4 для определения кодов ошибок в удаленных модулях.	SM30	V
1800 ~ 180F	Ошибка в модуле расширения	См. раздел 12.3 для определения кодов ошибок в модулях расширения.	SM10	V
1900 ~ 191C	Ошибка соединения с ведомыми сервоприводами Delta ASD-A2.	1. Проверьте кабель CANopen. 2. Проверьте ведомый, с которым возникает ошибка. Примечание: последние 2 цифры кода ошибки обозначают ID ведомого (шестнадцатеричное значение переводится в десятичное).	-	V

### 7.2.5 Светодиодный индикатор BAT. LOW включен

Ошибка возникает при низком заряде или при отсутствии батареи (CR1620). Пользователь может отключить этот параметр в HWCONFIG> CPU> Device Setting> Show Battery Low Voltage Error CPU, если функция RTC для работы с часами реального времени не нужна (по умолчанию данная функция включена).

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
0027	Низкий заряд / отсутствие батареи	Замените батарею или отключите данную функцию	SM219	X

### 7.2.6 Светодиодный индикатор BAT. LOW мигает с частотой 0,5 сек

Ошибка возникает, когда RTC не отслеживает реальное время.

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
0026	RTC не отслеживает текущее время	Обратитесь к поставщику.	SM218	V

### 7.2.7 Светодиодные индикаторы RUN и ERROR одновременно мигают с частотой 0,5 сек.

Происходит при обновлении прошивки модуля ЦПУ. Если эти процессы отображаются после подачи питания на модуль ЦПУ, это означает, что произошла ошибка во время предыдущего обновления прошивки. Необходимо обновить прошивку еще раз или обратиться к поставщику.

### 7.2.8 Светодиодные индикаторы RUN и ERROR попеременно мигают с частотой 0,5 сек.

Происходит, когда на карте памяти в модуле ЦПУ происходят процессы резервирования / восстановления / сохранения данных.

### 7.2.9 Прочие ошибки (без светодиодной индикации)

Код ошибки (16#)	Описание	Решение	Флаг	Журнал
0011	Некорректный номер ID ПЛК.	Проверьте номер ID ПЛК.	SM34	V
0012	Некорректный пароль ПЛК.	Проверьте пароль ПЛК.	SM34	V
002D	Превышено количество попыток ввода пароля в ПЛК.	Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские.	SM34	V
0050	Ошибка памяти в энергонезависимых специальных вспомогательных реле.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V
0051	Энергонезависимые специальные регистры данных некорректны.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V
0052	Ошибка памяти в энергонезависимых вспомогательных реле.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V
0054	Энергонезависимые счетчики некорректны.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V
0055	Энергонезависимые 32-битные счетчики некорректны.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V
0056	Энергонезависимое специальное вспомогательное реле является некорректным.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V

0059	Энергонезависимые регистры данных являются некорректными.	1. Перезагрузите модуль ЦПУ или сбросьте его настройки на заводские, затем снова загрузите программу или параметры. 2. Если ошибка остается, обратитесь к поставщику.	SM6	V
005D	Модуль ЦПУ не обнаруживает карту памяти.	Проверьте, правильно ли вставлена карта памяти в модуль ЦПУ	SM453	V
005E	Неправильная инициализация карты памяти.	Проверьте работоспособность карты памяти.	SM453	V
0063	Ошибка записи данных на карту памяти.	Проверьте правильность указанного пути для записи или работоспособность карты памяти.	SM453	V
0064	Данные с карты памяти не считываются.	Проверьте правильность указанного пути для считывания данных или работоспособность карты памяти.	SM453	V
1950	Инициализация Delta ASD-A2 не завершена, инструкции CANOpen не могут быть выполнены.	1. Проверьте кабель CANopen. 2. Проверьте ведомый, с которым возникает ошибка. 3. Снова инициализируйте Delta ASD-A2.	-	V
2001	Не используется карта FCOMP или неправильный режим для ASDA-A2 при использовании инструкций коммуникации CANOpen	Обязательно применяйте карту FCOMP на функциональной плате 2 и проверьте правильность режима работы.	SM0	V
2003	Значение программного объекта выходит за пределы допустимого диапазона.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
200B	Значение операнда n или других констант K / H выходят за пределы допустимого диапазона.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
200C	Значения операндов перекрываются.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
200D	Неправильное двоичное преобразование двоично-десятичного кода.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
200E	Строка не заканчивается символом 00.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
2012	Неправильная операция деления	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
2013	Значение превышает допустимый диапазон для чисел с плавающей запятой.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
2014	Задача, обозначенная как TKON / YKOFF, некорректна или выходит за пределы допустимого диапазона.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
2017	Инструкция BREAK записывается вне процедуры FOR-NEXT.	Проверьте программу, заново скомпилируйте и загрузите ее.	SM0	V
2027	Несуществующий номер в таблице позиционирования	1. Проверьте программу, заново скомпилируйте и загрузите ее. 2. Проверьте настройки таблицы позиционирования.	SM0	V
2028	Выполняется инструкция для высокоскоростного выхода. Одновременно может выполняться только одна инструкция	См. SR28 для просмотра номера выхода и измените порядок управления выходом.	-	V

6004	Фильтр IP-адреса установлен неправильно.	Снова установите параметры Ethernet для модуля CPU в HWCONFIG	SM1108	X
600D	Порт RJ45 не подключен.	Проверьте подключение.	SM1100	X
6012	Устройства используют одинаковый IP адрес.	1. Проверьте, есть ли в системе устройства, использующие один и тот же IP-адрес. 2. Проверьте наличие в сети более одного DHCP или BOOTP-сервера.	SM1101	V
6100	Потеряно соединение email.	Повторите попытку подключения по электронной почте позже. (Эта ошибка не приводит к остановке работы ПЛК. Пользователь может выполнить это действие с помощью соответствующего флага в программе)	SM1113	X
6103	Неправильно установлен режим вставки триггера в электронном письме.	Настройка режима подключения триггера в HWCONFIG > CPU Module > Device Setting > Options > Ethernet Port Advanced > Email > Trigger Setting > Trigger Attachment Mode.	SM1113	X
6104	Вложения в письме email не существует	Проверьте, существует ли вложение на карте памяти.	SM1113	X
6105	Вложение в электронном письме имеет большой размер.	Проверьте размер вложения. Если размер превышает 2 МБ, файл не может быть отправлен как вложение	SM1113	X
6106	Неверный адрес SMTP-сервера.	Проверьте корректность адреса и настройте SMTP-сервер в HWCONFIG > CPU Module > Device Setting > Options > Ethernet Port Advanced > Email again.	SM1113	X
6107	Превышение времени ожидания ответа SMTP-сервера.	1. Проверьте состояние SMTP-сервера. 2. Повторите попытку подключения по электронной почте позже. (Эта ошибка не приводит к остановке работы ПЛК. Пользователь может выполнить это действие с помощью соответствующего флага в программе)	SM1113	X
6108	Ошибка проверки SMTP	Проверьте правильность номера ID/пароля и проведите настройку в HWCONFIG > CPU Module > Device Setting > Options > Ethernet Port Advanced > Email again.	SM1113	X
6200	IP-адрес удаленной связи, установленный для сокета TCP, является некорректным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6201	Локальный порт связи, установленный для сокета TCP, является некорректным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6202	Удаленный порт связи, установленный в функции сокета TCP, является некорректным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X

6203	Устройство, с которого данные отправляются для сокета TCP, является незаконным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6206	Устройство, которое получает данные для сокета TCP, является незаконным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6208	Данные, полученные через сокет TCP, выходят за пределы допустимого диапазона.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6209	IP-адрес удаленной связи, установленный для сокета UDP, является некорректным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
620A	Локальный порт связи, установленный для сокета UDP, является некорректным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
620C	Устройство, с которого данные отправляются для сокета UDP, является незаконным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
620F	Устройство, которое получает данные для сокета UDP, является незаконным.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6210	Данные, полученные через сокет UDP, выходят за пределы допустимого диапазона.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.	-	X
6212	Превышение времени ожидания ответа от удаленного устройства.	Убедитесь, что удаленное устройство подключено.	-	X
6213	Полученные данные выходят за пределы допустимого диапазона.	1. Проверьте программу и соответствующие специальные регистры данных. 2. Задайте параметры Ethernet для модуля ЦПУ	-	X



		в HWCONFIG CPU Module > Device Setting > Options > Ethernet Port Advanced > TCP Socket.		
6214	Отказ от соединения от удаленного устройства.	Убедитесь, что удаленное устройство работает нормально.	-	X
6215	Сокет не открывается.	Проверьте правильность последовательности действий в программе.	-	X
6217	Сокет открыт.	Проверьте правильность последовательности действий в программе.	-	X
6218	Данные были отправлены через сокет.	Проверьте правильность последовательности действий в программе.	-	X
6219	Данные были получены через сокет.	Проверьте правильность последовательности действий в программе.	-	X
621A	Сокет закрыт.	Проверьте правильность последовательности действий в программе.	-	X
7011	Код коммуникации устройства через COM1 неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7012	Адрес устройства, используемый в связи через COM1, неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7013	Значение операнда, используемое в связи через COM1, выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7014	Длина данных, в процессе коммуникации через COM1 выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7017	Неправильная контрольная сумма устройства для коммуникации через COM1.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7021	Код коммуникации устройства через COM2 неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7022	Адрес устройства, используемый в связи через COM2, неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7023	Значение операнда, используемое в связи через COM2, выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7024	Длина данных, в процессе коммуникации через COM2 выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	H
7027	Неправильная контрольная сумма устройства для коммуникации через COM2.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве.	-	H

		2. Проверьте кабель связи.		
7031	Код коммуникации устройства через Ethernet неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7032	Адрес устройства, используемый в связи через Ethernet, неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7033	Значение операнда, используемое в связи через Ethernet, выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7034	Длина данных, в процессе коммуникации через Ethernet выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7037	Неправильная контрольная сумма устройства для коммуникации через Ethernet.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7041	Код коммуникации устройства через USB неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7042	Адрес устройства, используемый в связи через USB, неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7043	Значение операнда, используемое в связи через USB, выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7044	Длина данных, в процессе коммуникации через USB выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7047	Неправильная контрольная сумма устройства для коммуникации через USB.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70B1	Код коммуникации устройства через функциональную плату 1 неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70B2	Адрес устройства, используемый в связи через функциональную плату 1, неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70B3	Значение операнда, используемое в связи через функциональную плату 1, выходит за пределы	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом	-	Н

	допустимого диапазона.	устройстве. 2. Проверьте кабель связи.		
70B4	Длина данных, в процессе коммуникации через функциональную плату 1 выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70B7	Неправильная контрольная сумма устройства для коммуникации через функциональную плату 1.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70C1	Код коммуникации устройства через функциональную плату 2 неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70C2	Адрес устройства, используемый в связи через функциональную плату 2, неверен.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70C3	Значение операнда, используемое в связи через функциональную плату 2, выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70C4	Длина данных, в процессе коммуникации через функциональную плату 2 выходит за пределы допустимого диапазона.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
70C7	Неправильная контрольная сумма устройства для коммуникации через функциональную плату 2.	1. Проверьте настройку связи в ведущем устройстве и настройку связи в ведомом устройстве. 2. Проверьте кабель связи.	-	Н
7203	Недопустимый код функции коммуникации	Обратитесь к кодам функций, определенным протоколом коммуникации	-	Н
8105	Содержимое загруженной программы неверно. Синтаксическая ошибка.	Загрузите программу и параметры снова.	-	Н
8106	Содержимое загруженной программы неверно. Длина кода выполнения превышает допустимый лимит.	Загрузите программу и параметры снова.	-	Н
8107	Содержимое загруженной программы неверно. Длина исходного кода превышает допустимый предел.	Загрузите программу и параметры снова.	-	Н

## 7.3 Поиск неисправностей для модулей входов/выходов

### • Общие сведения

Модули дискретных входов/выходов, модули аналоговых входов/выходов, модули измерения температуры, модули измерения давления и сетевые модули могут быть установлены в системе ПЛК серии AS. Существует 2 типа кодов, для ошибок и для предупреждений. Модуль ЦПУ и прочие модули перестанут работать при возникновении ошибки и не прекратят работу при выдаче предупреждений.

### 7.3.1 Поиск неисправностей для модулей аналоговых входов/выходов (AD/DA/XA) и температурных модулей (RTD/TC)

#### 7.3.1.1 Светодиодный индикатор ERROR включен

Необходимо настроить отображение в HWCONFIG следующих ошибок.

Код ошибки	Описание	Решение
16#1605	Аппаратный сбой	Обратитесь к поставщику.
16#1607	Недопустимое внешнее напряжение.	Проверьте источник питания.
16#1608	Неправильная заводская калибровка или CJC	Обратитесь к поставщику.

#### 7.3.1.2 Светодиодный индикатор ERROR мигает с частотой 0,5 сек.

В таблице указаны коды предупреждений для обеспечения работоспособности модуля ЦПУ при возникновении ошибок в модулях аналоговых входов/выходов. Настройка в HWCONFIG позволяет отображать первые 4 ошибки.

Код ошибки	Описание	Решение
16#1801	Недопустимое внешнее напряжение.	Проверьте источник питания.
16#1802	Аппаратный сбой	Обратитесь к поставщику.
16#1804	Неправильная заводская калибровка.	Обратитесь к поставщику.
16#1807	Неверный CJC.	Обратитесь к поставщику.
16#1808	Сигнал, полученный по каналу 1, превышает допустимый диапазон для аналоговых входов (температура).	Проверьте получение сигнала по каналу 1
16#1809	Сигнал, полученный по каналу 2, превышает допустимый диапазон для аналоговых входов (температура).	Проверьте получение сигнала по каналу 2
16#180A	Сигнал, полученный по каналу 3, превышает допустимый диапазон для аналоговых входов (температура).	Проверьте получение сигнала по каналу 3
16#180B	Сигнал, полученный по каналу 4, превышает допустимый диапазон для аналоговых входов (температура).	Проверьте получение сигнала по каналу 4

## 7.3.2 Поиск неисправностей для весового модуля AS02LC

### 7.3.2.1 Светодиодный индикатор ERROR включен

Настройка отображения производится в HWCONFIG.

Код ошибки	Описание	Решение
16#1605	Аппаратный сбой (плата драйверов)	Обратитесь к поставщику.
16#1607	Недопустимое внешнее напряжение.	Проверьте источник питания.

### 7.3.2.2 Светодиодный индикатор ERROR мигает с частотой 0,5 сек.

В таблице указаны коды предупреждений для обеспечения работоспособности модуля ЦПУ при возникновении ошибок в весовом модуле. Настройка в HWCONFIG позволяет отображать первые 3 ошибки.

Код ошибки	Описание	Решение
16#1801	Недопустимое внешнее напряжение.	Проверьте источник питания.
16#1802	Аппаратный сбой	Обратитесь к поставщику.
16#1807	Плата драйверов	Обратитесь к поставщику.
16#1808	Сигнал, полученный по каналу 1, превышает допустимый диапазон для аналоговых входов или напряжение SEN является недопустимым.	Проверьте сигнал, полученный по каналу 1 и кабельные соединения.
16#1809	Сигнал, полученный по каналу 1, превышает предельный заданный вес.	Проверьте значение, настроенное для канала 1 и настройку максимального веса.
16#180A	Заводская калибровка для канала 1 неверна.	Проверьте калибровку значения веса в канале 1.
16#180B	Сигнал, полученный по каналу 2, превышает допустимый диапазон для аналоговых входов или напряжение SEN является недопустимым.	Проверьте сигнал, полученный по каналу 2 и кабельные соединения.
16#180C	Сигнал, полученный по каналу 2, превышает предельный заданный вес.	Проверьте значение, настроенное для канала 2 и настройку максимального веса.
16#180D	Заводская калибровка для канала 2 неверна.	Проверьте калибровку значения веса в канале 2.

## 7.3.3 Поиск неисправностей для модуля коммуникации AS00SCM

### 7.3.3.1 Светодиодный индикатор ERROR включен

Для модуля коммуникации AS00SCM, устанавливаемого с правой стороны от модуля ЦПУ, отображаются следующие коды ошибок.

Код ошибки	Описание	Решение
16#1605	Аппаратный сбой	<ol style="list-style-type: none"> <li>1. Проверьте работоспособность модуля.</li> <li>2. Замените модуль AS00SCM или обратитесь к поставщику.</li> </ol>
16#1606	Неправильная настройка функциональной платы.	<ol style="list-style-type: none"> <li>1. Проверьте правильность установки функциональной платы.</li> <li>2. Замените функциональную плату или обратитесь к поставщику.</li> <li>3. Проверьте, соответствует ли настройка в HWCONFIG фактической настройке на функциональной плате.</li> <li>4. Измените и установите новый AS00SCM или свяжитесь с поставщиком.</li> </ol>

### 7.3.3.2 Светодиодный индикатор ERROR мигает с частотой 0,5 сек.

Для модуля коммуникации AS00SCM, устанавливаемом с правой стороны от модуля ЦПУ, отображаются следующие коды ошибок.

Код ошибки	Описание	Решение
16#1802	Неверные параметры	Проверьте параметры в HWCONFIG, снова загрузите параметры.
16#1803	Превышение времени ожидания для коммуникации	<ol style="list-style-type: none"> <li>1. Проверьте подключение кабеля связи.</li> <li>2. Проверьте правильность установки номера станции и формата связи.</li> <li>3. Проверьте, работает ли соединение с функциональной платой.</li> </ol>
16#1804	Неправильная настройка UD Link.	<ol style="list-style-type: none"> <li>1. Проверьте настройки UD Link.</li> <li>2. Проверьте настройки выдачи предупреждений в ПЛК.</li> </ol>

Следующие коды ошибок можно просматривать только через SCMSoft; эти ошибки не будут отображаться на светодиодных индикаторах, и система не будет отправлять сообщения об этих ошибках в модуль ЦПУ.

Код ошибки	Описание	Решение
16#0107	Настройки в HWCONFIG и фактические ручные настройки не соответствуют функциональной плате 1.	Проверьте настройки HWCONFIG и текущие ручные настройки для функциональной платы 1.
16#0108	Настройки в HWCONFIG и фактические ручные настройки не соответствуют функциональной плате 2.	Проверьте настройки HWCONFIG и текущие ручные настройки для функциональной платы 2.
16#0201	Неверные параметры	Проверьте настройки параметров в HWCONFIG и загрузите параметры снова.
16#0301	Превышение времени ожидания коммуникации на функциональной плате 1	<ol style="list-style-type: none"> <li>1. Проверьте правильность установки номера станции и формата связи.</li> <li>2. Проверьте, работает ли соединение с функциональной платой.</li> </ol>
16#0302	Превышение времени ожидания коммуникации на функциональной плате 2	<ol style="list-style-type: none"> <li>1. Проверьте правильность установки номера станции и формата связи.</li> <li>2. Проверьте, работает ли соединение с функциональной платой.</li> </ol>
16#0400	Неправильный ID группы UD Link для функциональной платы 1	<ol style="list-style-type: none"> <li>1. Проверьте настройки UD Link.</li> <li>2. Проверьте настройки выдачи предупреждений в ПЛК.</li> </ol>
16#0401	Неправильный ID группы UD Link для функциональной платы 2	<ol style="list-style-type: none"> <li>1. Проверьте настройки UD Link.</li> <li>2. Проверьте настройки выдачи предупреждений в ПЛК.</li> </ol>
16#0402	Недопустимая команда UD Link для функциональной платы 1	<ol style="list-style-type: none"> <li>1. Проверьте настройки UD Link.</li> <li>2. Проверьте настройки выдачи предупреждений в ПЛК.</li> </ol>
16#0403	Недопустимая команда UD Link для функциональной платы 2	<ol style="list-style-type: none"> <li>1. Проверьте настройки UD Link.</li> <li>2. Проверьте настройки выдачи предупреждений в ПЛК.</li> </ol>

### 7.3.4 Поиск неисправностей для удаленного модуля AS00SCM

Ошибки, возникающие в работе удаленных модулей, передаются как предупреждения на модуль ЦПУ ПЛК серии AS. Светодиодный индикатор модуля ЦПУ при этом будет мигать, а сам модуль ЦПУ будет продолжать работать. Можно использовать флаг SM30 для управления способами отображения ошибок, возникающих в удаленных модулях.

#### 7.3.4.1 Светодиодный индикатор ERROR включен

Коды ошибок:

Код ошибки	Описание	Решение
16#1301	Аппаратный сбой	<ol style="list-style-type: none"> <li>1. Проверьте работоспособность модуля.</li> <li>2. Замените модуль AS00SCM или обратитесь к поставщику.</li> </ol>
16#1302	Некорректные настройки функциональной платы	<ol style="list-style-type: none"> <li>1. Проверьте, надежно ли установлена функциональная плата AS-FCOPM.</li> <li>2. Замените функциональную плату или свяжитесь с поставщиком.</li> <li>3. Проверьте, соответствует ли настройка в HWCONFIG фактической настройке на функциональной плате.</li> <li>4. Замените модуль AS00SCM или свяжитесь с поставщиком.</li> </ol>

#### 7.3.4.2 Светодиодный индикатор ERROR мигает с частотой 0,5 сек.

Коды предупреждений:

Код ошибки	Описание	Решение
16#1502	Некорректные параметры	Проверьте настройки параметров в HWCONFIG и загрузите параметры снова.
16#1503	Превышение времени ожидания связи с удаленным модулем	Убедитесь, что удаленный модуль надежно подключен к модулю ЦПУ и повторно включите его.

#### 7.3.4.3 Светодиодный индикатор ERROR мигает с частотой 0,2 сек

Ошибка возникает, когда питание 24 В постоянного тока для удаленного модуля недостаточно. Проверьте питание. Если источник питания работает нормально, проверьте наличие удаленного модуля. Ниже приведены коды предупреждений.

Код ошибки	Описание	Решение
16#1303	Недостаточный уровень питания 24 В DC и время восстановления напряжения питания менее 10 мс.	Проверьте правильность подключения источника питания к модулю.

## 7.4 Коды ошибок и светодиодная индикация для модулей ЦПУ

### A. Колонки таблиц

- a. Код ошибки: Формируется при возникновении ошибки в системе.
- b. Описание: Описание ошибки
- c. Состояние ЦПУ: При возникновении ошибки ЦПУ может останавливать работу, продолжать работу или его состояние может быть задано пользователем.
  - Стоп: При возникновении ошибки ЦПУ останавливает работу.
  - Работа: При возникновении ошибки ЦПУ продолжает работу.
- d. Состояние светодиодного индикатора: При возникновении ошибки светодиодный индикатор может быть включен, выключен или мигать.
  - ERROR: Системная ошибка

### ● Описание

Модуль ПЛК	Светодиодный индикатор	Описание
цпу	ERROR	<p>Существует 5 типов индикации ошибок для модуля ЦПУ: светодиодный индикатор включен, выключен, быстро мигает, нормально мигает и медленно мигает. Когда светодиодный индикатор включен или быстро / нормально мигает, пользователю необходимо сначала устранить неполадку, чтобы запустить модуль ЦПУ. Когда светодиодный индикатор медленно мигает, указывая на код предупреждения, немедленных действий не требуется. Можно устранить неполадки, когда модуль выключен.</p> <p>Сообщение об ошибке:</p> <p>Включен: Серьезная ошибка модуля.</p> <p>Быстро мигает (около 0.2 сек): Нестабильное питание или аппаратная ошибка.</p> <p>Нормально мигает (около 0.5 сек): Системная программная ошибка или невозможность работы.</p> <p>Предупреждение:</p> <p>Медленно мигает (загорается на 1 сек каждые 3 сек): Выдается предупреждение, но система продолжает работу.</p> <p>Выключен: Выдается предупреждение, но система продолжает работу. Пользователь может настроить отображение предупреждений с помощью SM/SR.</p>



### 7.4.1 Коды ошибок и светодиодная индикация для модулей ЦПУ

Примечание: см. Раздел 12.3 для определения состояния светодиодных индикаторов.

Код ошибки	Описание	Состояние ЦПУ	Состояние светодиодного индикатора ERROR				
			ВКЛ	Быстро мигает	Норм. мигает	Медл. мигает	ВЫКЛ
000A	Превышение времени ожидания цикла сканирования	Останов	V				
000C	Программа, загруженная в ПЛК, некорректна.	Останов			V		
0010	Доступ к памяти в ЦПУ запрещен.	Останов			V		
0011	Некорректный номер ID ПЛК.	Работа					V
0012	Некорректный пароль ПЛК.	Работа					V
0026	RTC не отслеживает текущее время (индикатор батареи мигает)	Работа					
0027	Низкий заряд / отсутствие батареи (индикатор батареи включен)	Работа					
002A	Внешнее питание является некорректным	Работа		V			
002D	Превышено количество попыток ввода пароля в ПЛК.	Работа					V
002E	Отказ в доступе во внешнюю память ЦПУ.	Останов			V		
002F	Программы ПЛК не соответствуют системным журналам.	Останов			V		
0050	Ошибка памяти в энергонезависимых специальных вспомогательных реле.	Работа					V
0051	Энергонезависимые специальные регистры данных некорректны.	Работа					V
0052	Ошибка памяти в энергонезависимых вспомогательных реле.	Работа					V
0054	Энергонезависимые счетчики некорректны.	Работа					V
0055	Энергонезависимые 32-битные счетчики некорректны.	Работа					V
0056	Энергонезависимое специальное вспомогательное реле является некорректным.	Работа					V
0059	Энергонезависимые регистры данных являются некорректными.	Работа					V
005D	Модуль ЦПУ не обнаруживает карту памяти.	Работа					V
005E	Неправильная инициализация карты памяти.	Работа					V
0063	Ошибка записи данных на карту памяти.	Работа					V
0064	Данные с карты памяти не считываются.	Работа					V
0070	Фактическое расположение плат расширения не соответствует настройкам.	Останов			V		
0102	Номер прерывания превышает допустимый.	Останов			V		
0202	Параметры инструкции MC выходят за допустимые пределы	Останов			V		
0302	Параметры инструкции MCR выходят за допустимые пределы	Останов			V		
0D03	Операнды в DHSCS применяются неверно.	Останов			V		
0E05	Операнды HCXXX, применяемые в DCNT используются неверно.	Останов			V		

Код ошибки	Описание	Состояние ЦПУ	Состояние светодиодного индикатора ERROR				
			ВКЛ	Быстро мигает	Норм. мигает	Медл. мигает	ВЫКЛ
1300 ~ 130F	Ошибки возникают в удаленных модулях	Работа				V	
1402	Фактическое расположение модулей входов/выходов не соответствует таблице модулей.	Останов			V		
140B	Модулей коммуникации более 4.	Останов			V		
140D	Модулей расширения более 32.	Останов			V		
140E	Число удаленных модулей справа от модуля ЦПУ более 8	Останов			V		
1500	Потеря соединения с удаленным модулем	Работа				V	
1502 ~ 150F	Ошибка в удаленном модуле	Работа				V	
1600	Номер ID модуля расширения находится вне допустимого диапазона.	Останов			V		
1601	Номер ID модуля расширения не может быть задан.	Останов			V		
1602	Номер ID модуля расширения дублируется.	Останов			V		
1603	Модуль расширения не работает.	Останов			V		
1604	Превышение времени ожидания отклика для модуля коммуникации	Останов			V		
1605	Аппаратная неисправность	Останов			V		
1606	Ошибка платы расширения для модуля коммуникации	Останов			V		
1607	Внешнее напряжение неправильно.	Останов			V		
1608	Заводская настройка CJC неправильна.	Останов			V		
1609 ~ 160F	Зарезервированы (коды ошибок модулей расширения)	Останов			V		
1800 ~ 180F	Ошибка в модуле расширения	Работа				V	
1900 ~ 191C	Ошибка соединения с ведомыми сервоприводами Delta ASD-A2.	Работа				V	
1950	Инициализация Delta ASD-A2 не завершена, инструкции CANopen не могут быть выполнены.	Работа					V
2001	Не используется карта FCOMP или неправильный режим для ASDA-A2 при использовании инструкций коммуникации CANopen.	Работа					V

Код ошибки	Описание	Состояние ЦПУ	Состояние светодиодного индикатора ERROR				
			ВКЛ	Быстро мигает	Норм. мигает	Медл. мигает	ВЫКЛ
2003	Значение программного объекта выходит за пределы допустимого диапазона.	Работа					✓
200A	Неверная инструкция	Останов			✓		
200B	Значение операнда n или других констант K / H выходят за пределы допустимого диапазона.	Работа					✓
200C	Значения операндов перекрываются.	Работа					✓
200D	Неправильное двоичное преобразование двоично-десятичного кода.	Работа					✓
200E	Строка не заканчивается символом 00.	Работа					✓
2012	Неправильная операция деления	Работа					✓
2013	Значение превышает допустимый диапазон для чисел с плавающей запятой.	Работа					✓
2014	Задача, обозначенная как TKON / YKOFF, некорректна или выходит за пределы допустимого диапазона.	Работа					✓
2017	Инструкция BREAK записывается вне процедуры FOR-NEXT.	Работа					✓
2027	Несуществующий номер в таблице позиционирования	Работа					✓
2028	The high speed output instruction is being executed. Only one instruction can be executed at a time.	Работа					✓
6004	Фильтр IP-адреса установлен неправильно.	Работа					✓
600D	Порт RJ45 не подключен.	Работа					✓
6010	Превышено количество подключений MODBUS TCP.	Работа			✓		
6011	Превышено количество подключений EtherNet/IP.	Работа			✓		
6012	Устройства используют одинаковый IP адрес.	Работа					✓
6100	Потеряно соединение email.	Работа					✓
6103	Неправильно установлен режим вставки триггера в электронном письме.	Работа					✓
6104	Вложения в письме email не существует	Работа					✓
6105	Вложение в электронном письме имеет большой размер.	Работа					✓
6106	Превышение времени ожидания ответа SMTP-сервера.	Работа					✓
6107	Превышение времени ожидания ответа SMTP-сервера.	Работа					✓
6108	Ошибка проверки SMTP	Работа					✓
6200	IP-адрес удаленной связи, установленный для сокета TCP, является некорректным.	Работа					✓
6201	Локальный порт связи, установленный для сокета TCP, является некорректным.	Работа					✓
6202	Удаленный порт связи, установленный в функции сокета TCP, является некорректным.	Работа					✓
6203	Устройство, с которого данные отправляются для сокета TCP, является незаконным.	Работа					✓
6206	Устройство, которое получает данные для сокета TCP, является незаконным.	Работа					✓
6208	Данные, полученные через сокет TCP, выходят за пределы допустимого диапазона.	Работа					✓

Код ошибки	Описание	Состояние ЦПУ	Состояние светодиодного индикатора ERROR				
			ВКЛ	Быстро мигает	Норм. мигает	Медл. мигает	ВЫКЛ
6209	IP-адрес удаленной связи, установленный для сокета UDP, является некорректным.	Работа					✓
620A	Локальный порт связи, установленный для сокета UDP, является некорректным.	Работа					✓
620C	Устройство, с которого данные отправляются для сокета UDP, является незаконным.	Работа					✓
620F	Устройство, которое получает данные для сокета UDP, является незаконным.	Работа					✓
6210	Данные, полученные через сокет UDP, выходят за пределы допустимого диапазона.	Работа					✓
6212	Превышение времени ожидания ответа от удаленного устройства.	Работа					✓
6213	Полученные данные выходят за пределы допустимого диапазона.	Работа					✓
6214	Отказ от соединения от удаленного устройства.	Работа					✓
6215	Сокет не открывается.	Работа					✓
6217	Сокет открыт.	Работа					✓
6218	Данные были отправлены через сокет.	Работа					✓
6219	Данные были получены через сокет.	Работа					✓
621A	Сокет закрыт.	Работа					✓
7011	Код коммуникации устройства через COM1 неверен.	Работа					✓
7012	Адрес устройства, используемый в связи через COM1, неверен.	Работа					✓
7013	Значение операнда, используемое в связи через COM1, выходит за пределы допустимого диапазона.	Работа					✓
7014	Длина данных, в процессе коммуникации через COM1 выходит за пределы допустимого диапазона.	Работа					✓
7017	Неправильная контрольная сумма устройства для коммуникации через COM1.	Работа					✓
7021	Код коммуникации устройства через COM2 неверен.	Работа					✓
7022	Адрес устройства, используемый в связи через COM2, неверен.	Работа					✓
7023	Значение операнда, используемое в связи через COM2, выходит за пределы допустимого диапазона.	Работа					✓
7024	Длина данных, в процессе коммуникации через COM2 выходит за пределы допустимого диапазона.	Работа					✓
7027	Неправильная контрольная сумма устройства для коммуникации через COM2.	Работа					✓
7031	Код коммуникации устройства через Ethernet неверен.	Работа					✓
7032	Адрес устройства, используемый в связи через Ethernet, неверен.	Работа					✓
7033	Значение операнда, используемое в связи через Ethernet, выходит за пределы допустимого диапазона.	Работа					✓
7034	Длина данных, в процессе коммуникации через Ethernet выходит за пределы допустимого диапазона.	Работа					✓
7037	Неправильная контрольная сумма устройства для коммуникации через Ethernet.	Работа					✓

Код ошибки	Описание	Состояние ЦПУ	Состояние светодиодного индикатора ERROR				
			ВКЛ	Быстро мигает	Норм. мигает	Медл. мигает	ВЫКЛ
7041	Код коммуникации устройства через USB неверен.	Работа					V
7042	Адрес устройства, используемый в связи через USB, неверен.	Работа					V
7043	Значение операнда, используемое в связи через USB, выходит за пределы допустимого диапазона.	Работа					V
7044	Длина данных, в процессе коммуникации через USB выходит за пределы допустимого диапазона.	Работа					V
7047	Неправильная контрольная сумма устройства для коммуникации через USB.	Работа					V
70B1	Код коммуникации устройства через функциональную плату 1 неверен.	Работа					V
70B2	Адрес устройства, используемый в связи через функциональную плату 1, неверен.	Работа					V
70B3	Значение операнда, используемое в связи через функциональную плату 1, выходит за пределы допустимого диапазона.	Работа					V
70B4	Длина данных, в процессе коммуникации через функциональную плату 1 выходит за пределы допустимого диапазона.	Работа					V
70B7	Неправильная контрольная сумма устройства для коммуникации через функциональную плату 1.	Работа					V
70C1	Код коммуникации устройства через функциональную плату 2 неверен.	Работа					V
70C2	Адрес устройства, используемый в связи через функциональную плату 2, неверен.	Работа					V
70C3	Значение операнда, используемое в связи через функциональную плату 2, выходит за пределы допустимого диапазона.	Работа					V
70C4	Длина данных, в процессе коммуникации через функциональную плату 2 выходит за пределы допустимого диапазона.	Работа					V
70C7	Неправильная контрольная сумма устройства для коммуникации через функциональную плату 2.	Работа					V
7203	Недопустимый код функции коммуникации	Работа					V
8105	Содержимое загруженной программы неверно. Синтаксическая ошибка.	Работа					V
8106	Содержимое загруженной программы неверно. Длина кода выполнения превышает допустимый лимит.	Работа					V
8107	Содержимое загруженной программы неверно. Длина исходного кода превышает допустимый предел.	Работа					V

### 7.4.2 Коды ошибок и светодиодная индикация для модулей аналоговых входов / выходов и температурных модулей

Код ошибки	Описание	Состояние светодиодного индикатора ERROR	
		A → D / D → A / A ↔ D	ERROR
16#1605	Аппаратный сбой	Выключен	Включен
16#1607	Недопустимое внешнее напряжение.	Выключен	Включен
16#1608	Неправильная заводская калибровка или CJC	Выключен	Включен
16#1801*1	Недопустимое внешнее напряжение.	Выключен	Мигает
16#1802*1	Аппаратный сбой	Выключен	Мигает
16#1804*1	Неправильная заводская калибровка.	RUN: Blinking STOP: Выключен	Мигает
16#1807*1	Неверный CJC.	Выключен	Мигает
16#1808	Сигнал, полученный по каналу 1, превышает допустимый диапазон для аналоговых входов (температура).	РАБОТА: Мигает СТОП: Выключен	Мигает
16#1809	Сигнал, полученный по каналу 2, превышает допустимый диапазон для аналоговых входов (температура).		
16#180A	Сигнал, полученный по каналу 3, превышает допустимый диапазон для аналоговых входов (температура).		
16#180B	Сигнал, полученный по каналу 4, превышает допустимый диапазон для аналоговых входов (температура).		

\*1: Указаны коды предупреждений для обеспечения работоспособности модуля ЦПУ при возникновении ошибок в модулях аналоговых входов/выходов. Настройка в HWCONFIG позволяет отображать первые 4 ошибки.

### 7.4.3 Коды ошибок и светодиодная индикация для весового модуля ASO2LC

Код ошибки	Описание	Состояние светодиодного индикатора ERROR	
		A → D	ERROR
16#1605	Аппаратный сбой (плата драйверов)	Выключен	Включен
16#1607	Недопустимое внешнее напряжение.	Выключен	Включен
16#1801*1	Недопустимое внешнее напряжение.	Выключен	Мигает
16#1802*1	Аппаратный сбой	Выключен	Мигает
16#1807*1	Плата драйверов	Выключен	Мигает
16#1808	Сигнал, полученный по каналу 1, превышает допустимый диапазон для аналоговых входов или напряжение SEN является недопустимым.	РАБОТА: Мигает СТОП: Выключен	Мигает
16#1809	Сигнал, полученный по каналу 1, превышает предельный заданный вес.		
16#180A	Заводская калибровка для канала 1 неверна.		

Код ошибки	Описание	Состояние светодиодного индикатора ERROR	
		A → D	ERROR
16#180B	Сигнал, полученный по каналу 2, превышает допустимый диапазон для аналоговых входов или напряжение SEN является недопустимым.		
16#180C	Сигнал, полученный по каналу 2, превышает предельный заданный вес.		
16#180D	Заводская калибровка для канала 2 неверна.		

\*1: Указаны коды предупреждений для обеспечения работоспособности модуля ЦПУ при возникновении ошибок в весовом модуле. Настройка в HWCONFIG позволяет отображать первые 3 ошибки.

#### 7.4.4 Коды ошибок и светодиодная индикация для модуля коммуникации ASOOSCM

Код ошибки	Описание	Состояние светодиодного индикатора ERROR	
		Включен	Мигает
16#1605	Аппаратный сбой	✓	
16#1606	Неправильная настройка функциональной платы.	✓	
16#1802	Неверные параметры		✓
16#1803	Превышение времени ожидания для коммуникации		✓
16#1804	Неправильная настройка UD Link.		✓

#### 7.4.5 Коды ошибок и светодиодная индикация для удаленного модуля ASOOSCM

Код ошибки	Описание	Состояние светодиодного индикатора ERROR		
		Включен	Мигает	Быстро мигает
16#1301	Аппаратный сбой	✓		
16#1302	Некорректные настройки функциональной платы	✓		
16#1303	Недостаточный уровень питания 24 В DC и время восстановления напряжения питания менее 10 мс.			✓
16#1502	Некорректные параметры		✓	
16#1503	Превышение времени ожидания связи с удаленным модулем		✓	