

ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ КОНТРОЛЛЕРЫ DVP

РУКОВОДСТВО ПО ПРОГРАММИРОВАНИЮ

Программируемые логические контроллеры

SS2 / SA2 / SX2 / ES2 / EX2 / SE

Руководство по программированию

ред. от 29.06.2012г. с дополнениями 2013 г.

Перевод и адаптация: компания СТОИК, (495) 661-24-41, www.stoikltd.ru

Содержание

Глава 1. Принципы программирования ПЛК

1.1 Различия между релейно-контактной логикой в ПЛК и физическими релейно-контактными электросхемами 6

1.2 Метод сканирования ПЛК 7

1.3 Регистры и реле ПЛК 7

1.4 Символы релейно-контактных (лестничных) логических диаграмм 8

1.4.1 Создание релейно-контактных (лестничных) программ для ПЛК 9

1.4.2 LD / LDI (Загрузка Н/О контакта / загрузка Н/З контакта) 9

1.4.3 LDP / LDF (Загрузка триггера с опросом по переднему фронту / загрузка триггера с опросом по заднему фронту) 9

1.4.4 AND / ANI (Последовательное соединение Н/О контактов / последовательное соединение Н/З контактов) 9

1.4.5 ANDP / ANDF (Последовательное соединение триггеров с опросом по переднему/заднему фронту) 10

1.4.6 OR / ORI (Параллельное соединение Н/О контактов / параллельное соединение Н/З контактов) 10

1.4.7 ORP / ORF (Параллельное соединение триггеров с опросом по переднему/заднему фронту) 10

1.4.8 ANB (Последовательное соединение блоков) 10

1.4.9 ORB (Параллельное соединение блоков) 10

1.4.10 MPS / MRD / MPP (Узловые команды) 10

1.4.11 STL (Шаговое лестничное программирование) 10

1.4.12 RET (Возврат) 10

1.5 Преобразование лестничной диаграммы в список команд 12

1.6 Различные варианты списков команд (мнемокодов) 13

1.7 Оптимизация лестничных диаграмм 14

1.8 Примеры написания программ 16

Глава 2. Операнды, применяемые при программировании

2.1 ES2/EX2 Карта параметров 22

2.2 SS2 Карта параметров 24

2.3 SA2/SX2 Карта параметров 26

2.4 SE Карта параметров 28

2.5 Установка области энергонезависимой памяти 30

2.6 ПЛК: биты, полубайты, байты, слова, двойные слова 31

2.7 Системы: двоичная, восьмеричная, десятичная, двоично-десятичная, шестнадцатеричная 31

2.8 Реле M 32

2.9 Реле S 45

2.10 Таймер T 45

2.11 Счетчик C 47

2.12 Высокоскоростые счетчики 49

2.13 Специальные регистры данных 53

2.14 Индексные регистры E, F 65

2.15 Указатель уровня вложения[N], Указатель[P], Указатель прерывания [I] 66

2.16 Описания специальных реле M и регистров D 67

Глава 3. Система команд для программирования контроллеров DVP

3.1 Базовые команды (без API номеров) ... 89

3.2 Описание базовых команд ... 90

3.3 Точки перехода ... 96

3.4 Точки прерывания ... 97

3.5 Прикладные команды (API) ... 97

3.6 Перечень команд (по функциональным характеристикам) ... 105

3.7 Перечень команд (по алфавиту) ... 113

3.8 Детальное описание команд ... 120

Глава 4. Параметры коммуникации

4.1 Коммуникационные порты ... 387

4.2 Протокол связи в ASCII-режиме ... 387

4.2.1 ADR (Коммуникационный адрес) ... 388

4.2.2 CMD (Командный код) и DATA (символы данных) ... 388

4.2.3 LRC CHK (Контрольная сумма) ... 389

4.3 Протокол связи в RTU-режиме ... 389

4.3.1 Адрес (Коммуникационный адрес) ... 390

4.3.2 CMD (Командный код) and DATA (символы данных) ... 390

4.3.3 CRC CHK (контрольная сумма) ... 390

4.4 Адреса внутренних устройств ПЛК ... 391

4.5 Командный код ... 393

4.5.1 Командный код: 01, Чтение состояния контактов (за исключением входов X) ... 393

4.5.2 Командный код: 02, Чтение состояния контактов (включая входы X) ... 393

4.5.3 Командный код: 03, Чтение содержимого регистров (T, C, D) ... 394

4.5.4 Командный код: 05, Состояние ВКЛ/ВЫКЛ одного контакта ... 394

4.5.5 Командный код: 06, Установка значения одного регистра ... 395

4.5.6 Командный код: 15, Состояние ВКЛ/ВЫКЛ нескольких контактов ... 395

4.5.7 Командный код: 16, Установка значений нескольких регистров ... 396

Глава 5. Последовательная функциональная диаграмма (SFC). Команды пошагового управления

5.1 Команды пошагового управления [STL], [RET] ... 397

5.2 Последовательная функциональная диаграмма (SFC) ... 397

5.3 Работа STL программы. ... 399

5.4 Особенности программирования пошагового управления. ... 402

5.5 Типы последовательностей ... 403

5.6 Команда IST ... 411

Глава 6. Поиск и устранение неисправностей

6.1 Общие проблемы и решения ... 412

6.2 Таблица кодов ошибок (шестнадцатеричный формат) ... 413

6.3 Отображение и фиксация возникающих ошибок ... 414

Глава 7. Функции и порядок работы с CANopen

7.1 Введение в CANopen 415
7.1.1 Описание функций CANopen 415
7.1.2 Области отображения входов/выходов 416
7.2 Инсталлирование и топология сети 416
7.2.1 Габариты 416
7.2.2 Профиль 417
7.2.3 Интерфейс CAN и топология сети 417
7.3 Протокол CANopen 420
7.3.1 Введение в протокол CANopen 420
7.3.2 Объект связи CANopen 421
7.3.3 Стандартный набор подключений 421
7.4 Отправка SDO, NMT и чтение Аварийных сообщений из программы контроллера 426
7.4.1 Структура данных запроса SDO 426
7.4.2 Структура данных сообщения NMT 428
7.4.3 Структура данных Аварийного запроса 429
7.4.4 Пример отправки SDO через лестничную диаграмму 431
7.5 Индикаторы и устранение неисправностей 432
7.5.1 Описание индикаторов 433
7.5.2 Отображение состояния сетевых узлов CANopen 434
7.6 Пример применения 435
7.7 Словарь объектов 441
Приложение A. Установка драйвера USB 447

Принципы программирования ПЛК

Введение

В этой главе представлены базовые и расширенные определения релейной логики, которые, главным образом, использует язык программирования ПЛК. Пользователи, знакомые с определениями и понятиями ПЛК, могут перейти к следующей главе для дальнейшего знакомства с концепцией программирования ПЛК, для прочих – рекомендуется внимательно прочитать данный раздел.

Программируемые логические контроллеры (далее по тексту ПЛК) серии DVP являются идеальным средством для построения высокоэффективных систем автоматического управления при минимальных затратах на приобретение оборудования и разработку системы.

ПЛК работают следующим образом:

шаг 1: чтение состояния внешних входных устройств (переключатели, датчики, клавиатура);

шаг 2: обработка процессором предварительно заданной программы и установка нового состояния выходов.

Программа состоит из последовательности отдельных управляющих инструкций, которые определяют функции управления. ПЛК обрабатывает инструкции последовательно, т.е. одну за другой. Общий проход программы непрерывно повторяется. Время, необходимое для прохода программы называется временем цикла, а проходы программы – циклическим сканированием.

Контроллеры способны работать в реальном масштабе времени и могут быть использованы как для построения узлов локальной автоматики, так и систем распределенного ввода-вывода с организацией обмена данными по RS-485 интерфейсу.

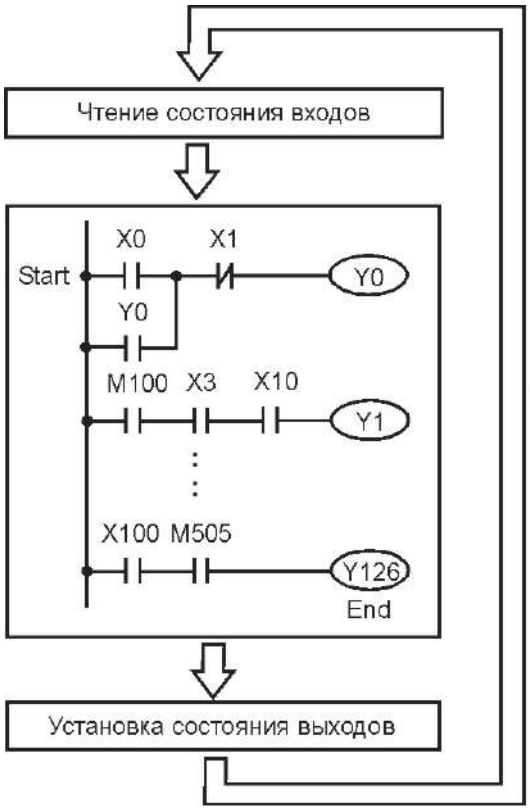
Для удобства отладки и написания программ мы предлагаем воспользоваться одним из бесплатных пакетов программирования WPLSoft или ISPSoft (доступны для скачивания на сайтах www.stoikltd.ru и www.deltronics.ru), которые не требует существенных ресурсов компьютера, и являются простым инструментом для всех категорий специалистов. Используются три языка программирования: LAD (релейно-контактная логика или лестничные диаграммы), IL (список инструкций) , SFC (последовательные функциональные диаграммы).

Настоящее руководство описывает и поясняет все инструкции, операнды, распределение адресов ввода/вывода, которые нужны для написания программы контроллеров DVP серий SS2 / SA2 / SX2 / ES2 / EX2 / SE.

Информация по установке, монтажу, вводу в эксплуатацию, обслуживанию и устранению ошибок есть в соответствующих руководствах по эксплуатации.

1.1 Различия между релейно-контактной логикой в ПЛК и физическими релейно-контактными электросхемами

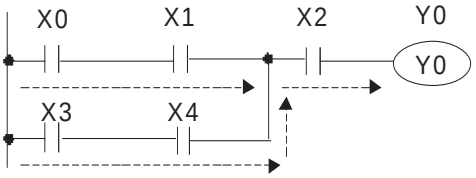
В обычных релейно-контактных электросхемах все задаваемые управляющие процессы выполняются одновременно (параллельно). Каждое изменение состояние входных сигналов сразу же действует на изменение состояния выходных сигналов.
При управлении от ПЛК изменение состояния входных сигналов, произошедшее во время текущего прохода программы, опознается только на следующем цикле программы. Этот недостаток ПЛК сглаживается только благодаря очень короткому времени цикла.
Время выполнения одного цикла программы зависит от количества выполняемых инструкций в программе и от типа используемых инструкций.

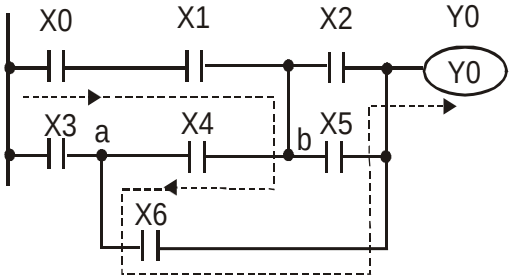


Один цикл программы

В процессе работы ПЛК непрерывно опрашивает текущее состояние входов и в соответствии с требованиями к производственному процессу изменяет состояние выходов (Вкл./Выкл).

1. Проверка текущего состояния входов: ПЛК проверяет текущее состояние входов и в зависимости от значения (Вкл./Выкл) выполняет последовательные действия. Состояние любого из входов сохраняется в памяти (в области данных).
2. Выполнение программы: Будем считать, что в ходе технологического процесса вход X0 переключился с выключено на включено, и в соответствии с технологическим процессом нам необходимо изменить текущее состояние выхода (Y0) с выключено на включено. Так как ЦП опросил текущее состояния всех входов и хранит их текущее состояние в памяти, то выбор последующего действия обусловлен только ходом технологического процесса.
3. Изменение текущего состояния выхода. ПЛК - изменяет текущее состояние выходов в зависимости от того, какие входы являются выключенными, а какие включенными исходов из хода вашей программы. То есть контроллер, физически переключил выход (Y0) и включились исполнительные механизмы лампочка, двигатель и т.д. После этого следует возврат на первый шаг. Еще одним отличием релейно-контактной логики ПЛК от обычных релейно-контактных электросхем заключается в том, что выполнение программ в строках идет только слева на право, а схема с "реверсивным направлением тока" (участок a-b на рис.) при компиляции вызовет ошибку.





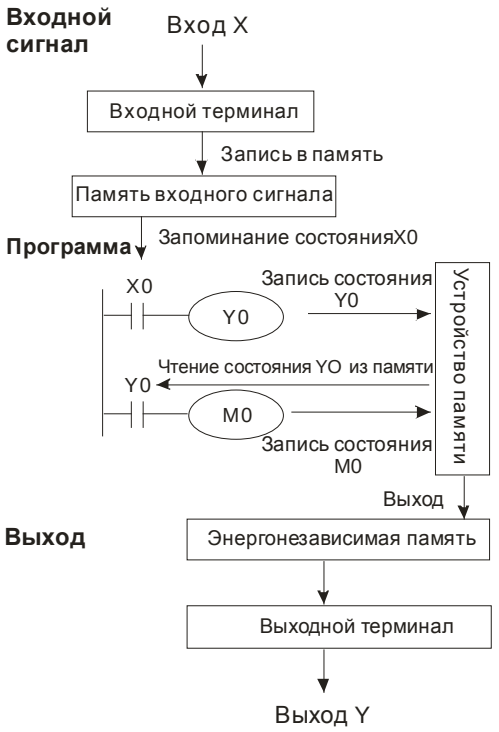
Ошибка в строке 3

1.2 Метод сканирования ПЛК

ПЛК применяет стандартный метод сканирования при выполнении программы.

Процесс сканирования:

Проверка состояния входов	Чтение физического состояния входов и сохранение результата во внутренней памяти
Оценка программы	Оценка программы с данными из внутренней памяти. Сканирование программы проводится «сверху вниз» и «слева направо» до достижения конца программы
Обновление выходов	Запись данных оценки в физические выходы



Входной сигнал: ПЛК читает состояние каждого входа (ВКЛ/ВЫКЛ) и сохраняет значение во внутреннюю память перед проведением оценки программы. После сохранения состояния в памяти любые изменения на входах не будут отображаться до проведения следующего цикла сканирования.

Программа: ПЛК выполняет команды в программе по схеме «сверху вниз» и «слева направо», а затем сохраняет результаты во внутреннюю память. Некоторые из этих данных фиксируются.

Выход: После выполнения команды END оценка программы завершается. Выходные данные передаются из памяти на физические выходы.

Время сканирования

Продолжительность полного цикла сканирования (чтение, оценка, запись) называется «временем сканирования». С увеличением количества входов/выходов и объема программы время сканирования растет.

Чтение времени сканирования	ПЛК измеряет время сканирования и сохраняет значение (0,1 мс) в регистре D1011, максимальное время сканирования – в регистре D1012.
Измерение времени сканирования	Время сканирования может быть измерено путем переключения выходов при каждом сканировании и последующего измерения ширины импульса на выходе.
Вычисление времени сканирования	Время сканирования может быть вычислено путем сложения времени работы всех команд в программе. Для определения времени работы каждой команды см. главу 3 настоящего руководства.

Исключения

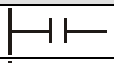
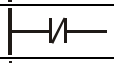
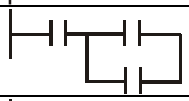
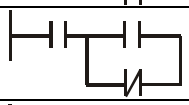
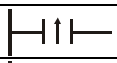
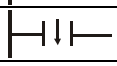
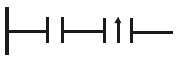
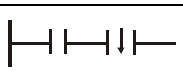
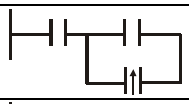
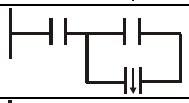
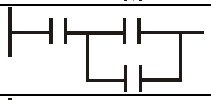
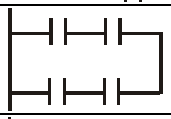
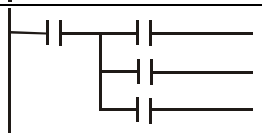
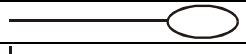
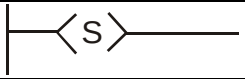
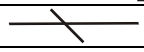
ПЛК может обрабатывать некоторые элементы быстрее времени сканирования. Некоторые из этих элементов могут остановить время сканирования для обработки подпрограммы прерывания. Команда REF для прямого обновления входов/выходов позволяет ПЛК получить доступ к входам/выходам, не дожидаясь окончания цикла сканирования.

1.3 Регистры и реле ПЛК. Введение в основные внутренние компоненты ПЛК

X (входное реле)	Битовая память – представляет внешние физические входы и получает внешние входные сигналы ■ Обозначается как **X,** адресация в восьмеричной системе, т.е. X0~X7, X10~X17…X377
Y (выходное реле)	Битовая память – представляет внешние физические выходы и сохраняют статус их обновления. ■ Обозначаются как **Y,** адресация в восьмеричной системе, т.е. Y0~Y7, Y10~Y17. ..Y377
M (внутреннее реле)	Битовая память – показывает состояние ПЛК. ■ Обозначаются как **M,** адресация в десятичной системе, т.е. M0, M1, M2…M4095
S (шаговое реле)	Битовая память – показывает состояние ПЛК в режиме пошагового управления (SFC). Если команда STL не применяется в программе, S может использоваться как внутреннее реле M или как сигнализатор. ■ Обозначаются как **S,** адресация в десятичной системе, т.е. S0, S1, S2…S1023
T	Битовая, словная, двойная словная память – реле времени, может быть контактом или

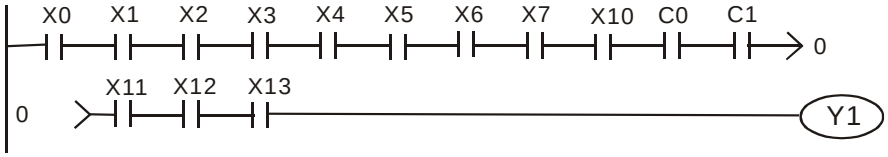
(реле времени - таймер) (слово) (двойное слово)	регистром. Когда катушка включена и установленное в таймере время достигается, включается связанный контакт. Таймер может иметь шаг: 1 мс / 10 мс / 100 мс). ■ Обозначаются как **T,** адресация в десятичной системе, т.е. T0, T1, T2…T255
C (счетчик) (реле) (слово) (двойное слово)	Битовая, словная, двойная словная память – счетчик, может быть контактом или регистром. Счетчик включений катушки (импульсов). Когда значение счетчика достигает заданного значения, включается связанный контакт. Доступны 16- и 32-битные высокоскоростные счетчики. ■ Обозначаются как **C,** адресация в десятичной системе, т.е. C0, C1, C2…C255
D (регистр данных) (слово)	Словная память – сохраняет значение и параметры данных. Каждый регистр может хранить слово (16-битное двоичное значение). Двойное слово занимает 2 последовательных регистра данных. ■ Обозначаются как **D,** адресация в десятичной системе, т.е. D0, D1, D2…D4999
E, F (индексный регистр) (слово)	Словная память – используется как модификатор для обозначения регистра (слово/двойное слово) путем смещения адресов. Индексные регистры, не являющиеся модификаторами, могут использоваться, как регистры общего назначения. ■ Адресация, как E0 ~ E7 и F0 ~ F7.

1.4 Символы релейно-контактных (лестничных) логических диаграмм

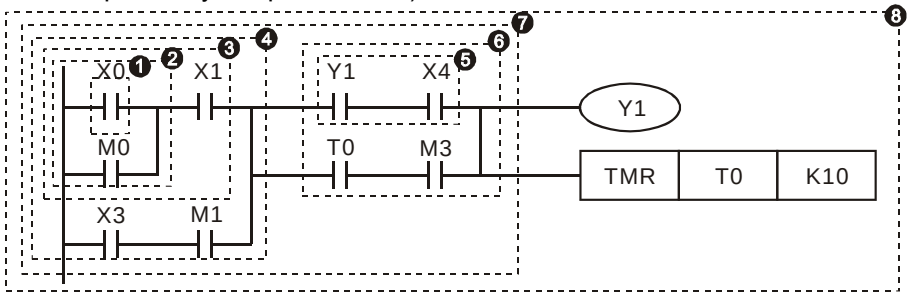
Элемент диаграммы	Описание	Команда	Доступное устройство (регистр, реле)
	Н/О контакт / А контакт	LD	X, Y, M, S, T, C
	Н/З контакт / В контакт	LDI	X, Y, M, S, T, C
	Н/О контакты последовательно	AND	X, Y, M, S, T, C
	Н/З контакты последовательно	ANI	X, Y, M, S, T, C
	Н/О контакты параллельно	OR	X, Y, M, S, T, C
	Н/З контакты параллельно	ORI	X, Y, M, S, T, C
	Триггер с опросом по переднему фронту	LDP	X, Y, M, S, T, C
	Триггер с опросом по заднему фронту	LDF	X, Y, M, S, T, C
	Триггер с опросом по переднему фронту последовательно	ANDP	X, Y, M, S, T, C
	Триггер с опросом по заднему фронту последовательно	ANDF	X, Y, M, S, T, C
	Триггер с опросом по переднему фронту параллельно	ORP	X, Y, M, S, T, C
	Триггер с опросом по заднему фронту параллельно	ORF	X, Y, M, S, T, C
	Блок последовательно	ANB	Нет
	Блок параллельно	ORB	Нет
	Схема из нескольких веток	MPS MRD MPP	Нет
	Выходная катушка	OUT	Y, M, S
	Шаговый контакт	STL	S
	Базовая или прикладная команда	-	Базовая команда или API
	Логическая инверсия	INV	Нет

1.4.1 Создание релейно-контактных (лестничных) программ для ПЛК

Оценка программы идет сначала слева направо, затем сверху вниз. Максимальное количество последовательных контактов в строке - 11. Если контактов больше 11, на выходе 11 контактов будет автоматически сгенерирован сигнал «0» и 12-й контакт будет перемещен на следующую строку. См. рис. ниже:



Во время оценки программы ПЛК сканирует строку слева направо, после чего переходит на следующую строку и так до выполнения команды END. Выходные катушки и базовые/прикладные команды располагаются в правой части диаграммы. На примере ниже показана диаграмма и порядок выполнения операций (цифры показывают порядок, операция – в рамке пунктирной линии).

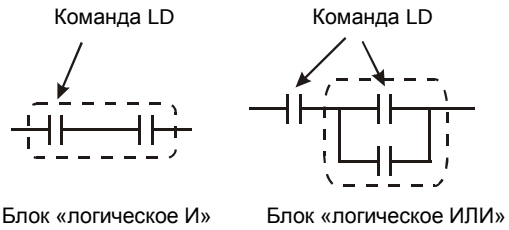


Пример порядка выполнения программы:

1	LD	X0
2	OR	M0
3	AND	X1
4	LD	X3
	AND	M1
	ORB	
5	LD	Y1
	AND	X4
6	LD	T0
	AND	M3
	ORB	
7	ANB	
8	OUT	Y1
	TMR	T0 K10

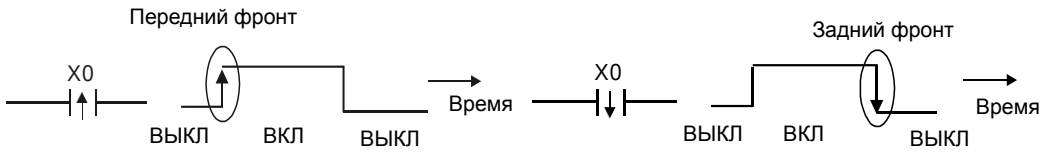
1.4.2 LD / LDI (Загрузка Н/О контакта / загрузка Н/З контакта)

LD или LDI запускают строку или блок



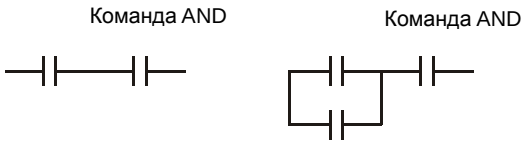
1.4.3 LDP / LDF (Загрузка триггера с опросом по переднему фронту / загрузка триггера с опросом по заднему фронту)

Подобно команде LD, команды LDP и LDF действуют только по переднему или заднему фронту, как показано на рис. ниже.



1.4.4 AND / ANI (Последовательное соединение Н/О контактов / последовательное соединение Н/З контактов)

Команда AND (ANI) последовательно соединяет Н/О (Н/З) контакты по отдельным контактам или блокам.

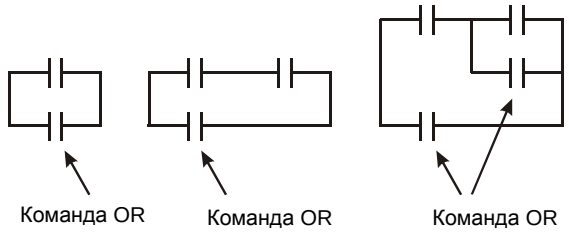


1.4.5 ANDP / ANDF (Последовательное соединение триггеров с опросом по переднему/ заднему фронту)

Аналогично команде AND, команда ANDP (ANDF) последовательно соединяет триггеры поштучно или блоками.

1.4.6 OR / ORI (Параллельное соединение Н/О контактов / параллельное соединение Н/З контактов)

Команда OR (ORI) параллельно соединяет Н/О (Н/З) контакты по отдельным контактам или блокам.

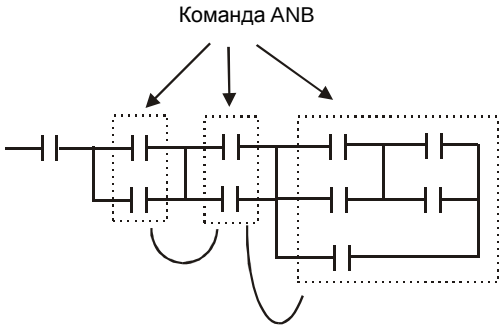


1.4.7 ORP / ORF (Параллельное соединение триггеров с опросом по переднему/ заднему фронту)

Аналогично команде OR, команды ORP (ORF) последовательно соединяет триггеры поштучно или блоками.

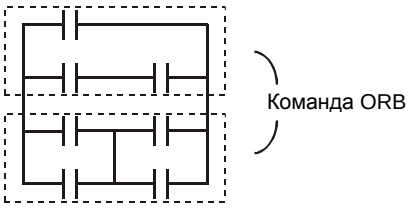
1.4.8 ANB (Последовательное соединение блоков)

Команда ANB последовательно соединяет блоки.



1.4.9 ORB (Параллельное соединение блоков)

Команда ORB параллельно соединяет блоки.



1.4.10 MPS / MRD / MPP (Узловые команды)

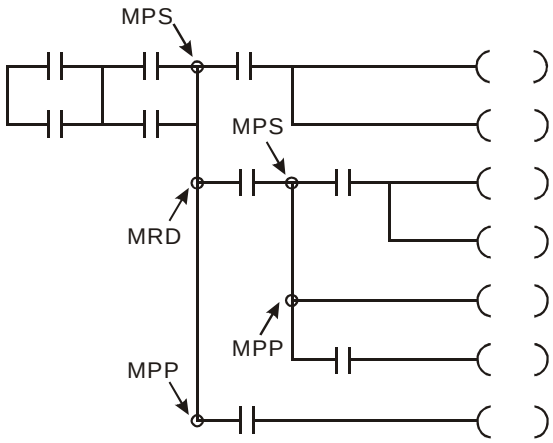
Эти команды определяют метод создания узлов выхода на несколько потоков, выходы на потоки осуществляются на основе текущей информации, сохраняемой командой MPS.

Узловые команды	Символ	Описание
MPS	┬	Стартовый узел. Сохраняет текущее значение оценки программы. Возможно макс. 8 MPS-MPP пар
MRD	├	Считывается сохраненный текущий результат предыдущего MPS
MPP	└	Конечный узел. Считывает и сбрасывает результат, сохраненный в предыдущем узле MPS

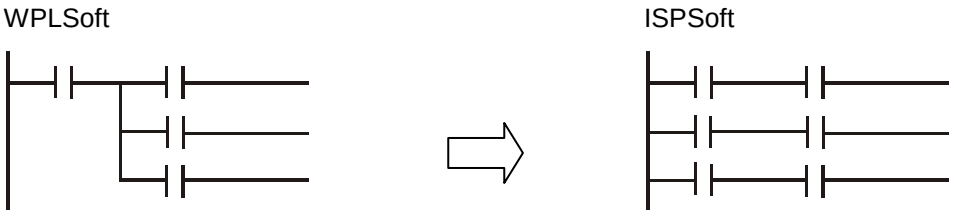
Примечание: При составлении лестничных диаграмм (релейно-контактных схем) с ПО WPLSoft, MPS, MRD и MPP автоматически добавляются в формат базовых/прикладных

команд. Однако иногда эти команды игнорируются ПО WPLSoft. Пользователь может вводить эти команды в программу при необходимости.

Узлы MPS, MRD и MPP:

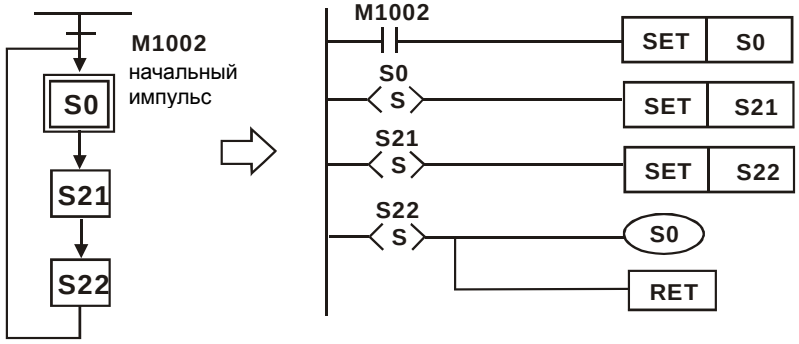


Примечание: Лестничные диаграммы, составленные в ISPSoft не поддерживают команды MPS, MRD and MPP. В этом случае, для получения результата, аналогичного действию узловых команд, пользователь должен подключать все ветви к левой шине. См. рис. ниже:



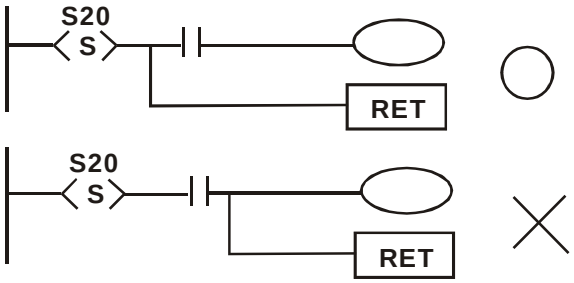
1.4.11 STL (Шаговое лестничное программирование)

STL программирование применяет схему «шаг от точки к точке», т.е. S0 S21, S22, что позволяет пользователю в процессе программирования создавать четкие и понятные блок-схемы. Программа приступает к выполнению следующего шага только после полной обработки предыдущего, что образует последовательный процесс управления, схожий со схемой SFC (Последовательная функциональная схема). Последовательность STL может быть преобразована в лестничную диаграмму ПЛК как показано ниже:



1.4.12 RET (Возврат)

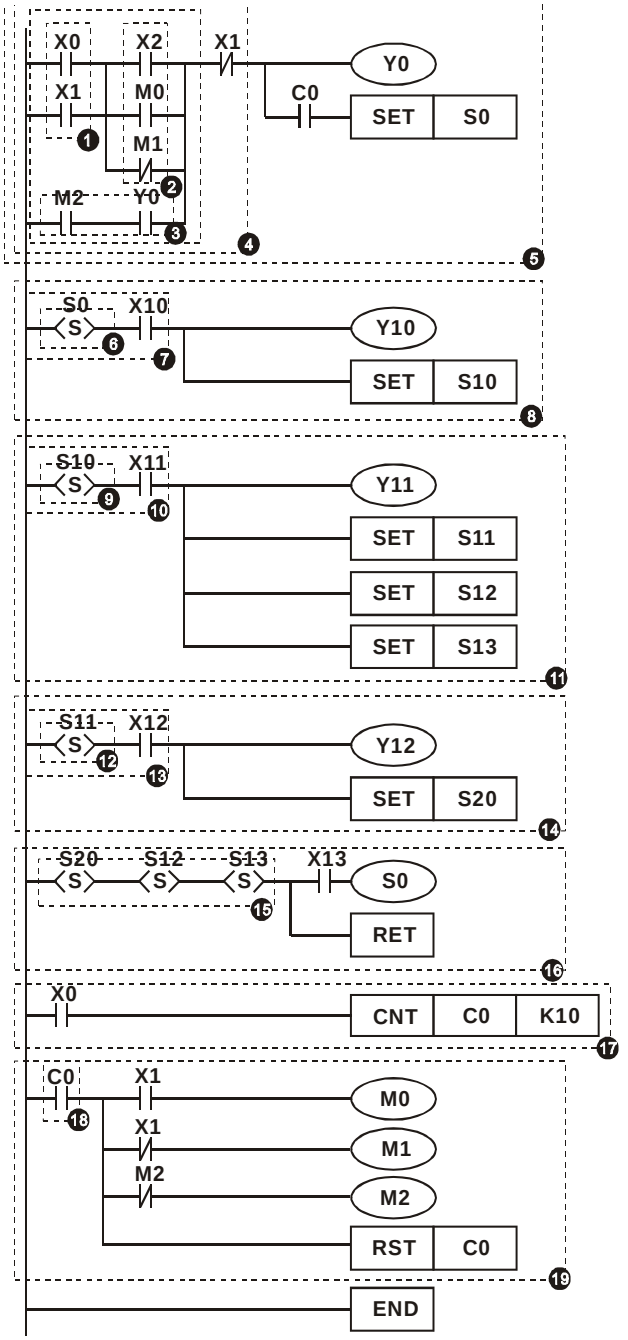
Команда RET помещается в конце последовательного процесса управления для указания завершения потока STL.



Примечание: Всегда подключайте RET как показано на рисунке во избежание ошибки.

1.5 Преобразование лестничной диаграммы в список команд

Лестничная диаграмма



Команда

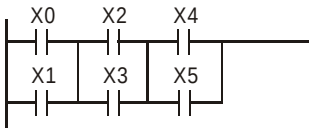
```
LD    X0      ❶ Блок ИЛИ
OR    X1
LD    X2      ❷ Блок ИЛИ
OR    M0
ORI   M1
ANB           Блоки последовательно
LD    M2      ❸ Блок И
AND   Y0
ORB           Блоки параллельно
AN I  X1      ❹ ANI
OUT   Y0      Несколько выходов
AND   C0
SET   S0      Старт лестничной диаграммы ❻
STL   S0      Состояние S0 определяется X10 ❼
LD    X10     ❽
OUT   Y10     Чтение состояния выхода Y10 и шаговый переход
SET   S10
STL   S10     ❾ Чтение состояния S10
LD    X11     ❿ S10 работает с X11
OUT   Y11     ⓫
SET   S11     Чтение состояния выхода Y11 и шаговый переход
SET   S12
SET   S13
STL   S11     ⓬ Чтение состояния S11
LD    X12     ⓭ S11 работает с X12
OUT   Y12     ⓮
SET   S20     Чтение состояния выхода Y12 и шаговый переход
STL   S20
STL   S12     ⓯ Схождение нескольких выходов
STL   S13
LD    X13     ⓰ Конец лестничной диаграммы
OUT   S0      Чтение состояния выхода Y13 и шаговый переход
RET           Возврат
LD    X0      ⓱
CNT   C0 K10
LD    C0      ⓲ Чтение C0
MPS           ⓳
AND   X1      Несколько выходов
OUT   M0
MRD
AN I  X1
OUT   M1
MPP
AN I  M2
OUT   M2
RST   C0
END           Конец программы
```

Выход определяется состоянием ❹ ❺

1.6 Различные варианты списков команд (мнемокодов).

Как правило, программирование лестничных диаграмм осуществляется по принципу “сверху вниз и слева направо”. Однако существуют методы программирования, не следующие этому принципу, но позволяющие получить тот же и даже лучший результат управления. Рассмотрим ряд примеров преобразования лестничной диаграммы в список команд (мнемокод):

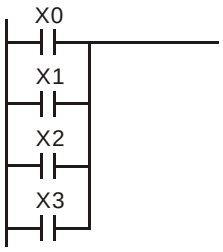
Пример 1:



Улучшенный метод		Обычный метод	
LD	X0	LD	X0
OR	X1	OR	X1
LD	X2	LD	X2
OR	X3	OR	X3
ANB		LD	X4
LD	X4	OR	X5
OR	X5	ANB	
ANB		ANB	

Оба мнемокода могут быть получено из одной лестничной диаграммы. Разница в том, что в обычном методе невозможно объединить более 8 блоков (более 8 раз использовать команду ANB), превышение этого числа приведет к ошибке выполнения программы. В улучшенном методе такого ограничения нет. Таким образом, улучшенный метод является более надежным и удобным с точки зрения пользователя.

Пример 2:

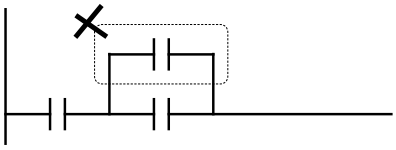
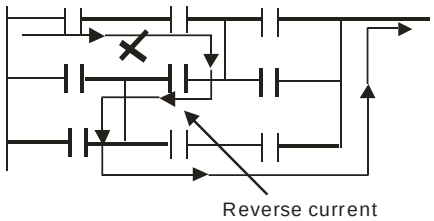
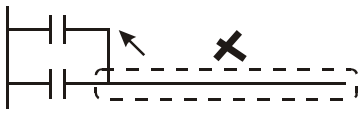
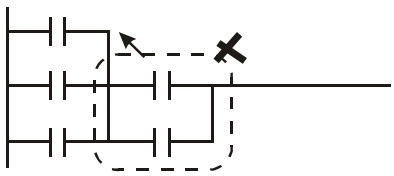


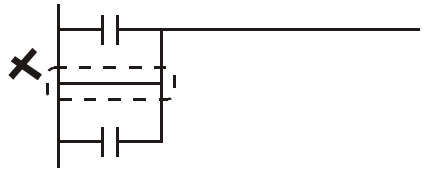
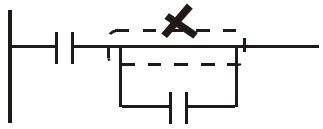
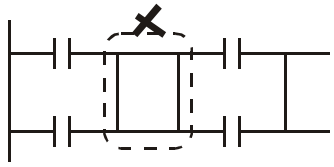
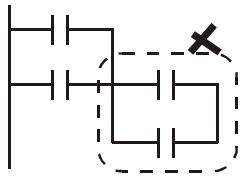
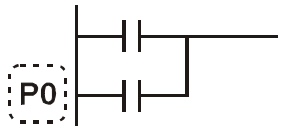
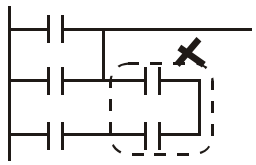
Лучший метод		Худший метод	
LD	X0	LD	X0
OR	X1	LD	X1
OR	X2	LD	X2
OR	X3	LD	X3
		ORB	
		ORB	
		ORB	

Лучший метод имеет меньший программный код и, как следствие, использует меньший ресурс оперативной памяти ПЛК, что снижает число ошибок и повышает быстродействие программы.

Распространенные ошибки при программировании

ПЛК обрабатывает программу слева направо и сверху вниз. При редактировании программ, пользователь должен это учитывать, иначе WPLSoft обнаружит ошибку. Наиболее распространенные ошибки:

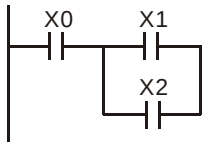
	Нельзя подключать параллельный контакт (операция ИЛИ) сверху.
	Не допускается реверсивное движение сигнала («тока»).
	Выходная линия должна подключаться сверху.
	Комбинация блоков: параллельный блок (операция ИЛИ) должен располагаться сверху.

	Нельзя подключить параллельно пустую линию.
	
	Нет объектов между блоками.
	Блоки должны располагаться на одном уровне.
	Указатель P0 должен располагаться напротив верхнего устройства.
	Не допускается реверсивное движение сигнала («тока»).

1.7 Оптимизация лестничных диаграмм.

Пример 1:

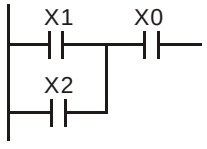
Подключение блока спереди позволит не применять команду ANB и улучшить скорость обработки



Мнемокод

LD	X0
LD	X1
OR	X2
ANB	

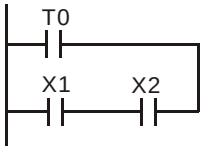
⇩



Мнемокод

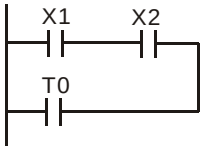
LD	X1
OR	X2
AND	X0

Пример 2:
Если поставить блок выше параллельного контакта, можно избежать применения ORB



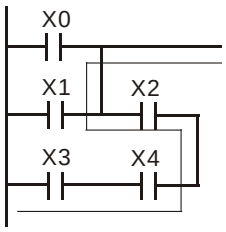

```
Мнемокод
LD    T0
LD    X1
AND   X2
ORB
```

⇩




```
Мнемокод
LD    X1
AND   X2
OR    T0
```

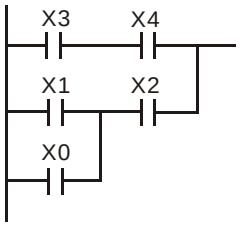
Пример 3:
В схеме с реверсивным «током» замена верхней и нижней линий позволяет избежать ошибки и нарушения принципов обработки программы ПЛК.



(a)

```
Мнемокод
LD    X0
OR    X1
AND   X2
LD    X3
AND   X4
ORB
```

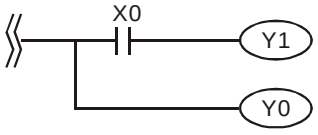
⇩



(b)

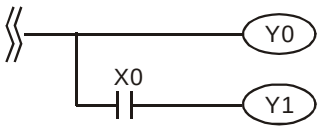
```
Мнемокод
LD    X3
AND   X4
LD    X1
OR    X0
AND   X2
ORB
```

Пример 4:
Для нескольких выходов, подключение верхней линии без дополнительного входа (выходы меняются местами) позволит избежать применения команд MPS и MPP.



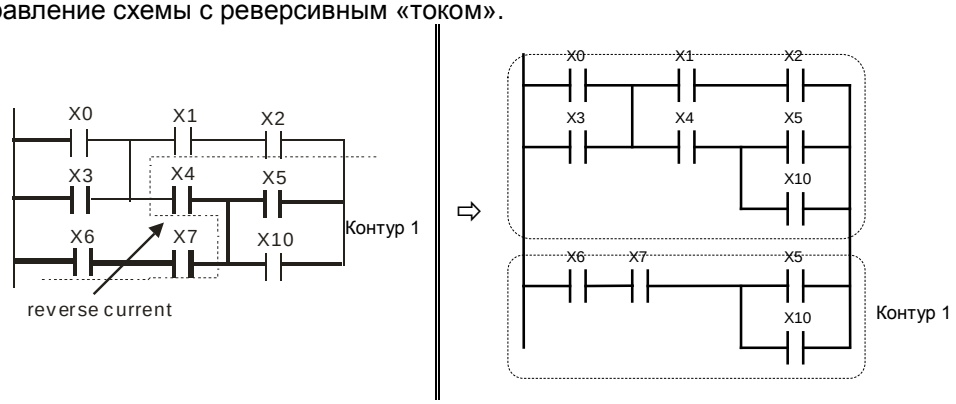

```
Мнемокод
MPS
AND   X0
OUT   Y1
MPP
OUT   Y0
```

⇩

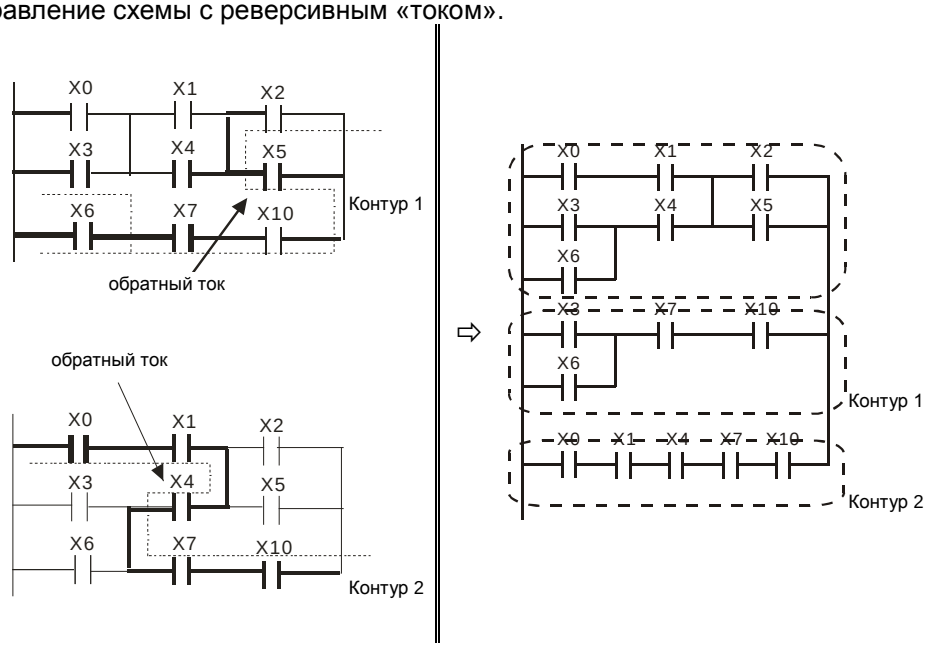



```
Мнемокод
OUT   Y0
AND   X0
OUT   Y1
```

Пример 5:
Исправление схемы с реверсивным «током».



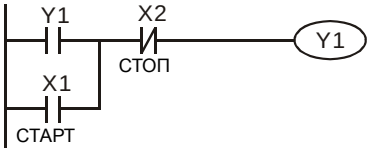
Пример 6:
Исправление схемы с реверсивным «током».



1.8 Примеры написания программ

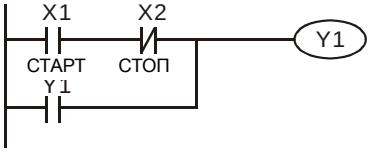
Пример 1 – Самоблокировка выхода с приоритетом STOP

Когда X1 (START) = ON и X2 (STOP) = OFF, Y1 = ON. Если X2 =ON, Y1 = OFF. В примере STOP имеет приоритет над START.



Пример 2 - Самоблокировка выхода с приоритетом START

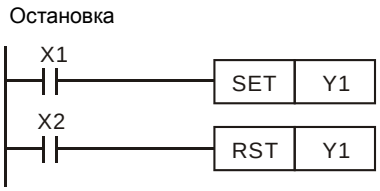
Когда X1 (START) = ON и X2 (STOP) = OFF, Y1 = ON и фиксируется. Если X2 = ON, Y1 = ON. В примере START имеет приоритет над STOP.



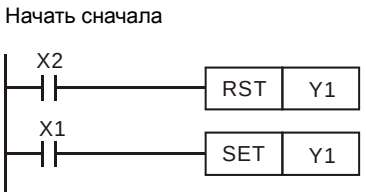
Пример 3 - Самоблокировка выхода с применением команд SET и RST

На диаграмме представлены две противоположных схемы, использующие команды RST и SET.

ПЛК обрабатывает программу сверху вниз, поэтому приоритет имеет команда, расположенная ниже, определяя

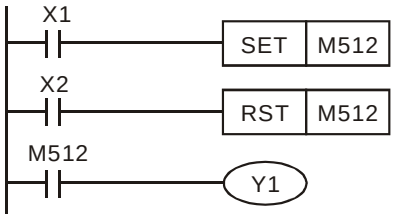


состояние выхода Y1. Следовательно, если X1 и X2 = ON, в верхней схеме Y1 = 0, в нижней Y1 = 1.

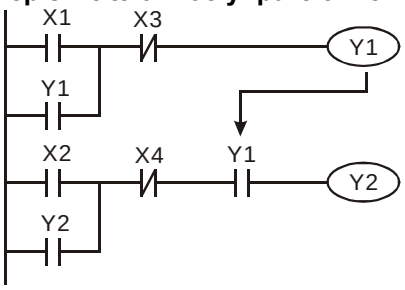


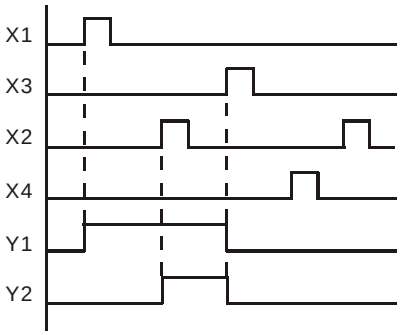
Пример 4 – Энергонезависимая самоблокировка выхода

Вспомогательное реле M512 – реле с блокировкой (энергонезависимое). Когда X1 = ON, Y1 сохранит свое состояние до отключения питания и возобновит его после включения.



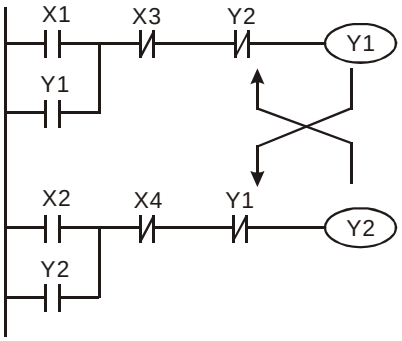
Пример 5 – Условное управление

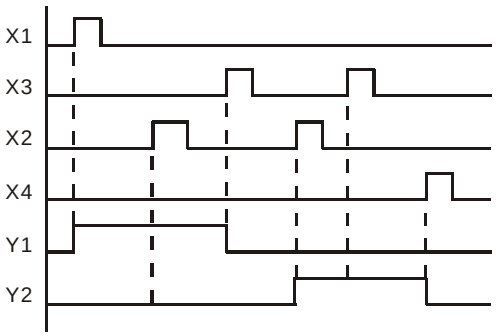




Н/О контакт Y1 подключен к цепи выхода Y2, Y1 является непременным условием включения Y2, т.е., чтобы Y2 = ON, Y1 = ON

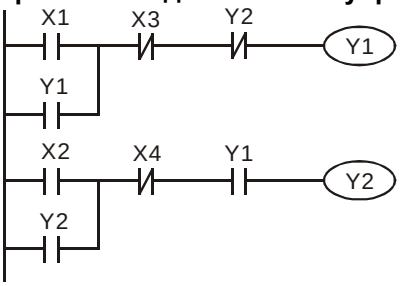
Пример 6- Управление с взаимоблокировкой





Н/З контакт Y1 соединен с цепью выхода Y2 и Н/З контакт Y2 соединен с цепью выхода Y1. Если Y1 = ON, Y2 = OFF и наоборот. Это предотвращает одновременное включение обоих выходов. При X1 и X2 = ON включится только Y1.

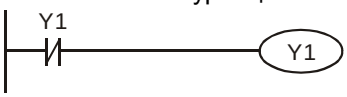
Пример 7 – Последовательное управление

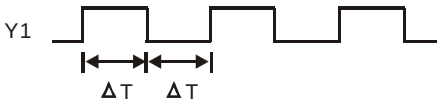


Соединим Н/О контакт Y2 с цепью выхода Y1 и контакт Y1 с цепью выхода Y2. Включение Y1 является условием включения выхода Y2. Кроме того, Y1 будет отключен при включении Y2, что формирует последовательную схему управления.

Пример 8 – Колебательный контур

Колебательный контур с циклом $\Delta T + \Delta T$

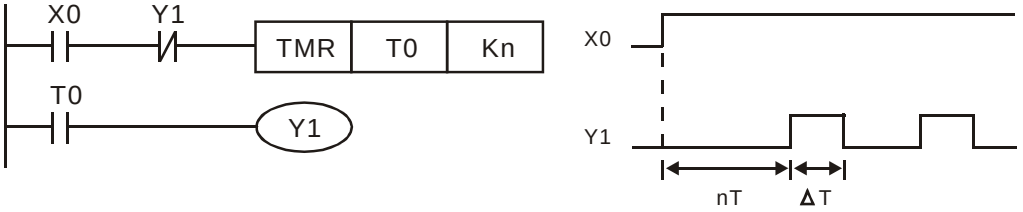




При первом сканировании, Y1 включается. При втором, Y1 выключится по причине обратного состояния контакта Y1. Состояние выхода Y1, меняющееся в каждом цикле сканирования, формирует колебательный контур с периодом $\Delta T(ON) + \Delta T(OFF)$

Пример 9 – Колебательный контур с таймером

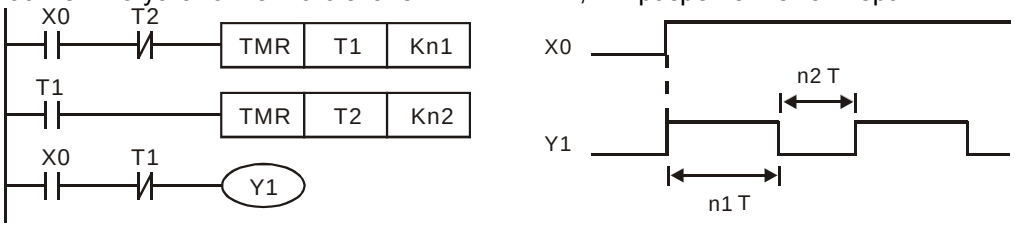
Период колебательного контура nT+ΔT



Когда X0 = ON, T0 запускает отсчет времени (nT). Как только заданное время выйдет, контакт T0 = ON включает Y1(ΔT). В следующем цикле сканирования, таймер T0 сбрасывается в связи с обратным статусом контакта Y1. Контакт T0 сбрасывается и Y1 = OFF. В следующем цикле, T0 снова запускает отсчет времени. Процесс формирует колебательный контур с периодом nT+ΔT.

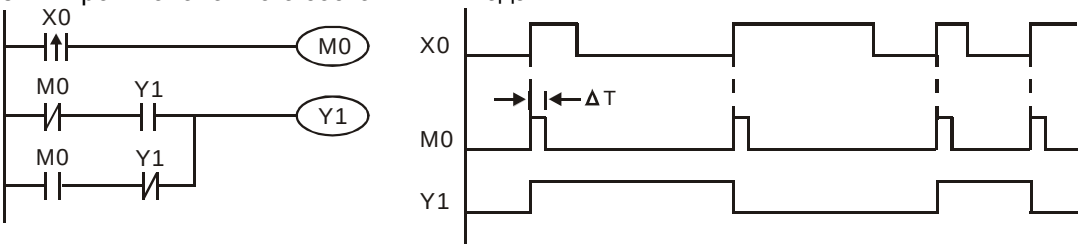
Пример 10 – Мигающий контур

Лестничная диаграмма использует два таймера для формирования колебательного контура, запускающего мигание звукового индикатора или срабатывание звуковой сигнализации. n1 и n2 относительно установленного значения в T1 и T2, T – разрешение таймера.



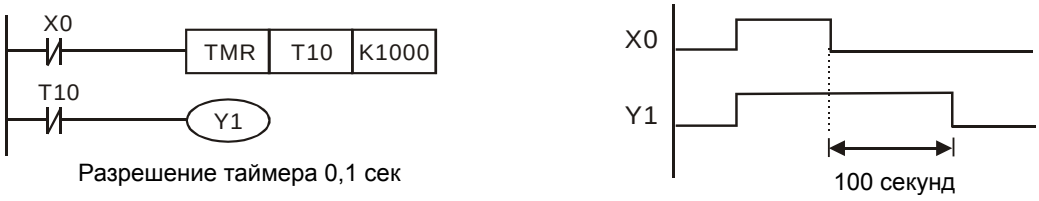
Пример 11 – Триггерный контур

На данной диаграмме контакт по переднему фронту X0 формирует триггерный импульс для получения противоположного состояния выхода Y1.



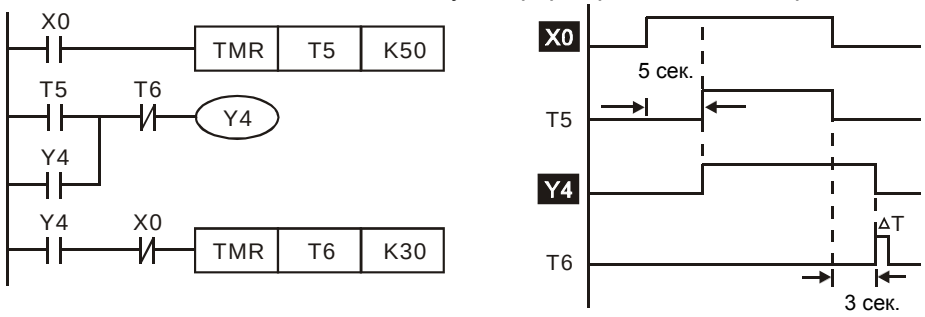
Пример 12 – Контур с задержкой на выключение

Если X0 включен, таймер T10 не запускается, хотя катушка Y1 включена. Когда X0 выключен, T10 включается. После 100 секунд (K1000 × 0.1 сек* = 100 сек), Н/З контакт T10 включается для отключения Y1. Таким образом, контур отключается с задержкой 100 сек от включения управляющего сигнала. * - дискретность таймера

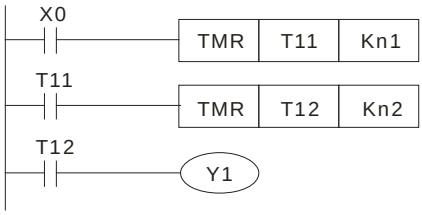


Пример 13 – Выходной сигнал с задержкой

Задержка выходного сигнала формируется 2-мя таймерами. Независимо от того, включен или выключен X0, сигнал на выходе Y4 будет формироваться с задержкой.

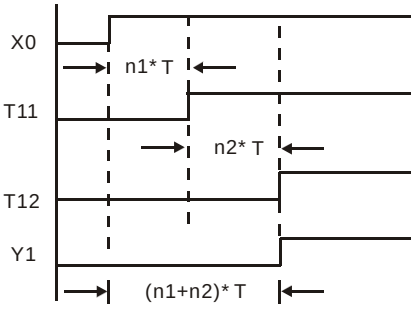


Пример 14 – Таймер с 2-х ступенчатой задержкой на включение

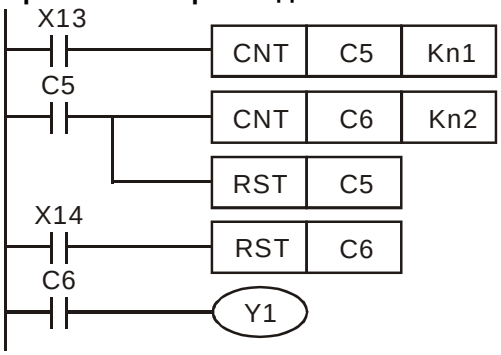


Таймеры = T11, T12
Разрешение таймера: T

Полное время задержки: (n1+n2)*T. T определяется разрешением таймера.



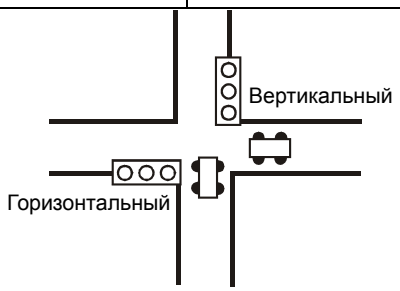
Пример 15 – Расширение диапазона счетчика



Диапазон 16-битного счетчика составляет 0 ~ 32 767. Применение второго счетчика позволит увеличить диапазон счета до значения n1*n2. Когда значение счетчика C6 достигает n2, импульсы от X13 будут отсчитываться до n1*n2.

Пример 16 – Управление светофором (с применением пошагового управления)
Управление светофором

	Красный	Желтый	Зеленый	Зеленый мигает
Вертикальный	Y0	Y1	Y2	Y2
Горизонтальный	Y20	Y21	Y22	Y22
Время свечения	35 сек	5 сек	25 сек	5 сек



Временная диаграмма:

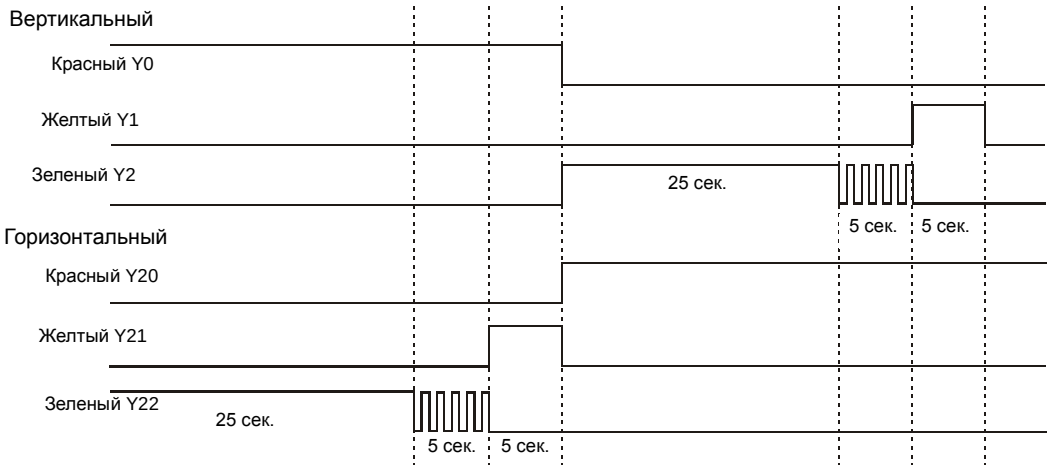
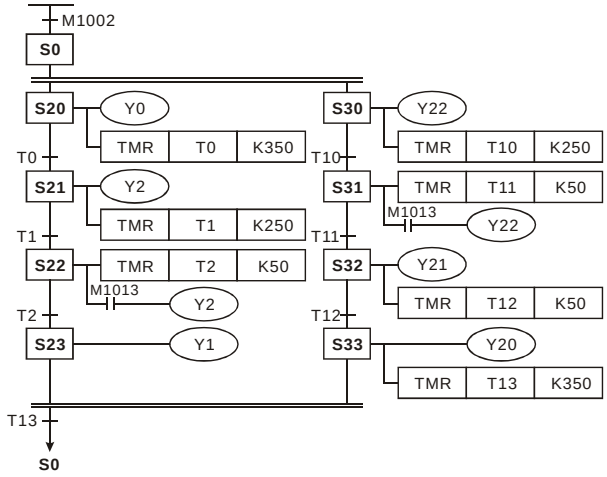
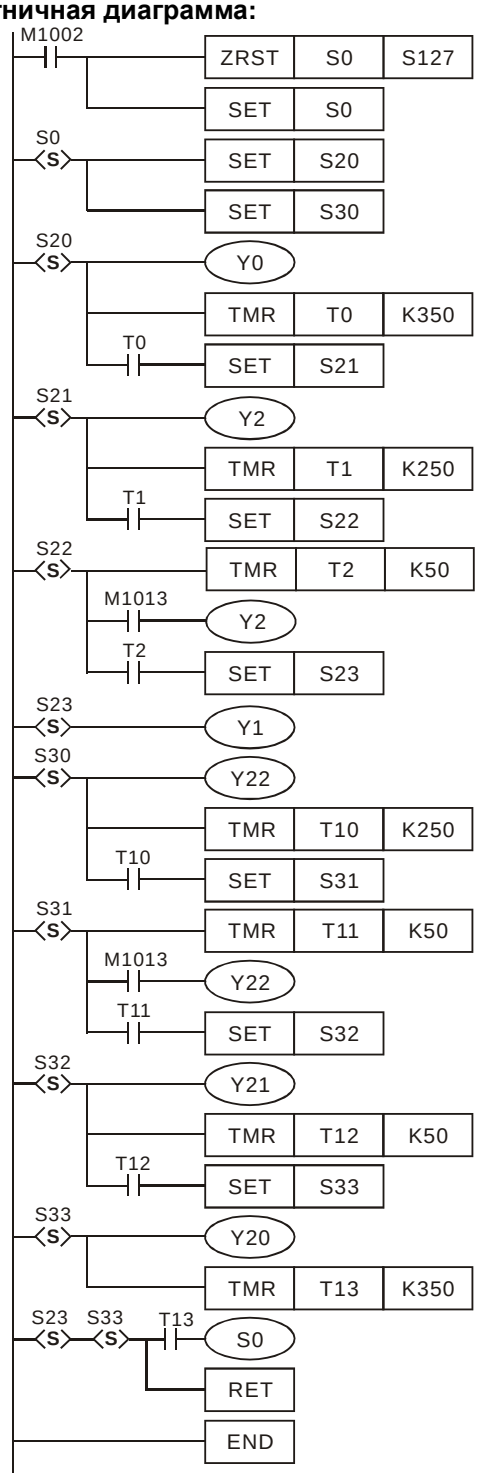


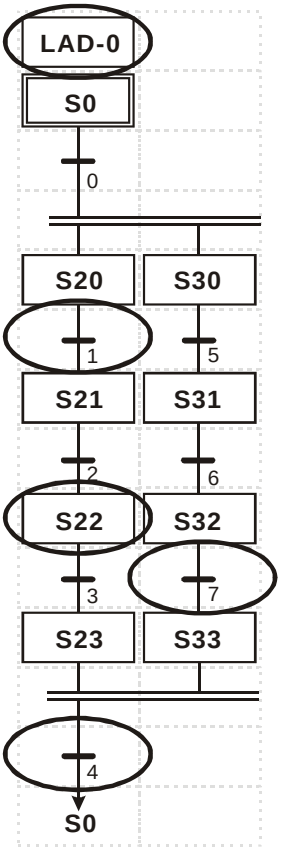
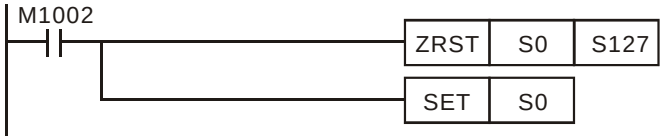
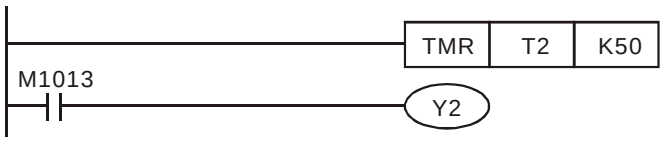
Схема SFC:



Лестничная диаграмма:



Программирование WPLSoft (режим SFC)

Логика SFC	Внутренняя релейно-контактная логика
	**LAD-0**  **Условие передачи 1**  **S22**  **Условие передачи 4**  **Условие передачи 7** 

Операнды, применяемые в программировании

Серия DVP – программируемые логические контроллеры (ПЛК), имеющие 10–256 входов/выходов (SS2/SA2/SX2/SE: 512 входов/выходов). ПЛК контролирует входы и изменяет состояние выходов, управляя ими при помощи пользовательских программ. Программы обеспечивают функции булевой логики, счета, времени, сложные математические операции и связь с другими устройствами, участвующими в процессе автоматического управления.

2.1 ES2/EX2 Карта параметров

<table>
<tr><th colspan="7">Описание</th></tr>
<tr><td colspan="4">Метод управления</td><td colspan="2">Загруженная программа, циклическое сканирование</td><td></td></tr>
<tr><td colspan="4">Метод обработки входов/выходов</td><td colspan="2">Метод пакетной обработки (при выполнении команды END)</td><td></td></tr>
<tr><td colspan="4">Скорость выполнения</td><td colspan="2">Команды LD – 0.54 мкс, команды MOV – 3.4 мкс</td><td></td></tr>
<tr><td colspan="4">Язык программирования</td><td colspan="2">Мнемокод+ лестничная диаграмма + SFC</td><td></td></tr>
<tr><td colspan="4">Размер программы</td><td colspan="2">До 15872 шагов</td><td></td></tr>
<tr><td rowspan="14">Битные контакты</td><td>X</td><td colspan="2">Внешние входы</td><td colspan="2">X0~X377, восьмеричные, макс. 256 точек, (*4)</td><td rowspan="2">Всего 256+16 Вх/Вых</td></tr>
<tr><td>Y</td><td colspan="2">Внешние выходы</td><td colspan="2">X0~X377, восьмеричные, макс. 256 точек, (*4)</td></tr>
<tr><td rowspan="3">M</td><td rowspan="3">Реле</td><td>Общие</td><td colspan="2">M0~M511, 512 точек, (*1)
M768~M999, 232 точек, (*1)
M2000~M2047, 48 точек, (*1)</td><td rowspan="3">Всего 4096 точек</td></tr>
<tr><td>Энергонезависимые</td><td colspan="2">M512~M767, 256 точек, (*2)
M2048~M4095, 2048 точек, (*2)</td></tr>
<tr><td>Специальные</td><td colspan="2">M1000~M1999, 1000 точек, вкл. энергонез.</td></tr>
<tr><td rowspan="5">T</td><td rowspan="5">Таймеры</td><td rowspan="3">100 мс (M1028 включен, T64~T126: 10 мс)</td><td colspan="2">T0~T126, 127 точек, (*1)
T128~T183, 56 точек, (*1)</td><td rowspan="5">Всего 256 точек</td></tr>
<tr><td colspan="2">T184~T199 для подпрограмм, 16 точек, (*1)</td></tr>
<tr><td colspan="2">T250~T255 (аккумулятивные), 6 точек (*1)</td></tr>
<tr><td rowspan="2">10 мс (M1038 включен, T200~T245: 1 мс)</td><td colspan="2">T200~T239, 40 точек, (*1)</td></tr>
<tr><td colspan="2">T240~T245 (аккумулятивные), 6 точек, (*1)</td></tr>
<tr><td rowspan="4">C</td><td rowspan="4">Счетчики</td><td>1 мс</td><td colspan="2">T127, 1 точка, (*1)
T246~T249 (аккумулятивные), 4 точки, (*1)</td><td>Всего 232 точек</td></tr>
<tr><td rowspan="2">16-бит инкрементный</td><td colspan="2">C0~C111, 112 точек, (*1)
C128~C199,72 точек, (*1)</td><td rowspan="4">Всего 232 точек</td></tr>
<tr><td colspan="2">C112~C127,16 точек, (*2)</td></tr>
<tr><td rowspan="2">32-бит инкр./декрем.</td><td colspan="2">C200~C223, 24 точек, (*1)</td></tr>
<tr><td colspan="2">C224~C231, 8 точек, (*2)</td></tr>
<tr><td rowspan="5">32-бит высокоскоростной инкр./ декрем.</td><td rowspan="2">программный</td><td colspan="2">C235~C242, 1 фаза 1 вход, 8 точек, (*2)
C232~C234, 2 фазы 2 входа, 3 точки, (*2)</td><td rowspan="5">Всего 23 точки</td></tr>
<tr><td rowspan="3">аппаратный</td><td colspan="2">C243~C244, 1 фаза 1 вход, 2 точки, (*2)</td></tr>
<tr><td colspan="2">C245~C250, 1 фаза 2 входа, 6 точек, (*2)</td></tr>
<tr><td colspan="2">C251~C254 2 фазы 2 входа, 4 точки, (*2)</td></tr>
</table>

Битные контакты	S	Шаговые реле	Начальные	S0~S9, 10 точек, (*2)	Всего 1024 точки
			Возврат к нулевой точке	S10~S19, 10 точек (используется с инструкцией IST), (*2)	
			Энергонезависимые	S20~S127, 108 точек, (*2)	
			Общие	S128~S911, 784 точки, (*1)	
			Сигнальные	S912~S1023, 112 точек, (*2)	
Словные регистры	T	Текущее значение таймера		T0~T255, 256 слов	
	C	Текущее значение счетчика		C0~C199, 16-бит счетчик, 200 слов	
				C200~C254, 32-бит счетчик, 55 слов	
	D	Регистры данных	Общие	D0~D407, 408 слов, (*1) D600~D999, 400 слов, (*1) D3920~D9999, 6080 слов, (*1)	Всего 10000 точек
			Энергонезависимые	D408~D599, 192 слова, (*2) D2000~D3919, 1920 слов, (*2)	
			Специальные	D1000~D1999, 1000 слов, some are latched	
			Для аналоговых модулей ввода/вывода	D9900~D9999，100 слов，(*1), (*5)	
			Индексные	E0~E7, F0~F7, 16 слов, (*1)	
Указатели	N	Управление обратной связью		N0~N7, 8 точек	
	P	Указатели		P0~P255, 256 точек	
	I	Прерывания	Внешние	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 точек (01: триггер с опросом по переднему фронту ┘‾, 00: триггер с опросом по заднему фронту ‾└)	
			С таймером	I602~I699, I702~I799, 2 точки (разрешение = 1 мс)	
			С высокоскоростным счетчиком	I010, I020, I030, I040, I050, I060, I070, I080,8 точек	
			Коммуникационные	I140(COM1), I150(COM2), I160(COM3), 3 точки, (*3)	
Константы	K	Десятичные		K-32,768 ~ K32,767 (16-bit operation), K-2,147,483,648 ~ K2,147,483,647 (32-bit operation)	
	H	Шестнадцатеричные		H0000 ~ HFFFF (16-битные операции), H00000000 ~HFFFFFFFF (32-битные операции)	
Последовательные порты				COM1: встроенный RS-232 (Master/Slave) COM2: встроенный RS-485 (Master/Slave) COM3: встроенный RS-485 (Master/Slave) COM1 - порт для программирования (в основном).	
Часы реального времени				Год, месяц, день, неделя, часы, минуты, секунды	
Специальные входные/выходные модули				До 8 специальных модулей	

Заметки:

1. Энергозависимая область не может быть изменена.
2. Энергонезависимая область не может быть изменена.
3. COM1: встроенный порт RS232. COM2: встроенный порт RS485. COM3: встроенный порт RS485.
4. Когда число входов (X) расширено до 256 точек, возможно только 16 выходов (Y), и наоборот, если число выходов (Y) расширено до 256 точек, возможно только 16 входов (X).

5. Эта область применяется только, когда MPU ES2/EX2 используется со специальными модулями. Каждый специальный модуль занимает 10 точек.

2.2 SS2 Карта параметров

<table>
<tr><th colspan="7">Описание</th></tr>
<tr><td colspan="3">Метод управления</td><td colspan="4">Загруженная программа, циклическое сканирование</td></tr>
<tr><td colspan="3">Метод обработки входов/выходов</td><td colspan="4">Метод пакетной обработки (при выполнении команды END)</td></tr>
<tr><td colspan="3">Скорость выполнения</td><td colspan="4">Команды LD – 0.54 мкс, команды MOV – 3.4 мкс</td></tr>
<tr><td colspan="3">Язык программирования</td><td colspan="4">Мнемокод+ лестничная диаграмма + SFC</td></tr>
<tr><td colspan="3">Размер программы</td><td colspan="4">До 7920 шагов</td></tr>
<tr><td rowspan="19">Битные контакты</td><td>X</td><td colspan="2">Внешние входы</td><td colspan="2">X0~X377, восьмеричные, макс. 256 точек</td><td rowspan="2">Всего 480+14 Вх/Вых (*4)</td></tr>
<tr><td>Y</td><td colspan="2">Внешние выходы</td><td colspan="2">Y0~Y377, восьмеричные, макс. 256 точек</td></tr>
<tr><td rowspan="3">M</td><td rowspan="3">Реле</td><td>Общие</td><td colspan="2">M0~M511, 512 точек, (*1)
M768~M999, 232 точки, (*1)
M2000~M2047, 48 точек, (*1)</td><td rowspan="3">Всего 4096 точек</td></tr>
<tr><td>Энергонезависимые</td><td colspan="2">M512~M767, 256 точек, (*2)
M2048~M4095, 2048 точек, (*2)</td></tr>
<tr><td>Специальные</td><td colspan="2">M1000~M1999, 1000 точек, вкл. энергонез.</td></tr>
<tr><td rowspan="5">T</td><td rowspan="5">Таймеры</td><td rowspan="3">100 мс
(M1028 включен,
T64~T126:
10 мс)</td><td colspan="2">T0~T126, 127 точек, (*1)
T128~T183, 56 точек, (*1)</td><td rowspan="5">Всего 256 точек</td></tr>
<tr><td colspan="2">T184~T199 для подпрограмм, 16 точек, (*1)</td></tr>
<tr><td colspan="2">T250~T255 (аккумулятивные),
6 точек (*1)</td></tr>
<tr><td>10 мс
(M1038 включен,
T200~T245: 1 мс)</td><td colspan="2">T200~T239, 40 точек, (*1)
T240~T245 (аккумулятивные),
6 точек, (*1)</td></tr>
<tr><td>1 мс</td><td colspan="2">T127, 1 точка, (*1)
T246~T249 (аккумулятивные), 4 точки, (*1)</td></tr>
<tr><td rowspan="6">C</td><td rowspan="6">Счетчики</td><td rowspan="2">16-бит инкрементные</td><td colspan="2">C0~C111, 112 точек, (*1)
C128~C199, 72 точки, (*1)</td><td rowspan="4">Всего 233 точек</td></tr>
<tr><td colspan="2">C112~C127, 16 точек, (*2)</td></tr>
<tr><td rowspan="2">32-бит инкрем./ декрем.</td><td colspan="2">C200~C223, 24 точки, (*1)</td></tr>
<tr><td colspan="2">C224~C232, 9 точек, (*2)</td></tr>
<tr><td rowspan="4">32-бит высоко-скорост-ной инкр./ декрем.</td><td>Про-граммный</td><td>C235~C242, 1 фаза 1 вход, 8 точек, (*2)</td><td rowspan="4">Всего 22 точки</td></tr>
<tr><td rowspan="3">Аппарат-ный</td><td>C233~C234, 2 фазы 2 входа, 2 точки, (*2)</td></tr>
<tr><td>C243~C244, 1 фаза 1 вход, 2 точки, (*2)</td></tr>
<tr><td>C245~C250, 1 фаза 2 входа, 6 точек, (*2)
C251~C254 2 фазы 2 входа, 4 точки, (*2)</td></tr>
<tr><td rowspan="5">S</td><td rowspan="5">Шаговые</td><td colspan="2">Начальные</td><td>S0~S9, 10 точек, (*2)</td><td rowspan="5">Всего 1024 точки</td></tr>
<tr><td colspan="2">Возврат к нулевой точке</td><td>S10~S19, 10 точек (с использованием команды IST), (*2)</td></tr>
<tr><td colspan="2">Энергонезависимые</td><td>S20~S127, 108 точек, (*2)</td></tr>
<tr><td colspan="2">Общие</td><td>S128~S911, 784 точки, (*1)</td></tr>
<tr><td colspan="2">Сигнальные</td><td>S912~S1023, 112 точек, (*2)</td></tr>
</table>

Словные регистры	T	Текущее значение таймера		T0~T255, 256 слов	
	C	Текущее значение счетчика		C0~C199, 16-битный счетчик, 200 слов	
				C200~C254, 32- битный счетчик, 55 слов	
	D	Регистры данных	Общие	D0~D407, 408 слов, (*1) D600~D999, 400 слов, (*1) D3920~D4999, 1080 слов, (*1)	Всего 5016 точек
			Энергонезависимые	D408~D599, 192 слова, (*2) D2000~D3919, 1920 слов, (*2)	
			Специальные	D1000~D1999, 1000 слов, вкл.энергонез.	
			Индексные	E0~E7, F0~F7, 16 слов, (*1)	
Указатели	N	Управление обратной связью		N0~N7, 8 точек	
	P	Указатель		P0~P255, 256 точек	
	I	Прерывания	Внешние	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 точек (01: триггер с опросом по переднему фронту ⌐, 00: триггер с опросом по заднему фронту ⌙)	
			С таймером	I602~I699, I702~I799, 2 точки (разрешение = 1 мс)	
			С высокоскоростным счетчиком	I010, I020, I030, I040, I050, I060, I070, I080, 8 точек	
			Коммуникационные	I140(COM1), I150(COM2), 2 точки, (*3)	
Константы	K	Десятичные		K-32 768 ~ K32 767 (16-битные команды), K-2 147 483 648 ~ K2 147 483 647 (32-битные команды)	
	H	Шестнадцатеричные		H0000 ~ HFFFF (16-битные команды), H00000000 ~HFFFFFFFF (32-битные команды)	
Последовательные порты				COM1: встроенный RS-232 (Master/Slave) COM2: встроенный RS-485 (Master/Slave) COM1 порт для программирования (в основном).	
Часы реального времени				Год, месяц, день, неделя, часы, минуты, секунды	
Специальные входные/выходные модули				До 8 специальных модулей	

Заметки:

1. Энергозависимая область не может быть изменена.
2. Энергонезависимая область не может быть изменена.
3. COM1: встроенный порт RS232. COM2: встроенный порт RS485.
4. SS2 MPU занимает 16 входов (X0~X17) и 16 выходов (Y0~Y17).

2.3 SA2/SX2 Карта параметров

<table>
<tr><th colspan="6">Описание</th></tr>
<tr><td colspan="3">Метод управления</td><td colspan="3">Загруженная программа, циклическое сканирование</td></tr>
<tr><td colspan="3">Метод обработки входов/выходов</td><td colspan="3">Метод пакетной обработки (при выполнении команды END)</td></tr>
<tr><td colspan="3">Скорость выполнения</td><td colspan="3">Команды LD – 0.54 мкс, команды MOV – 3.4 мкс</td></tr>
<tr><td colspan="3">Язык программирования</td><td colspan="3">Мнемокод+ лестничная диаграмма + SFC</td></tr>
<tr><td colspan="3">Размер программы</td><td colspan="3">До 15872 шагов</td></tr>
<tr><td rowspan="19">Битные контакты</td><td>X</td><td colspan="2">Внешние входы</td><td>X0~X377, восьмеричные, макс. 256 точек</td><td rowspan="2">Всего: 480+14 Вх/Вых(*4)</td></tr>
<tr><td>Y</td><td colspan="2">Внешние выходы</td><td>Y0~Y377, восьмеричные, макс. 256 точек</td></tr>
<tr><td rowspan="3">M</td><td rowspan="3">Реле</td><td>Общие</td><td>M0~M511, 512 точек, (*1)
M768~M999, 232 точки, (*1)
M2000~M2047, 48 точек, (*1)</td><td rowspan="3">Всего 4096 точек</td></tr>
<tr><td>Энергонезависимые</td><td>M512~M767, 256 точек, (*2)
M2048~M4095, 2048 точек, (*2)</td></tr>
<tr><td>Специальные</td><td>M1000~M1999, 1000 точек, некоторые энергонезавис.</td></tr>
<tr><td rowspan="5">T</td><td rowspan="5">Таймеры</td><td rowspan="3">100 мс
(M1028 включен, T64~T126:
10 мс)</td><td>T0~T126, 127 точек, (*1)
T128~T183, 56 точек, (*1)</td><td rowspan="5">Всего 256 точек</td></tr>
<tr><td>T184~T199 для подпрограмм, 16 точек, (*1)</td></tr>
<tr><td>T250~T255 (аккумулятивные), 6 точек (*1)</td></tr>
<tr><td rowspan="2">10 мс
(M1038 включен, T200~T245: 1 мс)</td><td>T200~T239, 40 точек, (*1)</td></tr>
<tr><td>T240~T245 (аккумулятивные), 6 точек, (*1)</td></tr>
<tr><td rowspan="6">C</td><td rowspan="6">Счетчики</td><td>1 мс</td><td>T127, 1 точка, (*1)
T246~T249 (аккумулятивные), 4 точки, (*1)</td></tr>
<tr><td colspan="2" rowspan="2">16-бит инкрементный</td><td>C0~C111, 112 точек, (*1)
C128~C199, 72 точки, (*1)</td><td rowspan="4">Всего 233 точки</td></tr>
<tr><td>C112~C127, 16 точек, (*2)</td></tr>
<tr><td colspan="2" rowspan="2">32-бит инкрем./ декрем.</td><td>C200~C223, 24 точки, (*1)</td></tr>
<tr><td>C224~C232, 9 точек, (*2)</td></tr>
<tr><td rowspan="6">32-бит высокоскоростной инкр./ декрем.</td><td rowspan="2">Программный</td><td>C235~C242, 1 фаза 1 вход, 8 точек, (*2)</td><td rowspan="6">Всего 22 точки</td></tr>
<tr><td>C233~C234, 2 фазы 2 входа, 2 точки, (*2)</td></tr>
<tr><td rowspan="4">Аппаратный</td><td>C243~C244, 1 фаза 1 вход, 2 точки, (*2)</td></tr>
<tr><td>C245~C250, 1 фаза 2 входа, 6 точек, (*2)</td></tr>
<tr><td>C251~C254 2 фазы 2 входа, 4 точки, (*2)</td></tr>
<tr><td rowspan="5">S</td><td rowspan="5">Шаговые</td><td colspan="2">Начальные</td><td>S0~S9, 10 точек, (*2)</td><td rowspan="5">Всего 1024 точек</td></tr>
<tr><td colspan="2">Возврат к нулевой точке</td><td>S10~S19, 10 точек (с использованием команды IST), (*2)</td></tr>
<tr><td colspan="2">Энергонезависимые</td><td>S20~S127, 108 точек, (*2)</td></tr>
<tr><td colspan="2">Общие</td><td>S128~S911, 784 точки, (*1)</td></tr>
<tr><td colspan="2">Сигнальные</td><td>S912~S1023, 112 точек, (*2)</td></tr>
</table>

Описание					
Словные регистры	T	Текущее значение таймера		T0~T255, 256 words	
	C	Текущее значение счетчика		C0~C199, 16-бит счетчик, 200 слов	
				C200~C254, 32-бит счетчик, 55 слов	
	D	Регистры данных	Общие	D0~D407, 408 слов, (*1) D600~D999, 400 слов, (*1) D3920~D9999, 6080 слов, (*1)	Всего 10000 точек
			Энергонезависимые	D408~D599, 192 слова, (*2) D2000~D3919, 1920 слов, (*2)	
			Специальные	D1000~D1999, 1000 слов, некоторые энергонезавис.	
			Индексные	E0~E7, F0~F7, 16 слов, (*1)	
Указатели	N	Управление обратной связью		N0~N7, 8 точек	
	P	Указатель		P0~P255, 256 точек	
	I	Прерывания	Внешние	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 точек (01: триггер с опросом по переднему фронту ┘‾, 00: триггер с опросом по заднему фронту ‾└)	
			С таймером	I602~I699, I702~I799, 2 точки (разрешение = 1 мс)	
			С высокоскоростным счетчиком	I010, I020, I030, I040, I050, I060, I070, I080, 8 точек	
			Коммуникационные	I140(COM1), I150(COM2), I160(COM3), 3 точки, (*3)	
Константы	K	Десятичные		K-32,768 ~ K32,767 (16-битная команда), K-2,147,483,648 ~ K2,147,483,647 (32-битная команда)	
	H	Шестнадцатеричные		H0000 ~ HFFFF (16-битная команда), H00000000 ~HFFFFFFFF (32-битная команда)	
Последовательные порты			SA2	COM1: встроенный RS-232 (Master/Slave) COM2: встроенный RS-485 (Master/Slave) COM3: встроенный RS-485 (Master/Slave) COM1 - порт для программирования (в основном).	
			SX2	COM1: встроенный RS-232 (Master/Slave) COM2: встроенный RS-485 (Master/Slave) COM3: встроенный USB (Slave) COM1 - порт для программирования (в основном).	
Часы реального времени				Год, месяц, день, неделя, часы, минуты, секунды	
Специальные входные/выходные модули				Правая сторона: До 8 специальных модулей Левая сторона: До 8 высокоскоростных спец. модулей	
Файловые регистры (*5)				K0~K4999, 5000 точек (*2)	

Заметки:

1. Энергозависимая область не может быть изменена.
2. Энергонезависимая область не может быть изменена.
3. COM1: встроенный порт RS232. COM2: встроенный порт RS485, COM3: встроенный порт RS485.SX2 не поддерживает I160.
4. SA2 и SX2 MPU занимает 16 входов (X0~X17) и 16 выходов (Y0~Y17). Внешние модули занимают входные точки, начиная с X20, и выходные точки, начиная с Y17.
5. Прошивка MPU версий 2.0 и выше позволяет MPU использовать файловые регистры. См. команды MEMR/MEMW для информации по чтению/записи данных.
6. При подключении специального модуля к правой стороне и отключенном M1183 задействуется весь диапазон регистров данных. Спец. модуль занимает 10 регистров данных.
7. При подключении специального модуля к левой стороне и отключенном M1182 задействуется весь диапазон регистров данных. Специальный модуль занимает 10 регистров данных.

2.4 SE Карта параметров

Описание					
Метод управления				Загруженная программа, циклическое сканирование	
Метод обработки входов/выходов				Метод пакетной обработки (при выполнении команды END)	
Скорость выполнения				Команды LD – 0.64 мкс, команды MOV – 2 мкс, 1000 шагов – 1 мс	
Язык программирования				Мнемокод+ лестничная диаграмма + SFC	
Размер программы				До 15872 шагов	
Битные контакты	X	Внешние входы		X0~X377, восьмеричные, макс. 256 точек	Всего 480+14 Вх/Вых(*4)
	Y	Внешние выходы		Y0~Y377, восьмеричные, макс. 256 точек	
	M	Реле	Общие	M0~M511, 512 точек, (*1) M768~M999, 232 точки, (*1) M2000~M2047, 48 точек, (*1)	Всего 4096 точек
			Энергонезависимые	M512~M767, 256 точек, (*2) M2048~M4095, 2048 точек, (*2)	
			Специальные	M1000~M1999, 1000 точек, некоторые энергонезавис.	
	T	Таймер	100 мс (M1028 включен, T64~T126: 10 мс)	T0~T126, 127 точек, (*1) T128~T183, 56 точек, (*1)	Всего 256 точек
				T184~T199 для подпрограмм, 16 точек, (*1)	
				T250~T255 (аккумулятивные), 6 точек (*1)	
			10 мс (M1038 включен, T200~T245: 1 мс)	T200~T239, 40 точек, (*1)	
				T240~T245 (аккумулятивные), 6 точек, (*1)	
			1 мс	T127, 1 точка, (*1) T246~T249 (аккумулятивные), 4 точки, (*1)	
	C	Счетчики	16-бит инкрементный	C0~C111, 112 точек, (*1) C128~C199, 72 точки, (*1)	Всего 232 точки
				C112~C127, 16 точек, (*2)	
			32-бит инкрем./ декрем.	C200~C223, 24 точки, (*1)	
				C224~C231, 8 точек, (*2)	
			32-бит высокоскоростной инкр./ декрем. Программный	C235~C242, 1 фаза 1 вход, 8 точек, (*2)	Всего 22 точки
				C233~C234, 2 фазы 2 входа, 2 точки, (*2)	
			32-бит высокоскоростной инкр./ декрем. Аппаратный	C243~C244, 1 фаза 1 вход, 2 точки, (*2)	
				C245~C248, 1 фаза 2 входа, 4 точек, (*2)	
				C251~C254 2 фазы 2 входа, 4 точки, (*2)	
	S	Шаговые	Начальные	S0~S9, 10 точек, (*2)	Всего 1024 точек
			Возврат к нулевой точке	S10~S19, 10 точек (с использованием команды IST), (*2)	
			Энергонезависимые	S20~S127, 108 точек, (*2)	
			Общие	S128~S911, 784 точки, (*1)	
			Сигнальные	S912~S1023, 112 точек, (*2)	

Словные регистры	T	Текущее значение таймера		T0~T255, 256 words	
	C	Текущее значение счетчика		C0~C199, 16-бит счетчик, 200 слов	
				C200~C254, 32-бит счетчик, 55 слов	
	D	Регистры данных	Общие	D0~D407, 408 слов, (*1) D600~D999, 400 слов, (*1) D3920~D9999, 6080 слов, (*1)	Всего 10000 точек
			Энергонезависимые	D408~D599, 192 слова, (*2) D2000~D3919, 1920 слов, (*2)	
			Специальные	D1000~D1999, 1000 слов, некоторые энергонезавис.	
			Индексные	E0~E7, F0~F7, 16 слов, (*1)	
Указатели	N	Управление обратной связью		N0~N7, 8 точек	
	P	Указатель		P0~P255, 256 точек	
	I	Прерывание	Внешние прерыватели	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 точек (01: триггер с опросом по переднему фронту ⌐, 00: триггер с опросом по заднему фронту ⌙)	
			С таймером	I602~I699, I702~I799, 2 точки (разрешение = 1 мс)	
			С высокоскоростным счетчиком	I010, I020, I030, I040, I050, I060, I070, I080, 8 точек	
			Коммуникационные	I140(COM1), I150(COM2), I160(COM3), 3 точки, (*3)	
Константы	K	Десятичные		K-32,768 ~ K32,767 (16-битная команда), K-2,147,483,648 ~ K2,147,483,647 (32-битная команда)	
	H	Шестнадцатеричные		H0000 ~ HFFFF (16-битная команда), H00000000 ~HFFFFFFFF (32-битная команда)	
Последовательные порты			SE	COM1: встроенный USB (Slave) COM2: встроенный RS-485 (Master/Slave) COM3: встроенный RS-485 (Master/Slave) Ethernet: встроенный Ethernet (см. Приложение B) COM1 - порт для программирования (в основном).	
Часы реального времени				Год, месяц, день, неделя, часы, минуты, секунды	
Специальные входные/выходные модули				Правая сторона: До 8 специальных модулей Левая сторона: До 8 высокоскоростных специальных модулей	

Заметки:

1. Энергозависимая область не может быть изменена.
2. Энергонезависимая область не может быть изменена.
3. COM1: встроенный порт RS232. COM2: встроенный порт RS485, COM3: встроенный порт RS485.SX2 не поддерживает I160.
4. SA2 и SX2 MPU занимает 16 входов (X0~X17) и 16 выходов (Y0~Y17). Внешние модули занимают входные точки, начиная с X20, и выходные точки, начиная с Y17.
5. При подключении специального модуля к правой стороне и отключенном M1183 задействуется весь диапазон регистров данных. Специальный модуль занимает 10 регистров данных.
6. При подключении специального модуля к левой стороне и отключенном M1182 задействуется весь диапазон регистров данных. Специальный модуль занимает 10 регистров данных.

2.5 Установка области энергонезависимой памяти

Тип памяти	Питание OFF=>ON	STOP=>RUN	RUN=>STOP	Очистка всех общих областей (M1031=ON)	Очистка всех энергонезависимых областей (M1032=ON)	Заводские установки
Энергозависимая	Очистка	Не меняется	Когда M1033=OFF, очистка	Очистка	Не меняется	0
			Когда M1033=ON, не меняется			
Энергонезависимая	Не меняется			Не меняется	Очистка	0
Специальные M, Специальные D, Индексные регистры	Начальные	Не меняется		Не меняется		Начальные установки

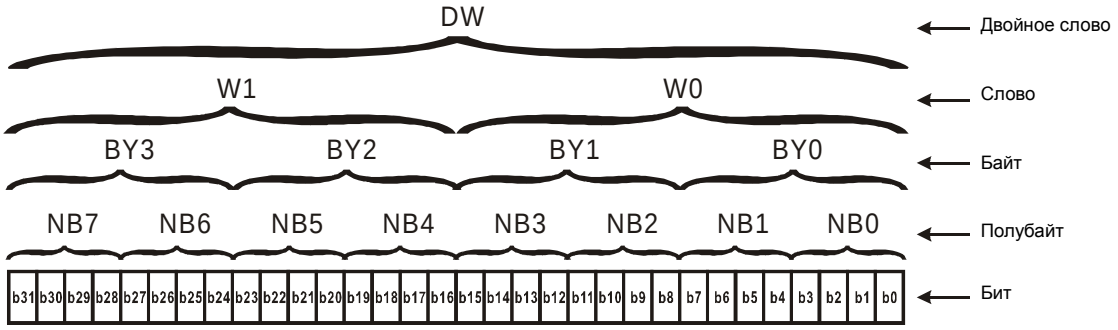
M Реле	Общие		Энергонезависимые		Специальные		
	M0~M511 M768~M999 M2000~M2047		M512~M999 M2048~M4095		M1000~M1999		
	Энергозависимые		Энергонезависимые		Некоторые энергонезависимые		
T Таймер	100 мс	100 мс	1 мс	10 мс	10 мс	1 мс	100 мс
	T0 ~T126 T128~T183	T184~T199	T127	T200~T239	T240~T245	T246~T249	T250~T255
	M1028=1,T64~T126 :10 мс	Для подпрограмм	-	M1038=1,T200~T245: 1 мс		-	
	Энергозависимые	Энергозависимые			Аккумулятивные энергозависимые		
C Счетчик	16-бит инкрементный		32-бит инкрем./декрем.		32-бит инкрем./декрем. высокоскоростной		
	C0~C111 C128~C199	C112~C127	C200~C223	C224~C231	C232~C254		
	Энергозависимые	Энергонезависимые	Энергозависимые	Энергонезависимые	Энергонезависимые		
S Шаговое реле	Начальные	Возврат к нулевой точке	Энергонезависимые	Общие	Сигнальные		
	S0~S9	S10~S19	S20~S127	S128~S911	S912~S1023		
	Энергонезависимые			Энергозависимые	Энергонезависимые		
D Регистр	Общие	Энергонезависимые	Специальные регистры	Для аналоговых модулей			
	D0~D407 D600~D999 D3920~D9899	D408~D599 D2000~D3919	D1000~D1999	D9900~D9999			
	Энергозависимые	Энергонезависимые	Некоторые энергонезависимые	Энергозависимые			

2.6 ПЛК: биты, полубайты, байты, слова, двойные слова

Существует пять форматов чисел, используемых в ПЛК DVP:

Формат	Описание
Бит	Бит – базовая величина двоичной системы счисления. Значения 0 или 1
Полубайт	Состоит из 4 последовательных бит: b3~b0. Представляет числа 0 ~ 9 в десятичном или 0~F в шестнадцатеричном форматах
Байт	Состоит из 2 последовательных полубайтов (8 бит): b7~b0. Представляет числа 00 ~ FF в шестнадцатеричном формате
Слово	Состоит из 2 последовательных байтов (16 бит): b15~b0. Представляет числа 0000 ~ FFFF в шестнадцатеричном формате
Двойное слово	Состоит из 2 последовательных слов (32 бита): b31~b1. Представляет числа 00000000 - FFFFFFFF в шестнадцатеричном формате

Бит, полубайт, байт, слово, двойное слово в двоичной системе:



2.7 Системы: двоичная, восьмеричная, десятичная, двоично-десятичная, шестнадцатеричная

Для выполнения различных внутренних операций ПЛК использует 5 систем счисления:

1. Двоичная, (BIN)
   Используется для внутренних вычислений ПЛК и хранения значений в памяти.

2. Восьмеричная, (OCT)
   Используется для нумерации внешних входов/выходов ПЛК.
   Внешние входы: X0～X7, X10～X17, ..., X377. (№ входа)
   Внешние выходы: Y0～Y7, Y10～Y17, ..., Y377. (№ выхода)

3. Десятичная, (DEC)
   Используется в следующих случаях:
   - Задание значений таймеров и счетчиков: TMR C0 K50. (значение K)
   - Нумерация операндов S, M, T, C, D, E, F, P, I: M10, T30. (№ операнда)
   - Для операндов в командах API: MOV K123 D0. (значение K)
   - Константы K:

   Десятичное значение определяется символом «K», т.е. K100 показывает значение 100 в десятичном формате.

Исключение: Когда константа K применяется для представления битовых операндов X, Y, M, S, 4 символа после «K» могут показывать: цифру (4 бит), байт (8 бит), слово (16 бит) или двойное слово (32 бит) в данных, т.е. K2Y10, K4M100 представляют Y10 ~ Y17 и M100~M115.

4. BCD (Двоично-десятичная)

BCD формат занимает 1 цифру или 4 бита, показывая число в десятичном формате, т.е. последовательные 16 бит показывают 4-х значное десятичное число. Используется, в основном, для чтения значений DIP-переключателей или для отправки данный на 7-сегментный дисплей.

5. Шестнадцатеричная, HEX

ПЛК DVP использует шестнадцатеричные операции в следующих случаях:

- Для операндов в командах API , например, MOV H1A2B D0 ◦ (значение H)
- Константа H:

Шестнадцатеричное значение в операциях PLC обозначается с “H”, т.е. H100 показывает значение 100 в шестнадцатеричном формате.

Таблица соотношений форматов:

Двоичное (BIN)	**Восьмеричное (OCT)**	**Десятичное (K) (DEC)**	**BCD (двоично-десятичное)**	**Шестнадцатеричное (H) (HEX)**
Для внутренних операций ПЛК	№ реле (вх/вых) X, Y	Константа K, № регистров M, S, T, C, D, E, F, P, I	Для DIP-переключателей и 7-сегментных индикаторов	Константа H
0000	0	0	0000	0
0001	1	1	0001	1
0010	2	2	0010	2
0011	3	3	0011	3
0100	4	4	0100	4
0101	5	5	0101	5
0110	6	6	0110	6
0111	7	7	0111	7
1000	10	8	1000	8
1001	11	9	1001	9
1010	12	10	0000	A
1011	13	11	0001	B
1100	14	12	0010	C
1101	15	13	0011	D
1110	16	14	0100	E
1111	17	15	0101	F
10000	20	16	0110	10
10001	21	17	0111	11

2.8 Реле M

Типы и функции специальных реле (специальные M) Представлены в таблице ниже. Необходимо помнить, что одинаковые M могут иметь разные значения в разных MPU. Специальные M и D, обозначенные “*” объяснены в разделе 2.13. Столбцы, помеченные “R” означают «только для чтения», “R/W” – «чтение и запись»”, “-“ состояние остается неизменным и “#” означает, что система настроит параметр в соответствии с состоянием ПЛК.

ON= ВКЛ, OFF=ВЫКЛ

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энергонезависимый	По умолчанию
M1000*	Указатель Н/О контакта	○	○	○	○	OFF	ON	OFF	R	NO	OFF
M1001*	Указатель Н/З контакта	○	○	○	○	ON	OFF	ON	R	NO	ON
M1002*	Положительный импульс (контакт замыкается) при включении ПЛК (Обычно - OFF)	○	○	○	○	OFF	ON	OFF	R	NO	OFF
M1003*	Отрицательный импульс (контакт размыкается) при включении ПЛК (Обычно - ON)	○	○	○	○	ON	OFF	ON	R	NO	ON
M1004*	Замыкается при ошибке	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1008*	Сторожевой таймер (ON: когда выходит время WDT)	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1009	Показывает сигнал LV 24VDC ниже допустимого	○	○	○	○	OFF	-	-	R	NO	OFF
M1011*	Генератор импульсов с периодом10мс, 5мс ON/5мс OFF	○	○	○	○	OFF	-	-	R	NO	OFF
M1012*	Генератор импульсов с периодом100мс, 50мс ON/50мс OFF	○	○	○	○	OFF	-	-	R	NO	OFF
M1013*	Генератор импульсов с периодом1с, 0,5с ON/0,5с OFF	○	○	○	○	OFF	-	-	R	NO	OFF
M1014*	Генератор импульсов с периодом1мин, 30с ON/30с OFF	○	○	○	○	OFF	-	-	R	NO	OFF
M1015*	Включение высокоскоростного таймера	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1016*	Отображение года (2/4 знака) в режиме реального времени (RTC).	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1017*	±30 сек. – корректировка реального времени	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1018	Флаг: Радиан/градусы, ON - градусы	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1020	Флаг нуля	○	○	○	○	OFF	-	-	R	NO	OFF
M1021	Флаг заимствования	○	○	○	○	OFF	-	-	R	NO	OFF
M1022	Флаг переноса	○	○	○	○	OFF	-	-	R	NO	OFF
M1024	Индикатор запроса передачи на COM1	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1025*	Указатель неверного запроса при связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1026	Выбор RAMP режима	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1027	Выбор режима PR выхода (8/16 байт)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1028	Переключение разрешения таймеров T64~T126 (10мс/100мс). ON=10мс	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1029*	CH0 (Y0, Y1): обработка импульсных выходов выполнена	○	○	○	○	OFF	-	-	R	NO	OFF
M1030*	Y1: обработка импульсного выхода выполнена	○	○	○	○	OFF	-	-	R	NO	OFF
M1031*	Очистка всей энергозависимой памяти	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1032*	Очистка всей энергонезависимой памяти	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1033*	Сохранение состояния выходов в режиме STOP	○	○	○	○	OFF	-	-	R/W	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1034*	Отключение всех выходов Y	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1035*	Включение входа X7 переключателем RUN/STOP	○	○	○	○	-	-	-	R/W	YES	OFF
M1037*	Включение 8-наб. функции SPD (применяется вместе с D1037)	×	×	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1038	Переключение разрешения таймеров T200~T255 (10мс/1мс). ON=1мс	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1039*	Фиксация времени сканирования	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1040	Отключение шагового режима STL	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1041	Запуск шагового режима STL	○	○	○	○	OFF	-	OFF	R/W	NO	OFF
M1042	Включение импульсных операций	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1043	Возврат в нулевую точку завершен	○	○	○	○	OFF	-	OFF	R/W	NO	OFF
M1044	Нахождение в нулевой точке	○	○	○	○	OFF	-	OFF	R/W	NO	OFF
M1045	Выключение функции “сброс всех выходов”	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1046	Индикация состояния STL	○	○	○	○	OFF	-	-	R	NO	OFF
M1047	Включение мониторинга STL	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1048	Индикация сигнального режима	○	○	○	○	OFF	-	-	R	NO	OFF
M1049	Включение мониторинга сигнального режима	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1050	Запрет прерывания I000 / I001	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1051	Запрет прерывания I100 / I101	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1052	Запрет прерывания I200 / I201	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1053	Запрет прерывания I300 / I301	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1054	Запрет прерывания I400 / I401	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1055	Запрет прерывания I500 / I501	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1056	Запрет прерывания I600~I699	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1057	Запрет прерывания I700~I799	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1058	Мониторинг запроса COM3	○	×	○	○	OFF	-	-	R/W	NO	OFF
M1059	Запрет прерывания высокоскоростным счетчиком I010~I080	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1060	Системная ошибка, сообщение 1	○	○	○	○	OFF	-	-	R	NO	OFF
M1061	Системная ошибка, сообщение 2	○	○	○	○	OFF	-	-	R	NO	OFF
M1062	Системная ошибка, сообщение 3	○	○	○	○	OFF	-	-	R	NO	OFF
M1063	Системная ошибка, сообщение 4	○	○	○	○	OFF	-	-	R	NO	OFF
M1064	Некорректное применение операнда	○	○	○	○	OFF	OFF	-	R	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1065	Синтаксическая ошибка	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1066	Ошибка цикла программы	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1067*	Ошибка выполнения программы	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1068*	Фиксация ошибки выполнения (D1068)	○	○	○	○	OFF	-	-	R	NO	OFF
M1070	Переключение времени импульса на выходе Y1 для команды PWM (ON: 100мкс; OFF: 1мс)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1071	Переключение времени импульса на выходе Y3 для команды PWM (ON: 100мкс; OFF: 1мс)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1072	Состояние ПЛК (RUN/STOP), ON = RUN	○	○	○	○	OFF	ON	OFF	R/W	NO	OFF
M1075	Ошибка записи во Flash-память	○	○	○	○	OFF	-	-	R	NO	OFF
M1078	Остановка импульсного выхо-даY0/CH0(Y0, Y1)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1079	Остановка импульсного выходаY1	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1080	Мониторинг запроса COM2	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1081	Изменение режима преобразования для команды FLT	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1083*	Выбор режима обнаружения ШИМ для X6. M1083 = ON, обнаружение ШИМ, когда X6 = ON; M1083 = OFF, обнару-жение ШИМ, когда X6 = OFF.	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1084*	Включение обнаружения ШИМ для X6 (применяется с M1183 и D1023)	○	○	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1085	Выбор дублирования функций DVP-PCC01	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1086	Включение функции пароля DVP-PCC01	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1088	Матричное сравнение. Сравнение эквивалентных значений (M1088 = ON) или различных значений (M1088 = OFF).	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1089	Индикация окончания матричного срав-нения. При достижении последнего би-та, M1089 = ON.	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1090	Индикация начала матричного сравне-ния. При нахождении первого бита, M1090 = ON.	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1091	Индикация результата матричного сравнения. При достижении результата сравнение останавливается и M1091 = ON.	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1092	Индикация ошибки указателя матрицы. Когда указатель Pr превышает задан-ный диапазон, M1092 = ON	○	○	○	○	OFF	OFF	-	R	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1093	Флаг увеличения указателя матрицы Pr на 1.	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1094	Сброс значения указателя матрицы Pr до 0	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1095	Флаг переноса для сдвига/вращения матриц	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1096	Флаг заимствования для сдви-га/вращения матриц	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1097	Флаг направления для сдвига/вращения матриц	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1098	Счетчик битов в матрице:“1” или “0”	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1099	Включается при результате счетчика битов в матрице, равного “0”	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1102*	Завершение обработки импульсных выходов Y2/CH1 (Y2, Y3)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1103*	Завершение обработки импульсного выхода Y3	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1104	Остановка импульсного выходаY2/CH1 (Y2, Y3)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1105	Остановка импульсного выходаY3	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1106	Выбор нулевой точки. M1106=ON, вы-бор справа от переключателяDOG для возврата в нулевую точку на CH0.	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1107	Выбор нулевой точки. M1107=ON, вы-бор справа от переключателяDOG для возврата в нулевую точку на CH1.	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1108	Y0/CH0 (Y0, Y1): пауза на импульсном выходе (по заднему фронту)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1109	Y1: пауза на импульсном выходе (по заднему фронту)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1110	Y2/CH1 (Y2, Y3): пауза на импульсном выходе (по заднему фронту)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1111	Y3: пауза на импульсном выходе (по заднему фронту)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1112	Переключение времени импульса на Y0 для команды PWM (ON: 100мкс; OFF: 1мс)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1113	Переключение времени импульса на Y2 для команды PWM (ON: 100мкс; OFF: 1мс)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1119*	Включение 2-х скоростной выходной функции для команды DDRVI	○	╳	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1120*	Сохранение настроек связи COM2 (RS-485), изменение D1120 невозможно при выставленном M1120	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1121	Для COM2(RS-485): готовность пере-дачи данных	○	○	○	○	OFF	OFF	-	R	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1122	Для COM2(RS-485): отправление заявки	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1123	Для COM2(RS-485): прием данных завершен	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1124	Для COM2(RS-485): готовность приема данных	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1125	Для COM2(RS-485): сброс состояния готовности к связи	○	○	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1126	Для COM2(RS-485): установка STX/ETX как определенные пользователем или системой	○	○	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1127	Для COM2(RS-485): прием / передача данных / преобразование выполнены. (не подд. командой RS)	○	○	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1128	Для COM2(RS-485): индикация состояния приема/передачи	○	○	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1129	Для COM2(RS-485): пауза приема	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1130	Для COM2(RS-485): выбор STX/ETX	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1131	Для COM2(RS-485): включен, когда данные MODRD/RDST/MODRW преобразуются из ASCII-режима в Hex-формат	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1132	Включен, когда отсутствует связь соответствующих команд в программе	○	○	○	○	OFF	-	-	R	NO	OFF
M1136*	Для COM3(RS-485/USB): сохранение настроек связи	○	╳	○	○	OFF	-	-	R/W	NO	OFF
M1137	Сохранение образа данных DNET во время паузы программы	╳	╳	○	○	-	-	-	R/W	NO	OFF
M1138*	Для COM1(RS-232): сохранение настроек связи. Изменение D1036 невозможно при выставленном M1138	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1139*	Для COM1(RS-232): выбор режима ASCII/RTU (OFF: ASCII; ON: RTU)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1140	Для COM2 (RS-485): MODRD / MODWR / MODRW ошибка приема данных	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1141	Для COM2 (RS-485): ошибка параметров MODRD / MODWR / MODRW	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1142	Ошибка приема данных VFD-A	○	○	○	○	OFF	OFF	-	R	NO	OFF
M1143*	Для COM2(RS-485): выбор режима ASCII/RTU (OFF: ASCII; ON: RTU)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1156*	Включение функции маскировки и маркировки I400/I401(X4), относящимся к Y0	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1158*	Включение функции маскировки и маркировки I600/I601(X6), относящимся к Y2	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1161	Выбор режима 8/16 бит (ON = 8 битный режим	○	○	○	○	OFF	-	-	R/W	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1162	Переключение между целым десятич-ным и двоичным с плавающей запятой для команды SCLP. ON: двоичное с плавающей запятой; OFF: десятичное целое	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1167	16-битный режим для входа HKY	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1168	Индикация режима работы SMOV	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1177	Выбор связи для ПЧ серии Delta VFD. ON: VFD-A (по умолчанию), OFF: дру-гие модели VFD	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1178	Включение VR0	×	×	○	○	OFF	-	-	R/W	NO	OFF
M1179	Включение VR1	×	×	○	○	OFF	-	-	R/W	NO	OFF
M1182	M1182 = ON, отключение автоотобра-жения при подключении левосторонних модулей ■ Для SA2 /SX2: значения модулей AIO будут автоматически отобра-жены в D9800 и выше. ■ Если левая сторона связана с ком-муникационным модулем, будут за-няты дополнительные 10 слов. Пример: 04AD-SL + EN01-SL + SA2, среднее значение Ch1~Ch4 из 04AD-SL отображаются в D9810~D9813.	×	×	○	○	OFF	-	-	R/W	NO	OFF
M1183	M1183 = ON, отключение автоотобра-жения при подключении специальных модулей #: ES2/EX2: OFF; SS2/SA2/SX2: ON (отображаются в D9900 и выше)	○	○	○	○	#	-	-	R/W	NO	#
M1190	Установка высокоскоростного выхода Y0 как 0.01 ~ 100Гц	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1191	Установка высокоскоростного выхода Y1 как 0.01 ~ 100Гц	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1192	Установка высокоскоростного выхода Y2 как 0.01 ~ 100Гц	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1193	Установка высокоскоростного выхода Y3 как 0.01 ~ 100Гц	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1200	Режим счета C200 (ON: обратный отсчет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1201	Режим счета C201 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1202	Режим счета C202 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1203	Режим счета C203 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1204	Режим счета C204 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1205	Режим счета C205 (ON:обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1206	Режим счета C206 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1207	Режим счета C207 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1208	Режим счета C208 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1209	Режим счета C209 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1210	Режим счета C210 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гонезави-симый	По умолча-нию
M1211	Режим счета C211 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1212	Режим счета C212 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1213	Режим счета C213 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1214	Режим счета C214 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1215	Режим счета C215 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1216	Режим счета C216 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1217	Режим счета C217 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1218	Режим счета C218 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1219	Режим счета C219 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1220	Режим счета C220 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1221	Режим счета C221 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1222	Режим счета C222 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1223	Режим счета C223 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1224	Режим счета C224 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1225	Режим счета C225 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1226	Режим счета C226 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1227	Режим счета C227 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1228	Режим счета C228 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1229	Режим счета C229 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1230	Режим счета C230 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1231	Режим счета C231 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1232	Режим счета C232 (ON: обратный от-счет)	╳	○	╳	╳	OFF	-	-	R/W	NO	OFF
	Режим счета C232 (ON: обратный от-счет)	○	╳	○	○	OFF	-	-	R	NO	OFF
M1233	Режим счета C233 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1234	Режим счета C234 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1235	Режим счета C235 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1236	Режим счета C236 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1237	Режим счета C237 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1238	Режим счета C238 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1239	Режим счета C239 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1240	Режим счета C240 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1241	Режим счета C241 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1242	Режим счета C242 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1243	C243 Управление сбросом. ON = сброс отключен	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1244	C244 Управление сбросом. ON = сброс отключен	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1245	Режим счета C245 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1246	Режим счета C246 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1247	Режим счета C247 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1248	Режим счета C248 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1249	Режим счета C249 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1250	Режим счета C250 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1251	Режим счета C251 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1252	Режим счета C252 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1253	Режим счета C253 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1254	Режим счета C254 (ON: обратный от-счет)	○	○	○	○	OFF	-	-	R	NO	OFF
M1257	Установка рампы на Y0, Y2 как “S-образная.” ON = S-образн.	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1260	Установка X7 для сброса сигнала со всех счетчиков C235 ~ C241	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1262	Отключение цикличного выхода для табличной функции команды DPTPO. ON = отключение.	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1270	Режим счета C235 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1271	Режим счета C236 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1272	Режим счета C237 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1273	Режим счета C238 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1274	Режим счета C239 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1275	Режим счета C240 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1276	Режим счета C241 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1277	Режим счета C242 (ON: счет по заднему фронту)	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1280*	Для I000 / I001: обратный сигнал прерывателя	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1284*	Для I400 / I401: обратный сигнал прерывателя	○	○	○	○	OFF	OFF	-	R/W	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энергонезависимый	По умолчанию
M1286*	Для I600 / I601: обратный сигнал прерывателя	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1303	Обмен старшим/младшими байтами для команды XCH	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1304*	Принудительное изменение состояния ON/OFF входа X	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1305	Реверсивная работа импульсного выхода Y1 для команд обработки высокоскоростных импульсных выходов	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1306	Реверсивная работа импульсного выхода Y3 для команд обработки высокоскоростных импульсных выходов	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1307	Для команды ZRN: включение левого концевого выключателя	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1308*	Выдача указанного количества выходных импульсов или поиск Z фазы сигнала при достижении нулевой точки.	○	○	○	○	OFF	OFF	OFF	R/W	NO	OFF
M1312	Для COM1(RS-232): заявка отправлена (только для команд MODRW и RS)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1313	Для COM1(RS-232): готовность приема данных (только для команд MODRW и RS)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1314	Для COM1(RS-232): прием данных завершен (только для команд MODRW и RS)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1315	Для COM1(RS-232): ошибка приема данных (только для команд MODRW и RS)	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1316	Для COM3(RS-232): заявка отправлена (только для команд MODRW и RS)	○	×	○	×	OFF	OFF	-	R/W	NO	OFF
M1317	Для COM3(RS-232): готовность приема данных (только для команд MODRW и RS)	○	×	○	×	OFF	OFF	-	R/W	NO	OFF
M1318	Для COM3(RS-232): прием данных завершен (только для команд MODRW и RS)	○	×	○	×	OFF	OFF	-	R/W	NO	OFF
M1319	Для COM3(RS-232): ошибка приема данных (только для команд MODRW и RS)	○	×	○	×	OFF	OFF	-	R/W	NO	OFF
M1320*	Для COM3 (RS-485): выбор режима ASCII/RTU. (OFF: ASCII; ON: RTU)	○	×	○	×	OFF	-	-	R/W	NO	OFF
M1346*	Очистка выходного сигнала после завершения команды ZRN	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1347	Автоочистка Y0 после завершения работы высокоскоростного импульсного выхода	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1348	Автоочистка Y1 после завершения работы высокоскоростного импульсного выхода	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1350*	Включение PLC LINK	○	○	○	○	Off	-	OFF	R/W	NO	OFF
M1351*	Запуск в автоматическом режиме PLC LINK	○	○	○	○	OFF	-	-	R/W	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энергонезависимый	По умолчанию
M1352*	Запуск в ручном режиме PLC LINK PLC LINK	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1353*	Включение доступа размером до 50 слов через PLC LINK (если M1353 = ON, D1480~D1511 энергонезависимые.)	○	○	○	○	OFF	-	-	R/W	YES	OFF
M1354*	Включение одновременного чтения/записи в опросе по PLC LINK	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1355*	Выбор Slave-режима в PLC LINK (ON: ручной; OFF: автообнаружение)	○	○	○	○	-	-	-	R/W	YES	OFF
M1356*	Включение выбора абонентского номера устройства. Когда M1353 и M1356 включены, пользователь может указать абонентский номер в D1900~D1931	○	×	○	○	-	-	-	R/W	YES	OFF
M1360*	Статус Slave ID#1 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1361*	Статус Slave ID#2 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1362*	Статус Slave ID#3 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1363*	Статус Slave ID#4 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1364*	Статус Slave ID#5 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1365*	Статус Slave ID#6 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1366*	Статус Slave ID#7 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1367*	Статус Slave ID#8 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1368*	Статус Slave ID#9 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1369*	Статус Slave ID#10 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1370*	Статус Slave ID#11 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1371*	Статус Slave ID#12 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1372*	Статус Slave ID#13 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1373*	Статус Slave ID#14 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1374*	Статус Slave ID#15 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1375*	Статус Slave ID#16 в сети PLC LINK	○	○	○	○	-	-	-	R/W	YES	OFF
M1376*	Индикация Slave обмена данными о состоянии PLC LINK для ID#1	○	○	○	○	OFF	-	-	R	NO	OFF
M1377*	Индикация Slave обмена данными о состоянии PLC LINK для ID#2	○	○	○	○	OFF	-	-	R	NO	OFF
M1378*	Индикация Slave обмена данными о состоянии PLC LINK для ID#3	○	○	○	○	OFF	-	-	R	NO	OFF
M1379*	Индикация Slave обмена данными о состоянии PLC LINK для ID#4	○	○	○	○	OFF	-	-	R	NO	OFF
M1380*	Индикация Slave обмена данными о состоянии PLC LINK для ID#5	○	○	○	○	OFF	-	-	R	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1381*	Индикация Slave обмена данными о состоянии PLC LINK для ID#6	○	○	○	○	OFF	-	-	R	NO	OFF
M1382*	Индикация Slave обмена данными о состоянии PLC LINK для ID#7	○	○	○	○	OFF	-	-	R	NO	OFF
M1383*	Индикация Slave обмена данными о состоянии PLC LINK для ID#8	○	○	○	○	OFF	-	-	R	NO	OFF
M1384*	Индикация Slave обмена данными о состоянии PLC LINK для ID#9	○	○	○	○	OFF	-	-	R	NO	OFF
M1385*	Индикация Slave обмена данными о состоянии PLC LINK для ID#10	○	○	○	○	OFF	-	-	R	NO	OFF
M1386*	Индикация Slave обмена данными о состоянии PLC LINK для ID#11	○	○	○	○	OFF	-	-	R	NO	OFF
M1387*	Индикация Slave обмена данными о состоянии PLC LINK для ID#12	○	○	○	○	OFF	-	-	R	NO	OFF
M1388*	Индикация Slave обмена данными о состоянии PLC LINK для ID#13	○	○	○	○	OFF	-	-	R	NO	OFF
M1389*	Индикация Slave обмена данными о состоянии PLC LINK для ID#14	○	○	○	○	OFF	-	-	R	NO	OFF
M1390*	Индикация Slave обмена данными о состоянии PLC LINK для ID#15	○	○	○	○	OFF	-	-	R	NO	OFF
M1391*	Индикация Slave обмена данными о состоянии PLC LINK для ID#16	○	○	○	○	OFF	-	-	R	NO	OFF
M1392*	Slave ID#1 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1393*	Slave ID#2 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1394*	Slave ID#3 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1395*	Slave ID#4 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1396*	Slave ID#5 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1397*	Slave ID#6 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1398*	Slave ID#7 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1399*	Slave ID#8 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1400*	Slave ID#9 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1401*	Slave ID#10 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1402*	Slave ID#11 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1403*	Slave ID#12 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1404*	Slave ID#13 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1405*	Slave ID#14 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1406*	Slave ID#15 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1407*	Slave ID#16 ошибка связи	○	○	○	○	OFF	-	-	R	NO	OFF
M1408*	Индикация завершения чтения Slave ID#1	○	○	○	○	OFF	-	-	R	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энер-гоне-зави-симый	По умолча-нию
M1409*	Индикация завершения чтения Slave ID#2	○	○	○	○	OFF	-	-	R	NO	OFF
M1410*	Индикация завершения чтения Slave ID#3	○	○	○	○	OFF	-	-	R	NO	OFF
M1411*	Индикация завершения чтения Slave ID#4	○	○	○	○	OFF	-	-	R	NO	OFF
M1412*	Индикация завершения чтения Slave ID#5	○	○	○	○	OFF	-	-	R	NO	OFF
M1413*	Индикация завершения чтения Slave ID#6	○	○	○	○	OFF	-	-	R	NO	OFF
M1414*	Индикация завершения чтения Slave ID#7	○	○	○	○	OFF	-	-	R	NO	OFF
M1415*	Индикация завершения чтения Slave ID#8	○	○	○	○	OFF	-	-	R	NO	OFF
M1416*	Индикация завершения чтения Slave ID#9	○	○	○	○	OFF	-	-	R	NO	OFF
M1417*	Индикация завершения чтения Slave ID#10	○	○	○	○	OFF	-	-	R	NO	OFF
M1418*	Индикация завершения чтения Slave ID#11	○	○	○	○	OFF	-	-	R	NO	OFF
M1419*	Индикация завершения чтения Slave ID#12	○	○	○	○	OFF	-	-	R	NO	OFF
M1420*	Индикация завершения чтения Slave ID#13	○	○	○	○	OFF	-	-	R	NO	OFF
M1421*	Индикация завершения чтения Slave ID#14	○	○	○	○	OFF	-	-	R	NO	OFF
M1422*	Индикация завершения чтения Slave ID#15	○	○	○	○	OFF	-	-	R	NO	OFF
M1423*	Индикация завершения чтения Slave ID#16	○	○	○	○	OFF	-	-	R	NO	OFF
M1424*	Индикация завершения записи Slave ID#1	○	○	○	○	OFF	-	-	R	NO	OFF
M1425*	Индикация завершения записи Slave ID#2	○	○	○	○	OFF	-	-	R	NO	OFF
M1426*	Индикация завершения записи Slave ID#3	○	○	○	○	OFF	-	-	R	NO	OFF
M1427*	Индикация завершения записи Slave ID#4	○	○	○	○	OFF	-	-	R	NO	OFF
M1428*	Индикация завершения записи Slave ID#5	○	○	○	○	OFF	-	-	R	NO	OFF
M1429*	Индикация завершения записи Slave ID#6	○	○	○	○	OFF	-	-	R	NO	OFF
M1430*	Индикация завершения записи Slave ID#7	○	○	○	○	OFF	-	-	R	NO	OFF
M1431*	Индикация завершения записи Slave ID#8	○	○	○	○	OFF	-	-	R	NO	OFF
M1432*	Индикация завершения записи Slave ID#9	○	○	○	○	OFF	-	-	R	NO	OFF
M1433*	Индикация завершения записи Slave ID#10	○	○	○	○	OFF	-	-	R	NO	OFF
M1434*	Индикация завершения записи Slave ID#11	○	○	○	○	OFF	-	-	R	NO	OFF
M1435*	Индикация завершения записи Slave ID#12	○	○	○	○	OFF	-	-	R	NO	OFF
M1436*	Индикация завершения записи Slave ID#13	○	○	○	○	OFF	-	-	R	NO	OFF
M1437*	Индикация завершения записи Slave ID#14	○	○	○	○	OFF	-	-	R	NO	OFF
M1438*	Индикация завершения записи Slave ID#15	○	○	○	○	OFF	-	-	R	NO	OFF
M1439*	Индикация завершения записи Slave ID#16	○	○	○	○	OFF	-	-	R	NO	OFF

Спец. реле M	Функция	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энергонезависимый	По умолчанию
M1524	Авто-сброс Y2 после обработки высокоскоростного импульсного выхода	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1525	Авто-сброс Y3 после обработки высокоскоростного импульсного выхода	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1534	Включение времени замедления рампы, установленного в Y0. Используется вместе с D1348	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1535	Включение времени замедления рампы, установленного в Y2. Используется вместе с D1349	○	○	○	○	OFF	-	-	R/W	NO	OFF
M1538	Индикация состояния паузы Y0	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1539	Индикация состояния паузы Y1	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1540	Индикация состояния паузы Y2	○	○	○	○	OFF	OFF	-	R/W	NO	OFF
M1541	Индикация состояния паузы Y3	○	○	○	○	OFF	OFF	-	R/W	NO	OFF

2.9 Реле S

Начальный шаг реле Начальная команда в последовательной функциональной схеме (SFC).
S0~S9, всего 10 точек.

Реле возврата в нулевую точку Возврат в нулевую точку при использовании команды IST в программе. Реле возврата, не применяемые командой IST можно использовать в качестве общих шаговых реле.
S10~S19, всего 10 точек.

Энергонезависимые шаговые реле В SFC энергонезависимые реле сохранять статус после прекращения подачи питания и возобновят его при подаче питания. Состояние будет аналогично состоянию до потери питания.
S20 ~ S127, всего 108 точек.

Шаговые реле общего назначения Общие реле входят в SFC. Они будут очищены после прекращения подачи питания и возобновления подачи.
S128 ~ S911, всего 784 точек.

Сигнальные шаговые реле Используются с командой управления сигнализацией API 46 ANS в качестве контакта, сигнализирующего о неисправности и т.п.
S912 ~ S1023, всего 112 точек.

2.10 Таймер T

Шаг таймера, используемый в программах – 1 мс, 10 мс и 100 мс, и отсчет времени ведется только по нарастающей (обратного отсчета нет). Когда значение таймера достигает заданного, включается связанная с ним катушка (контакт). Заданное значение должно быть десятичной K и может содержаться в регистрах данных D.

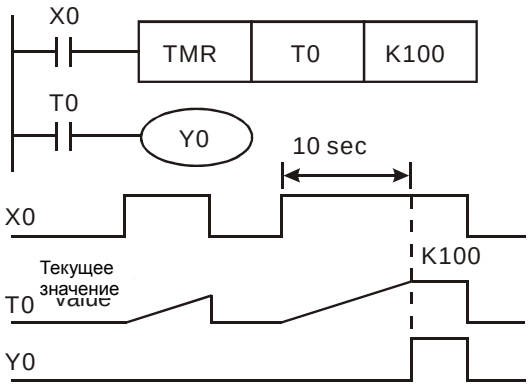
Реально установленное в таймере время = разрешение таймера x заданное значение

Пример: Если заданное значение K200 и разрешение 10 мс, реально установленное значение равно 10 мс*200 = 2000 мс = 2 сек.

Основной таймер

Таймер отрабатывается один раз при выполнении команды END. Когда команда TMR выполняется, катушка, связанная с таймером, включается при достижении заданного значения.

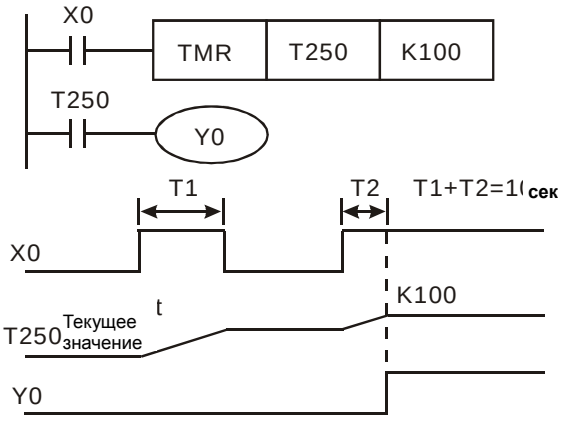
Когда X0 включается, выполняется команда TMR. Когда текущее значение достигает K100, связанный с таймером контакт T0 включается, управляя Y0. Если X0 выключен или выключено питание, текущее значение в T0 сбрасывается до 0 и выход Y0, работающий от контакта T0 выключается.



Аккумулятивный таймер

Таймер отрабатывается один раз при выполнении команды END. Когда команда TMR выполняется, катушка, связанная с таймером, включается при достижении заданного значения. Для аккумулятивных таймеров текущее значение не будет изменено при прерываниях отсчета времени.

Таймер T250 запускается при включении X0. Если X0 выключен или выключено питание, таймер T250 возьмет паузу в работе и сохранит текущее значение. При новом включении X0 T250 возобновит отсчет времени с места паузы.



Таймеры для подпрограмм и прерываний

Таймеры для подпрограмм и прерываний срабатывают один раз при выполнении команды END. Катушка, связанная с таймером, включается при достижении заданного значения при выполнении команды END. T184~T199 являются единственными таймерами для подпрограмм и прерываний. Основные таймеры, используемые в подпрограммах и прерываниях, не будут работать, если подпрограммы или прерывания не выполняются.

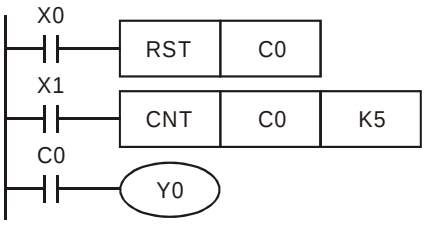
2.11 Счетчик C

Счетчики увеличивают текущее значение на 1 при появлении входных сигналов.

	16-битные счетчики	32-битные счетчики		
Тип	Основной	Основной	Высокоскоростной	
Счетчики	C0~C199	C200~C231(C232)	C232(C233)~C242, C245~C254	C243, C244
Направление счета	Прямой	Прямой/обратный		Прямой
Диапазон	0~32 767	-2 147 483 648 ~ +2 147 483 647		0~2 147 483 647
Регистры предустановок	Константа K или регистр данных D (слово)	Константа K или регистр данных D (двойное слово)		
Работа счетчика	Счетчик останавливается при достижении заданного значения	Счетчик продолжит отсчет при достижении заданного значения. Значение счетчика будет -2,147,483,648, если еще одно значение добавляется к +2,147,483,647		Счетчик продолжит отсчет при достижении заданного значения. Значение счетчика будет 0, если еще одно значение добавляется к +2,147,483,647
Работа выходного контакта	Выходная катушка включается при достижении заданного значения	Выходная катушка включается при достижении заданного значения.		Выходная катушка включается при достижении или превышении заданного значения
Высокоскоростное сравнение	-	Связанные устройства включаются при достижении заданного значения независимо от времени сканирования	-	
Сброс	Текущее значение будет сброшено до 0, когда команда RST выполнена, выходная катушка будет выключена.			

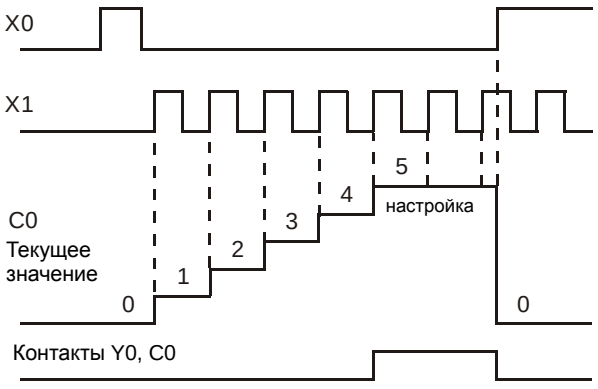
Пример:

```
LD      X0
RST     C0
LD      X1
CNT     C0  K5
LD      C0
OUT     Y0
```



Когда X0 включен, команда RST сбросит C0. В то же время, когда X1 запускается, C0 осуществит прямой счет (добавляя по 1).

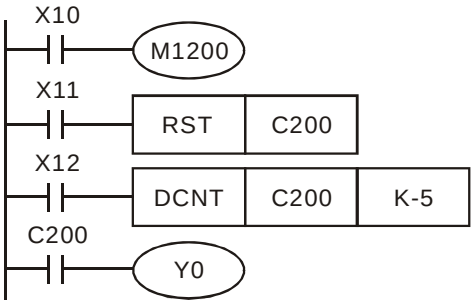
Когда C0 достигает заданного значения K5, выходная катушка Y0 включается и C0 остановит счет и будет игнорировать сигналы от X1.



Реле M1200~M1254 используются для установки направления счета соответственно C200~C254. Установка реле M во включенное положение запустит обратный счет.

Пример:

```
LD    X10
OUT   M1200
LD    X11
RST   C200
LD    X12
CNT   C200  K-5
LD    C200
OUT   Y0
```

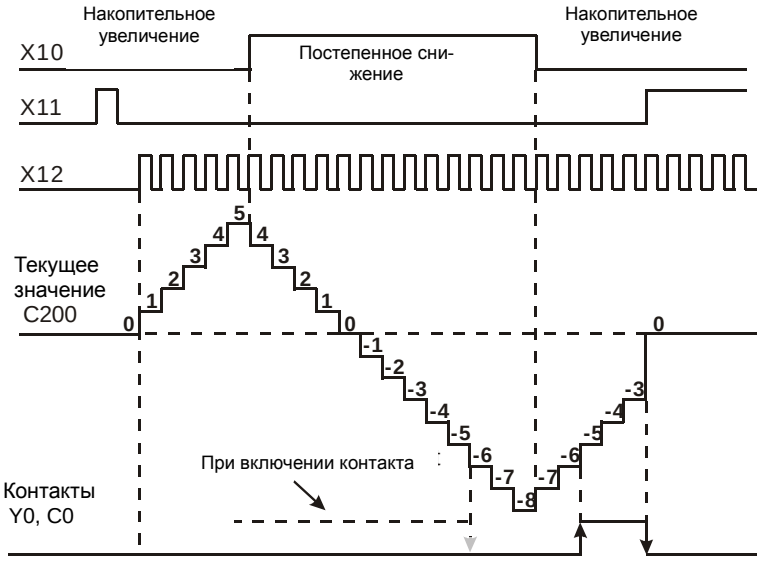


a) X10 запускает M1200 для определения направления счета C200

b) При включении X11 команда RST выполняется и PV (текущее значение) в C200 будет очищено, C200 выключится.

c) Когда включается X12, PV C200 считает инкрементно или декрементно.

d) Когда PV в C200 меняется с K-6 на K-5, контакт C200 будет активирован. Если наоборот, контакт C200 будет очищен.



2.12 Высокоскоростные счетчики

Существует два типа высокоскоростных счетчиков - программный (SHSC) и аппаратный (HHSC). Один и тот же вход (X) может использоваться только с одним высокоскоростной счетчиком, иначе возникает синтаксическая ошибка при выполнении команды DCNT.

Соответствия программных высокоскоростных счетчиков:

C / X	1-фазный вход								2 фазы 2 входа		
	C235	C236	C237	C238	C239	C240	C241	C242	C232	C233	C234
X0	U/D								A		
X1		U/D									
X2			U/D						B		
X3				U/D							
X4					U/D					A	
X5						U/D				B	
X6							U/D				A
X7								U/D			B
R/F	M1270	M1271	M1272	M1273	M1274	M1275	M1276	M1277	-	-	-
U/D	M1235	M1236	M1237	M1238	M1239	M1240	M1241	M1242	-	-	-

U: Прямой счет **D:** Обратный счет **A: Фаза A** **B: Фаза B**

Заметки:

1. U/D (прямой/обратный счет) определяется состоянием M. OFF - прямой; ON – обратный.
2. R/F (триггер по переднему / заднему фронту) определяется состоянием M. OFF – передний фронт; ON – задний фронт.
3. SHSC поддерживает макс. частоту входных импульсов 10 кГц на одной точке и не более 8 счетчиков одновременно.
4. SS2 не поддерживает 2-фазный 2-входной счетчик (X0,X2) (C232).
5. Для 2-фазного 2-входного счетчика (X4, X5) (C233) и (X6, X7) (C234) макс. 5 кГц, для (X0,X2) (C232) макс. 15 кГц.
6. 2-фазный 2-входной счетчик поддерживает двойные и четверные частоты, выбранные в D1022, как показано в табл. на стр. 2-32:

Соответствия аппаратных высокоскоростных счетчиков:

C / X	1-фазный вход		1 фаза 2 входа						2 фазы 2 входа			
	C243	C244	C245	C246	C247	C248	C249	C250	C251	C252	C253	C254
X0	U		U/D	U/D	U	U			A	A		
X1	R		Dir	Dir	D	D			B	B		
X2		U					U/D	U/D			A	A
X3		R					Dir	Dir			B	B
X4				R		R				R		
X5								R				R

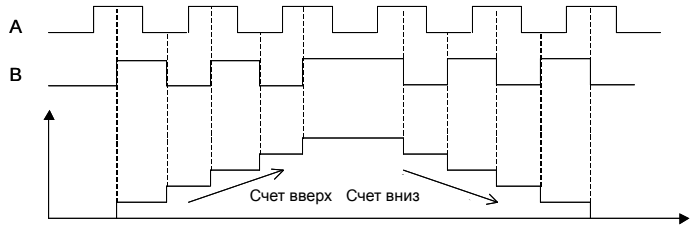
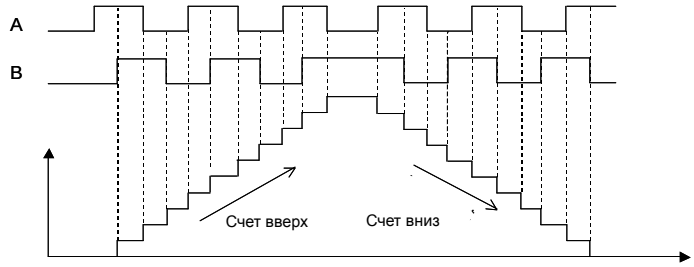
U: Прямой D: Обратный A: Вход фаза A B: Вход фаза B Dir: Направление входного сигнала R: Сброс входного сигнала

Заметки:

1. Максимальная частота 1-фазных входных счетчиков X0 (C243) и X2(C244) 100 кГц для ES2/EX2/SA2/SX2 и 20 кГц для SS2.
2. Максимальная частота 1-фазных 2-входных счетчиков (X0, X1)(C245, C246) и (X2, X3)(C249, C250) 100 кГц для ES2/EX2/SA2/SX2 и 20 кГц для SS2.
3. Максимальная частота 1-фазных 2-входных счетчиков (X0, X1)(C247, C248) 10 кГц для

ES2/EX2/SS2/SX2 и 100 кГц для 32ES211T и SA2.

4. Максимальная частота 2-фазных 2-входных счетчиков (X0, X1)(C251, C252) 5 кГц для ES2/EX2, 10 кГц для SS2/SX2 и 50 кГц для 32ES211T и SA2.
5. Максимальная частота 2-фазных 2-входных счетчиков (X2, X3)(C253, C254) 5 кГц для ES2/EX2/SA2, 10 кГц для SS2/SX2 и 50 кГц для 32ES211T.
6. 2-фазный 2-входной счетчик поддерживает двойные и четверные частоты, выбранные в D1022, как показано в табл. ниже:

D1022	Режим счета
K2 (двойная частота)	
K4 или другие (четверная частота) (по умолчанию)	

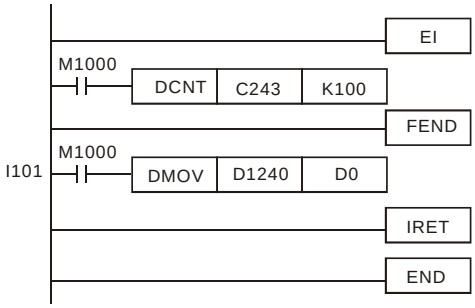
7. C243 и C244 поддерживают только прямой счет и реализует со связанными входами X1 и X3 функцию сброса (“R”). Для отключения функции сброса необходимо установить соответствующие реле (M1243 и M1244).
8. “Dir” определяет направление счета. OFF показывает прямой счет; ON – обратный.
9. Когда X1, X3, X4 и X5 используются для реализации функции сброса и связанные с ними внешние прерыватели отключены, пользователь может задать сброс специальными реле M (триггеры по переднему/заднему фронту – R/F)

Функция сброса	X1	X3	X4	X5
Триггер R/F	M1271	M1273	M1274	M1275

10. Когда X1, X3, X4 и X5 используются для реализации функции сброса и применяются связанные с ними внешние прерыватели, команды прерывания являются приоритетными для выходов. Кроме того, ПЛК будет записывать текущие данные счетчиков в регистры (см. табл. ниже), после чего счетчики будут обнулены.

Специальный регистр D	D1241, D1240				D1243, D1242		
Счетчик	C243	C246	C248	C252	C244	C250	C254
Внешний прерыватель	X1 (I100/I101)	X4(I400/I401)			X3 (I300/I301)	X5(I500/I501)	

Пример:

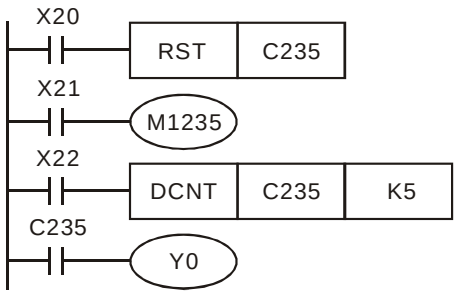


Когда C243 вдет счет и запускается внешний прерыватель из X1(I101), значение в C243 помещается в (D1241, D1240) и счетчик C243 немедленно обнуляется. После этого I101 выполняет прерывание.

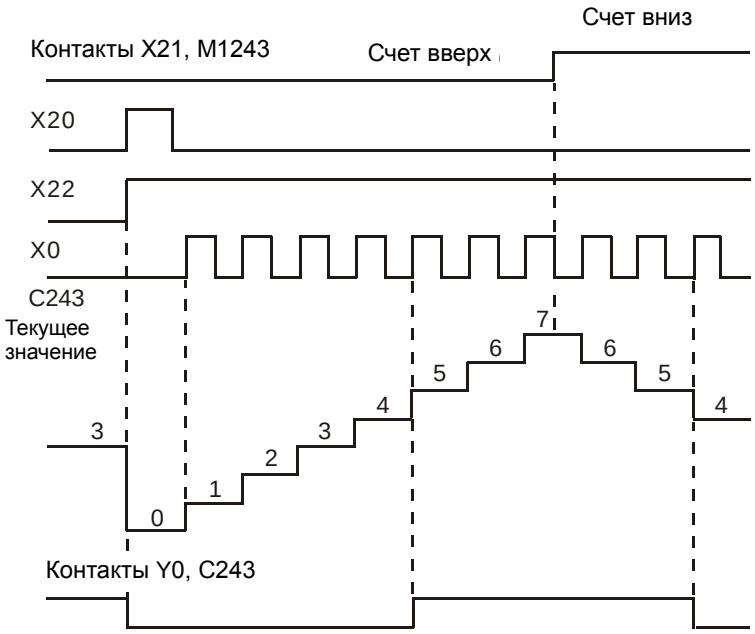
1-фазный 1–входный высокоскоростной счетчик:

Пример:

```
LD      X20
RST     C235
LD      X21
OUT     M1235
LD      X22
DCNT    C235  K5
LD      C235
OUT     Y0
```



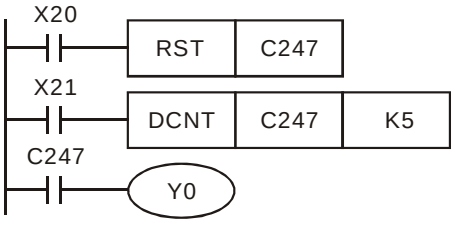
1. X21 запускает M1235 для определения направления счета (прямой/обратный) для C235.
2. Когда X20 включен, команда RST выполняется и текущее значение в C235 будет очищено. Контакт C235 будет отключен.
3. Когда X22 включен, C235 получает сигналы от X0 и счетчик считает инкрементно (+1) или декрементно (-1).
4. Когда счетчик C235 достигает K5, контакт C235 включается. При поступлении новых сигналов на X0 счет в C235 будет сохранен.



1-фазный 2-входный высокоскоростной счетчик:

Пример:

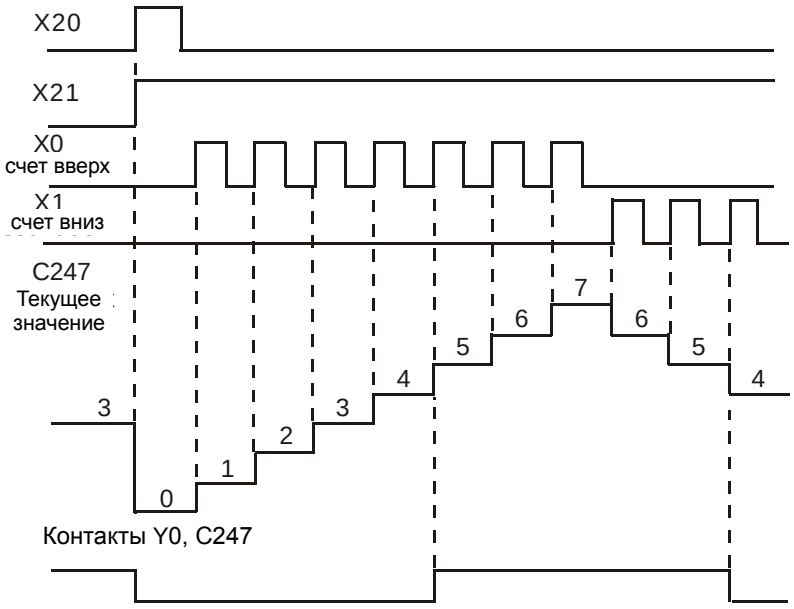
```
LD      X20
RST     C247
LD      X21
DCNT    C247 K5
LD      C247
OUT     Y0
```



a. Когда X20 включен, команда RST выполняется и текущее значение в C247 будет очищено. Контакт C247 будет отключен.

b. Когда X21 включен, C247 принимает счетные сигналы от X0 и счетчик считает инкрементно (+1), если C247 принимает счетные сигналы от X1 и счетчик считает декрементно (-1)

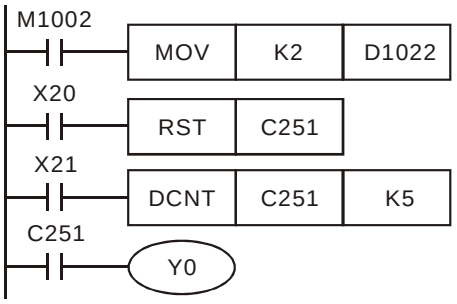
c. Когда счетчик C247 достигает K5, контакт C247 включается. При поступлении новых сигналов на X0 или X1 счет в C247 будет сохранен.



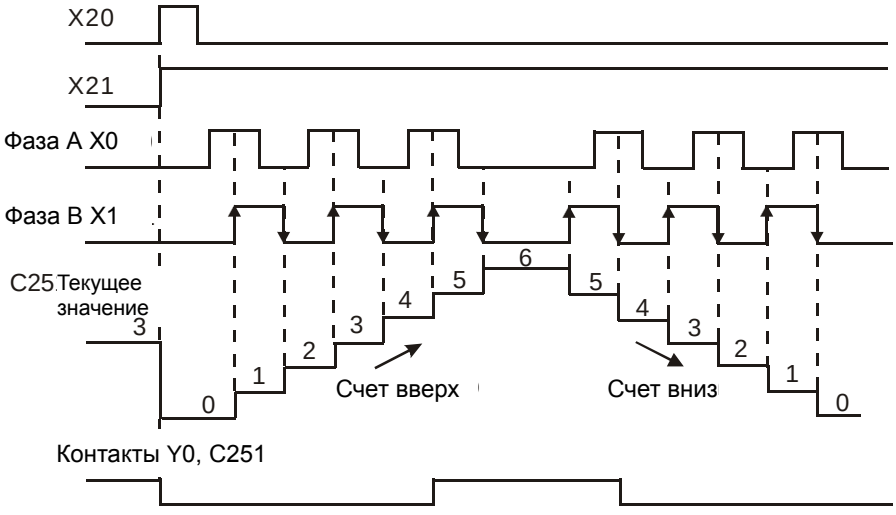
Высокоскоростной счетчик с AB-фазными входами:

Пример:

```
LD      M1002
MOV     K2 D1022
LD      X20
RST     C251
LD      X21
DCNT    C251 K5
LD      C251
OUT     Y0
```



1. Когда X20 включен, команда RST выполняется и текущее значение в C251 будет очищено. Контакт C251 будет отключен.
2. Когда X21 включен, C251 принимает A-фазу счетного сигнала от входа X0, B-фазу от входа X1 и осуществляет инкрементный или декрементный счет.
3. Когда счетчик C251 достигает K5, контакт C251 включается. При поступлении новых сигналов на X0 или X1 счет в C251 будет сохранен.
4. Режим счета задается как двойная или четверная частота в D1022. По умолчанию: четверная.



2.13 Специальные регистры данных

Типы и функции специальных регистров (D) приведены в таблице ниже. Следует помнить, что регистры с одинаковым номером могут иметь разные значения для различных MPU. Специальные M и D, обозначенные "*" рассмотрены в разделе 2.13. Маркировка "R" означает "только для чтения", "R/W" - "чтение и запись", "-" показывает неизменный статус и "#" показывает системные настройки ПЛК. Для детального описания см. раздел 2.13.

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолчанию
D1000*	Сторожевой таймер WDT (шаг: 1мс)	○	○	○	○	200	-	-	R/W	NO	200
D1001	Отображение версии ПО ПЛК DVP(заводская установка)	○	○	○	○	-	-	-	R	NO	#
D1002*	Объем программы	○	○	○	○	-	-	-	R	NO	#
D1003	Суммарная память программы	○	○	○	○	#	-	-	R	YES	15872
D1004*	Код синтаксических ошибок	○	○	○	○	0	0	-	R	NO	0
D1008*	Шаговый адрес при WDT = ON	○	○	○	○	0	-	-	R	NO	0
D1009	Количество появлений сигналов LV (низкое напряжение)	○	○	○	○	-	-	-	R	YES	0
D1010*	Текущее время сканирования (шаг: 0.1мс)	○	○	○	○	#	#	#	R	NO	0
D1011*	Минимальное время сканирования (шаг: 0.1мс)	○	○	○	○	#	#	#	R	NO	0
D1012*	Максимальное время сканирования (шаг: 0.1мс)	○	○	○	○	#	#	#	R	NO	0
D1015*	Значение аккумулятивного высокоскоростного таймера (0~32 767 шаг: 0.1мс)	○	○	○	○	0	-	-	R/W	NO	0
D1018*	Число π (младший байт)	○	○	○	○	H' 0FDB	H' 0FDB	H' 0FDB	R/W	NO	H' 0FDB
D1019*	Число π (старший байт)	○	○	○	○	H' 4049	H' 4049	H' 4049	R/W	NO	H' 4049
D1020*	Входной фильтр X0~X7 (шаг: 1ms) диапазон 0~20мс	○	○	○	○	10	-	-	R/W	NO	10
D1022	Выбор режима счета (двойная частота/ четверная частота) для AB-фазного счетчика (входы X0, X1)	○	○	○	○	4	-	-	R/W	NO	4
D1023*	Регистр хранения обнаруженной ширины импульса (шаг: 0.1ms)	○	○	○	○	0	-	-	R/W	NO	0
D1025*	Код ошибки коммуникации	○	○	○	○	0	-	-	R	NO	0
D1026*	Номер импульса для маскировки Y0 когда M1156 = ON (младшее слово)	○	○	○	○	0	0	-	R/W	NO	0
D1027*	Номер импульса для маскировки Y0 когда M1156 = ON (старшее слово)	○	○	○	○	0	0	-	R/W	NO	0
D1028	Индексный регистр E0	○	○	○	○	0	-	-	R/W	NO	0
D1029	Индексный регистр F0	○	○	○	○	0	-	-	R/W	NO	0
D1030	Количество импульсов на выходе Y0 (младшее слово)	○	○	○	○	-	-	-	R/W	YES	0
D1031	Количество импульсов на выходе Y0 (старшее слово)	○	○	○	○	-	-	-	R/W	YES	0
D1032	Количество импульсов на выходе Y1 (младшее слово)	○	○	○	○	0	-	-	R/W	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1033	Количество импульсов на выходе Y1 (старшее слово)	○	○	○	○	0	-	-	R/W	NO	0
D1036*	COM1 (RS-232) протокол связи	○	○	○	○	H'86	-	-	R/W	NO	H'86
D1037*	Регистр установок 8-наб. функции SPD (используется с M1037)	○	○	○	○	0	-	-	R/W	NO	0
D1038	1. Время задержки ответа ПЛК в SLAVE-режиме по COM2 / COM3 RS-485. Диапазон: 0 ~ 10 000 (шаг: 0.1мс). 2. С помощью PLC LINK по COM2 (RS-485), D1038 устанавливает время задержки следующей отправки. Диапазон: 0 ~ 10 000 (шаг: 1 цикл сканирования)	○	○	○	○	-	-	-	R/W	NO	0
D1039*	Фиксированное время сканирования (мс)	○	○	○	○	0	-	-	R/W	NO	0
D1040	Состояние ON для 1-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1041	Состояние ON для 2-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1042	Состояние ON для 3-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1043	Состояние ON для 4-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1044	Состояние ON для 5-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1045	Состояние ON для 6-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1046	Состояние ON для 7-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1047	Состояние ON для 8-й шаговой точки.	○	○	○	○	0	-	-	R	NO	0
D1049	Состояние ON для сигнализации	○	○	○	○	0	-	-	R	NO	0
D1050 ↓ D1055	Конвертирование данных для обмена данными по Modbus. ПЛК автоматически преобразует данные ASCII в D1070~D1085 в данные Hex и сохраняет 16-битныet Hex данные в D1050~D1055	○	○	○	○	0	-	-	R	NO	0
D1062*	Среднее число аналоговых сигналов, подаваемых на ПЛК серий EX2/SX2. По умолчанию K10 для EX2 версий 2.6 и 2.8.	○	×	×	○	2	-	-	R/W	YES	2
D1067*	Код ошибки выполнения программы	○	○	○	○	0	0	-	R	NO	0
D1068*	Адрес ошибки выполнения программы	○	○	○	○	0	-	-	R	NO	0
D1070 ↓ D1085	Данные обратной связи (ASCII) по протоколу Modbus. Когда по RS-485 инструкции связи получают сигнал обратной связи, данные сохраняются в регистрах D1070~D1085..	○	○	○	○	0	-	-	R	NO	0
D1086	Старшее слово пароля DVP-PCC01 (отображается в Hex в соответствии с ASCII -кодами)	○	○	○	○	0	-	-	R/W	NO	0
D1087	Младшее слово пароля DVP-PCC01 (отображается в Hex в соответствии с ASCII -кодами)	○	○	○	○	0	-	-	R/W	NO	0
D1089 ↓ D1099	Принятые по Modbus данные.	○	○	○	○	0	-	-	R	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолчанию
D1109*	COM3 (RS-485) Протокол связи	○	×	○	○	H'86	-	-	R/W	NO	H'86
D1110*	Среднее число аналоговых сигналов на входе канала 0 (AD 0) для EX2/SX2. Когда среднее в D1062 установлено как 1, D1110 показывает текущее значение.	○	×	×	○	0	-	-	R	NO	0
D1111*	Среднее число аналоговых сигналов на входе канала 1 (AD 1) для EX2/SX2. Когда среднее в D1062 установлено как 1, D1111 показывает текущее значение.	○	×	×	○	0	-	-	R	NO	0
D1112*	Среднее число аналоговых сигналов на входе канала 2 (AD 2) для EX2/SX2. Когда среднее в D1062 установлено как 1, D1112 показывает текущее значение.	○	×	×	○	0	-	-	R	NO	0
D1113*	Среднее число аналоговых сигналов на входе канала 3 (AD 3) для 20EX2/SX2. Когда среднее в D1062 установлено как 1, D1113 показывает текущее значение.	○	×	×	○	0	-	-	R	NO	0
	Отображение состояния аналогового ввода 30EX2	○	×	×	×	0	-	-	R	NO	0
D1114*	Вкл/выкл 20EX2/SX2 AD каналов (0: вкл (по умолчанию) / 1: выкл) bit0~bit3 устанавливают AD0~AD3. P.S. 30EX2 не поддерживает функцию.	○	×	×	○	0	-	-	R/W	YES	0
D1115*	20EX2/SX2 установка аналогового входного/выходного режима	○	×	×	○	0	0	0	R/W	YES	0
	30EX2 установка аналогового входного/выходного режима	○	×	×	×	-	-	-	R/W	YES	H'FFFF
D1116*	Выходное значение на аналоговом выходном канале 0 (DA 0) для EX2/SX2	○	×	×	○	0	0	0	R/W	NO	0
D1117*	Выходное значение на аналоговом выходном канале 1 (DA 0) для EX2/SX2 P.S. 30EX2 не поддерживает данную функцию.	○	×	×	○	0	0	0	R/W	NO	0
D1118*	EX2/SX2: время выборки (фильтр) аналого-цифрового преобразователя. По умолчанию: 2мс. Шаг: 1мс. Время выборки будет 2мс, если D1118≦2	○	×	×	○	2	-	-	R/W	YES	2
D1120*	COM2 (RS-485) протокол связи	○	○	○	○	H'86	-	-	R/W	NO	H'86
D1121*	COM1(RS-232) и COM2(RS-485): Коммуникационные адреса	○	○	○	○	-	-	-	R/W	Yes	1
D1122	COM2(RS-485): кол-во остаточных от передачи данных слов	○	○	○	○	0	0	-	R	NO	0
D1123	COM2(RS-485): кол-во остаточных от приема данных слов	○	○	○	○	0	0	-	R	NO	0
D1124	COM2(RS-485): Начало символьного определения (STX)	○	○	○	○	H'3A	-	-	R/W	NO	H'3A
D1125	COM2(RS-485): Первое окончание символьного определения (ETX1)	○	○	○	○	H'0D	-	-	R/W	NO	H'0D
D1126	COM2(RS-485) Второе окончание символьного определения (ETX2)	○	○	○	○	H'0A	-	-	R/W	NO	H'0A
D1127	Число импульсов рамповых операций (нарастание) для команд позиционирования (младшее слово)	○	○	○	○	0	-	-	R/W	NO	0
D1128	Число импульсов рамповых операций	○	○	○	○						

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
	(нарастание) для команд позициониро-вания (старшее слово)										
D1129	COM2 (RS-485) установка паузы связи (мс)	○	○	○	○	0	-	-	R/W	NO	0
D1130	COM2 (RS-485) Код ошибки, возвра-щенный Modbus	○	○	○	○	0	-	-	R	NO	0
D1131	Вх/вых процентное значение на CH0(Y0,Y1) закрытого контура управле-ния	○	○	○	○	100	-	-	R/W	NO	100
D1132	Вх/вых процентное значение на CH1(Y2,Y3) закрытого контура управле-ния	○	○	○	○	100	-	-	R/W	NO	100
D1133	Число импульсов рамповых операций (замедление) для команд позициониро-вания (младшее слово)	○	○	○	○	0	-	-	R	NO	0
D1134	Число импульсов рамповых операций (замедление) для команд позициониро-вания (старшее слово)	○	○	○	○	0	-	-	R	NO	0
D1135*	Число импульсов для маскировки Y2 когда M1158 = ON (младшее слово)	○	○	○	○	0	0	-	R/W	NO	0
D1136*	Число импульсов для маскировки Y2 когда M1158 = ON (старшее слово)	○	○	○	○	0	0	-	R/W	NO	0
D1137*	Адрес некорректного использования операндов	○	○	○	○	0	0	-	R	NO	0
D1140*	Число вх/вых модулей (макс. 8)	○	○	○	○	0	-	-	R	NO	0
D1142*	Число точек ввода (X) модулей расши-рения	○	○	○	○	0	-	-	R	NO	0
D1143*	Число точек вывода (X) модулей расши-рения	○	○	○	○	0	-	-	R	NO	0
D1145*	Количество подключенных по левой стороне модулей	×	×	○	○	0	-	-	R	NO	0
D1167	Конечное слово обнаружения для ко-манды RS – выполнение запроса пре-рывания (I140) на COM1 (RS-232).	○	○	○	○	0	-	-	R/W	NO	0
D1168	Конечное слово обнаружения для ко-манды RS – выполнение запроса пре-рывания (I150) на COM2 (RS-485)	○	○	○	○	0	-	-	R/W	NO	0
D1169	Конечное слово обнаружения для ко-манды RS – выполнение запроса пре-рывания (I160) на COM3 (RS-485)	○	×	○	×	0	-	-	R/W	NO	0
D1178	Значение VR0	×	×	○	○	0	-	-	R	NO	0
D1179	Значение VR1	×	×	○	○	0	-	-	R	NO	0
D1182	Индексный регистр E1	○	○	○	○	0	-	-	R/W	NO	0
D1183	Индексный регистр F1	○	○	○	○	0	-	-	R/W	NO	0
D1184	Индексный регистр E2	○	○	○	○	0	-	-	R/W	NO	0
D1185	Индексный регистр F2	○	○	○	○	0	-	-	R/W	NO	0
D1186	Индексный регистр E3	○	○	○	○	0	-	-	R/W	NO	0
D1187	Индексный регистр F3	○	○	○	○	0	-	-	R/W	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1188	Индексный регистр E4	○	○	○	○	0	-	-	R/W	NO	0
D1189	Индексный регистр F4	○	○	○	○	0	-	-	R/W	NO	0
D1190	Индексный регистр E5	○	○	○	○	0	-	-	R/W	NO	0
D1191	Индексный регистр F5	○	○	○	○	0	-	-	R/W	NO	0
D1192	Индексный регистр E6	○	○	○	○	0	-	-	R/W	NO	0
D1193	Индексный регистр F6	○	○	○	○	0	-	-	R/W	NO	0
D1194	Индексный регистр E7	○	○	○	○	0	-	-	R/W	NO	0
D1195	Индексный регистр F7	○	○	○	○	0	-	-	R/W	NO	0
D1220	Первая группа импульсных выходов CH0 (Y0, Y1)	○	○	○	○	0	-	-	R/W	NO	0
D1221	Вторая группа импульсных выходов CH1 (Y2, Y3)	○	○	○	○	0	-	-	R/W	NO	0
D1232*	Число выходных импульсов для CH0 (Y0, Y1) –замедление рамповой функ-ции при получении сигнала датчиком маркировки (младшее слово).	○	○	○	○	0	0	--	R/W	NO	0
D1233*	Число выходных импульсов для CH0 (Y0, Y1) –замедление рамповой функ-ции при получении сигнала датчиком маркировки (старшее слово).	○	○	○	○	0	0	--	R/W	NO	0
D1234*	Число выходных импульсов для CH1 (Y2, Y3) –замедление рамповой функ-ции при получении сигнала датчиком маркировки (младшее слово).	○	○	○	○	0	0	--	R/W	NO	0
D1235*	Число выходных импульсов для CH1 (Y2, Y3) –замедление рамповой функ-ции при получении сигнала датчиком маркировки (старшее слово).	○	○	○	○	0	0	--	R/W	NO	0
D1240*	При срабатывании прерывателей I400/I401/I100/I101 D1240 сохраняет младшее слово высокоскоростного счетчика	○	○	○	○	0	0	-	R	NO	0
D1241*	При срабатывании прерывателей I400/I401/I100/I101 D1240 сохраняет старшее слово высокоскоростного счет-чика.	○	○	○	○	0	0	-	R	NO	0
D1242*	При срабатывании прерывателей I500/I501/I300/I301 D1242 сохраняет младшее слово высокоскоростного счетчика	○	○	○	○	0	0	-	R	NO	0
D1243*	При срабатывании прерывателей I500/I501/I300/I301 D1242 сохраняет старшее слово высокоскоростного счет-чика	○	○	○	○	0	0	-	R	NO	0
D1244	Время простоя (число импульсов), уста-новленное для CH0 (Y0, Y1). Функция не работает при установленном значении ≦0.	○	○	○	○	0	-	-	R/W	NO	0
D1245	Время простоя (число импульсов), уста-новленное для CH1 (Y2, Y3). Функция не работает при установленном значении ≦0.	○	○	○	○	0	-	-	R/W	NO	0
D1249	Установленное значение для COM1 (RS-232) паузы приема данныхt (шаг:	○	○	○	○	0	-	-	R/W	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
	1мс, мин. 50мс, при меньшем значении применяется 50мс) (только для команд MODRW/RS) В команде RS при установленном значении «0», паузы не происходит.										
D1250	COM1 (RS-232) код ошибки связи (только для команд MODRW/RS)	○	○	○	○	0	-	-	R/W	NO	0
D1252	Установленное значение для COM3 (RS-485) паузы приема данных (шаг: 1мс, мин. 50мс, при меньшем значении применяется 50мс) (только для команд MODRW/RS) В команде RS при установленном значении «0», паузы не происходит.	○	×	○	×	50	-	-	R/W	NO	50
D1253	COM3 (RS-485) код ошибки связи (только для команд MODRW/RS)	○	×	○	×	0	-	-	R/W	NO	0
D1255*	COM3 (RS-485) Коммуникационный адрес ПЛК	○	×	○	○	50	-	-	R/W	YES	1
D1256 ↓ D1295	Для: COM2, RS-485, команда MODRW. D1256~D1295 сохраняют принятые данные.	○	○	○	○	0	-	-	R	NO	0
D1296 ↓ D1311	Для: COM2, RS-485, команда MODRW. D1296~D1311 сохраняют преобразованные Hex данные из D1070 ~ D1085 (формат ASCII). ПЛК автоматически преобразует принятые ASCII -данные из D1070 ~ D1085 в Hex данные.	○	○	○	○	0	-	-	R	NO	0
D1312*	Указанное число доп. импульсов для доп. импульсных выходов и поиска Z-фазы командой ZRN (применяется вместе с M1308). M1308)	○	×	○	○	0	0	-	R/W	NO	0
D1313*	Секунды часов реального времени (RTC): 00 ~ 59	○	○	○	○	-	-	-	R/W	YES	0
D1314*	Минуты RTC: 00 ~ 59	○	○	○	○	-	-	-	R/W	YES	0
D1315*	Часы RTC: 00 ~ 23	○	○	○	○	-	-	-	R/W	YES	0
D1316*	Дни RTC: 01 ~ 31	○	○	○	○	-	-	-	R/W	YES	1
D1317*	Месяцы RTC: 01 ~ 12	○	○	○	○	-	-	-	R/W	YES	1
D1318*	Дни недели RTC: 1 ~ 7	○	○	○	○	-	-	-	R/W	YES	2
D1319*	Годы RTC: 00 ~ 99 (A.D.)	○	○	○	○	-	-	-	R/W	YES	8
D1320*	ID 1го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1321*	ID 2го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1322*	ID 3го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1323*	ID 4го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1324*	ID 5го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1325*	ID 6го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1326*	ID 7го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0
D1327*	ID 8го правостороннего модуля	○	×	×	×	0	-	-	R	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1336	Текущее значение (PV) импульсного выхода Y2 (младшее слово)	○	○	○	○	-	-	-	R/W	YES	0
D1337	PV импульсного выхода Y2 (старшее слово)	○	○	○	○	-	-	-	R/W	YES	0
D1338	PV импульсного выхода Y3 (младшее слово)	○	○	○	○	-	-	-	R/W	NO	0
D1339	PV импульсного выхода Y3 (старшее слово)	○	○	○	○	-	-	-	R/W	NO	0
D1340	Начальная/конечная частота 1й группы импульсных выходов CH0 (Y0, Y1)	○	○	○	○	100	-	-	R/W	NO	100
D1343	Время рамповой функции 1й группы импульсных выходов CH0 (Y0, Y1)	○	○	○	○	100	-	-	R/W	NO	100
D1348*	Когда M1534 = ON, D1348 сохраняет время рамповой функции CH0(Y0, Y1)	○	○	○	○	100	-	-	R/W	NO	100
D1349*	Когда M1535 = ON, D1349 сохраняет время рамповой функции CH1(Y2, Y3)	○	○	○	○	100	-	-	R/W	NO	100
D1352	Начальная/конечная частота 2й группы импульсных выходов CH1 (Y2, Y3)	○	○	○	○	100	-	-	R/W	NO	100
D1353	Время рамповой функции 2й группы импульсных выходов CH1 (Y2, Y3)	○	○	○	○	100	-	-	R/W	NO	100
D1354	Цикл сканирования PLC LINK (шаг: 1мс) ■ Макс: K32000 ■ D1354 = K0 когда PLC LINK останавливается или после выполнения 1-го цикла	○	○	○	○	0	0	0	R	NO	0
D1355*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#1	○	○	○	○	-	-	-	R/W	YES	H'1064
D1356*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#2	○	○	○	○	-	-	-	R/W	YES	H'1064
D1357*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#3	○	○	○	○	-	-	-	R/W	YES	H'1064
D1358*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#4	○	○	○	○	-	-	-	R/W	YES	H'1064
D1359*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#5	○	○	○	○	-	-	-	R/W	YES	H'1064
D1360*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#6	○	○	○	○	-	-	-	R/W	YES	H'1064
D1361*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#7	○	○	○	○	-	-	-	R/W	YES	H'1064
D1362*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#8	○	○	○	○	-	-	-	R/W	YES	H'1064
D1363*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#9	○	○	○	○	-	-	-	R/W	YES	H'1064
D1364*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#10	○	○	○	○	-	-	-	R/W	YES	H'1064
D1365*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#11	○	○	○	○	-	-	-	R/W	YES	H'1064
D1366*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#12	○	○	○	○	-	-	-	R/W	YES	H'1064
D1367*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#13	○	○	○	○	-	-	-	R/W	YES	H'1064

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1368*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#14	○	○	○	○	-	-	-	R/W	YES	H'1064
D1369*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#15	○	○	○	○	-	-	-	R/W	YES	H'1064
D1370*	Начальный адрес в Master-режиме для чтения от Slave-устройства ID#16	○	○	○	○	-	-	-	R/W	YES	H'1064
D1386	ID 1го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1387	ID 2го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1388	ID 3го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1389	ID 4го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1390	ID 5го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1391	ID 6го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1392	ID 7го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1393	ID 8го левостороннего модуля	×	×	○	○	0	-	-	R	NO	0
D1399*	Стартовый ID назначенного Slave-устройства для PLC LINK	○	○	○	○	-	-	-	R/W	YES	1
D1415*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#1	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1416*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#2	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1417*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#3	○	○	○	○	-	-	-	R/W	YES	10C8
D1418*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#4	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1419*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#5	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1420*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#6	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1421*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#7	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1422*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#8	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1423*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#9	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1424*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#10	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1425*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#11	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1426*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#12	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1427*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#13	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1428*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#14	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1429*	Начальный адрес в Master-режиме для записи в Slave-устройство ID#15	○	○	○	○	-	-	-	R/W	YES	H'10C8
D1430*	Начальный адрес в Master-режиме для	○	○	○	○	-	-	-	R/W	YES	H'10C8

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
	записи в Slave-устройство ID#16										
D1431*	Время цикла опроса PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1432*	Текущее время цикла опроса PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1433*	Количество ведомых устройств EASY PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1434*	Размер читаемых данных в Slave-режиме ID#1	○	○	○	○	-	-	-	R/W	YES	16
D1435*	Размер читаемых данных в Slave-режиме ID#2	○	○	○	○	-	-	-	R/W	YES	16
D1436*	Размер читаемых данных в Slave-режиме ID#3	○	○	○	○	-	-	-	R/W	YES	16
D1437*	Размер читаемых данных в Slave-режиме ID#4	○	○	○	○	-	-	-	R/W	YES	16
D1438*	Размер читаемых данных в Slave-режиме ID#5	○	○	○	○	-	-	-	R/W	YES	16
D1439*	Размер читаемых данных в Slave-режиме ID#6	○	○	○	○	-	-	-	R/W	YES	16
D1440*	Размер читаемых данных в Slave-режиме ID#7	○	○	○	○	-	-	-	R/W	YES	16
D1441*	Размер читаемых данных в Slave-режиме ID#8	○	○	○	○	-	-	-	R/W	YES	16
D1442*	Размер читаемых данных в Slave-режиме ID#9	○	○	○	○	-	-	-	R/W	YES	16
D1443*	Размер читаемых данных в Slave-режиме ID#10	○	○	○	○	-	-	-	R/W	YES	16
D1444*	Размер читаемых данных в Slave-режиме ID#11	○	○	○	○	-	-	-	R/W	YES	16
D1445*	Размер читаемых данных в Slave-режиме ID#12	○	○	○	○	-	-	-	R/W	YES	16
D1446*	Размер читаемых данных в Slave-режиме ID#13	○	○	○	○	-	-	-	R/W	YES	16
D1447*	Размер читаемых данных в Slave-режиме ID#14	○	○	○	○	-	-	-	R/W	YES	16
D1448*	Размер читаемых данных в Slave-режиме ID#15	○	○	○	○	-	-	-	R/W	YES	16
D1449*	Размер читаемых данных в Slave-режиме ID#16	○	○	○	○	-	-	-	R/W	YES	16
D1450*	Размер записываемых данных в Slave-режиме ID#1	○	○	○	○	-	-	-	R/W	YES	16
D1451*	Размер записываемых данных в Slave-режиме ID#2	○	○	○	○	-	-	-	R/W	YES	16
D1452*	Размер записываемых данных в Slave-режиме ID#3	○	○	○	○	-	-	-	R/W	YES	16
D1453*	Размер записываемых данных в Slave-режиме ID#4	○	○	○	○	-	-	-	R/W	YES	16
D1454*	Размер записываемых данных в Slave-режиме ID#5	○	○	○	○	-	-	-	R/W	YES	16
D1455*	Размер записываемых данных в Slave-режиме ID#6	○	○	○	○	-	-	-	R/W	YES	16

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1456*	Размер записываемых данных в Slave-режиме ID#7	○	○	○	○	-	-	-	R/W	YES	16
D1457*	Размер записываемых данных в Slave-режиме ID#8	○	○	○	○	-	-	-	R/W	YES	16
D1458*	Размер записываемых данных в Slave-режиме ID#9	○	○	○	○	-	-	-	R/W	YES	16
D1459*	Размер записываемых данных в Slave-режиме ID#10	○	○	○	○	-	-	-	R/W	YES	16
D1460*	Размер записываемых данных в Slave-режиме ID#11	○	○	○	○	-	-	-	R/W	YES	16
D1461*	Размер записываемых данных в Slave-режиме ID#12	○	○	○	○	-	-	-	R/W	YES	16
D1462*	Размер записываемых данных в Slave-режиме ID#13	○	○	○	○	-	-	-	R/W	YES	16
D1463*	Размер записываемых данных в Slave-режиме ID#14	○	○	○	○	-	-	-	R/W	YES	16
D1464*	Размер записываемых данных в Slave-режиме ID#15	○	○	○	○	-	-	-	R/W	YES	16
D1465*	Размер записываемых данных в Slave-режиме ID#16	○	○	○	○	-	-	-	R/W	YES	16
D1480* ↓ D1495*	Считываемые данные в Slave-режиме ID#1 в PLC LINK при M1353 = OFF	○	○	○	○	0	-	-	R	NO	0
	Исходные регистры считываемых данных вSlave-режиме ID#1~ID#16 в PLC LINK, сохраняются, когда M1353= ON	○	○	○	○	-	-	-	R	YES	0
D1496* ↓ D1511*	Записываемые данные в Slave-режиме ID#1 в PLC LINK при M1353 = OFF	○	○	○	○	0	-	-	R/W	NO	0
	Исходные регистры записываемых данных вSlave-режиме ID#1~ID#16 в PLC LINK, сохраняются, когда M1353= ON	○	○	○	○	-	-	-	R/W	YES	0
D1512* ↓ D1527*	Данные, считываемые в Slave-режиме ID#2 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1528* ↓ D1543*	Данные, записываемые в Slave-режиме ID#2 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1544* ↓ D1559*	Данные, считываемые в Slave-режиме ID#3 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1560* ↓ D1575*	Данные, записываемые в Slave-режиме ID#3 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1576* ↓ D1591*	Данные, считываемые в Slave-режиме ID#4 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1592* ↓	Данные, записываемые в Slave-режиме ID#4 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1607*											
D1608* ↓ D1623*	Данные, считываемые в Slave-режиме ID#5 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1624* ↓ D1639*	Данные, записываемые в Slave-режиме ID#5 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1640* ↓ D1655*	Данные, считываемые в Slave-режиме ID#6 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1656* ↓ D1671*	Данные, записываемые в Slave-режиме ID#6 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1672* ↓ D1687*	Данные, считываемые в Slave-режиме ID#7 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1688* ↓ D1703*	Данные, записываемые в Slave-режиме ID#7 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1704* ↓ D1719*	Данные, считываемые в Slave-режиме ID#8 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1720* ↓ D1735*	Данные, записываемые в Slave-режиме ID#8 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1736* ↓ D1751*	Данные, считываемые в Slave-режиме ID#9 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1752* ↓ D1767*	Данные, записываемые в Slave-режиме ID#9 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1768* ↓ D1783*	Данные, считываемые в Slave-режиме ID#10 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1784* ↓ D1799*	Данные, записываемые в Slave-режиме ID#10 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1800* ↓ D1815*	Данные, считываемые в Slave-режиме ID#11 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1816* ↓ D1831*	Данные, записываемые в Slave-режиме ID#11 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0

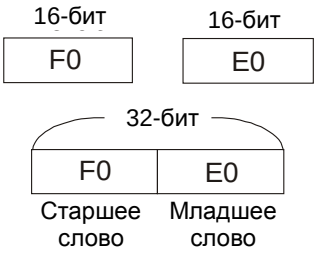
Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолча-нию
D1832* ↓ D1847*	Данные, считываемые в Slave-режиме ID#12 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1848* ↓ D1863*	Данные, записываемые в Slave-режиме ID#12 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1864* ↓ D1879*	Данные, считываемые в Slave-режиме ID#13 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1880* ↓ D1895*	Данные, записываемые в Slave-режиме ID#13 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1896* ↓ D1911*	Данные, считываемые в Slave-режиме ID#14 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1900* ↓ D1931*	Указанный номер Slave-станции для PLC Link, когда M1356 = ON. Последо-вательные станции, установленные в D1399, в этом случае недействительны. Регистры фиксируются только при M1356 = ON.	○	×	○	○	0	-	-	R/W	NO	
D1912* ↓ D1927*	Данные, записываемые в Slave-режиме ID#14 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1928* ↓ D1943*	Данные, считываемые в Slave-режиме ID#15 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1944* ↓ D1959*	Данные, записываемые в Slave-режиме ID#15 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1960* ↓ D1975*	Данные, считываемые в Slave-режиме ID#16 в PLC LINK	○	○	○	○	0	-	-	R	NO	0
D1976* ↓ D1991*	Данные, записываемые в Slave-режиме ID#16 в PLC LINK	○	○	○	○	0	-	-	R/W	NO	0
D1994	Оставшееся время для установленного в ПЛК пароля для DVP-PCC01	○	○	○	○	0					
D1995	Размер данных для установок ID ПЛК для DVP-PCC01	○	○	○	○	0	-	-	R/W	NO	0
D1996	1^{e} слово ID установок для DVP-PCC01 (отображаются в Hex формате, соответствующем ASCII кодам)	○	○	○	○	0	-	-	R/W	NO	0
D1997	2^{e} слово ID установок для DVP-PCC01 (отображаются в Hex формате, соответствующем ASCII кодам)	○	○	○	○	0	-	-	R/W	NO	0
D1998	3^{e} слово ID установок для DVP-PCC01	○	○	○	○	0	-	-	R/W	NO	0

Спец. D	Содержимое регистра	ES2 EX2	SS2	SA2 / SE	SX2	OFF ⇩ ON	STOP ⇩ RUN	RUN ⇩ STOP	Атрибут	Энерго незав.	По умолчанию
	(отображаются в Hex формате, соответствующем ASCII кодам)										
D1999	4e слово ID установок для DVP-PCC01 (отображаются в Hex формате, соответствующем ASCII кодам)	○	○	○	○	0	-	-	R/W	NO	0
D9800~ D9879	Для левосторонних специальных модулей ввода/вывода для SA2/SX2/SE	×	×	○	○	-	-	-	R/W	NO	0
D9900~ D9979	Для специальных модулей ввода/вывода для ES2/EX2. (См. **DVP-PLC Инструкция по эксплуатации - модули**) Для правосторонних специальных модулей ввода/вывода для SA2/SX2/SE	○	×	○	○	-	-	-	R/W	NO	0
D9980	Код сообщения о статусе CANopen (только для DVP-ES2-C)	○	×	×	×	-	-	-	R	NO	0
D9981~ D9996	Kodы сообщения о статусе CANopen для Ведомых 1-16 (только для DVP-ES2-C)	○	×	×	×	-	-	-	R	NO	0
	Биты 0-15, определяющие Ведомых 1-16. При включенном бите возникает ошибка.	○	×	×	×	-	-	-	R	NO	0

2.14 Индексные регистры E, F

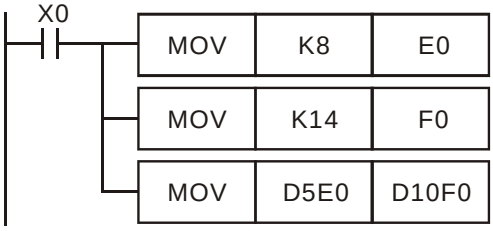
Индексные регистры применяются для изменения операндов (слово, двойное слово) путем добавления индекса. Изменяемые операнды могут быть байтовыми (KnX, KnY, KnM, KnS, T, C, D) и битовыми (X, Y, M, S). Регистры E, F не могут изменять константы (K, H). Неиспользуемые индексные регистры могут использоваться в качестве общих регистров.

Индексные регистры [E], [F]
Индексные регистры являются 16-битными, они могут читаться и записываться. 16 точек обозначаются как E0~E7 и F0~F7. Если необходим 32-битный регистр, адресом назначения указывается индексный регистр E. F в этом случае не используется.



При назначении 32-битного регистра применяется комбинация E и F:
(E0, F0) , (E1, F1) (E2, F2) (E3, F3) (E4, F4) , (E5, F5) (E6, F6) (E7, F7)

Пример:



Когда X0 включен и E0 = 8, F0 = 14, D5E0 = D(5+8) = D13, D10F0 = D(10+14) = D24, содержимое в D13 перемещается в D24.

2.15 Указатель уровня вложения[N], Указатель[P], Указатель прерывания [I]

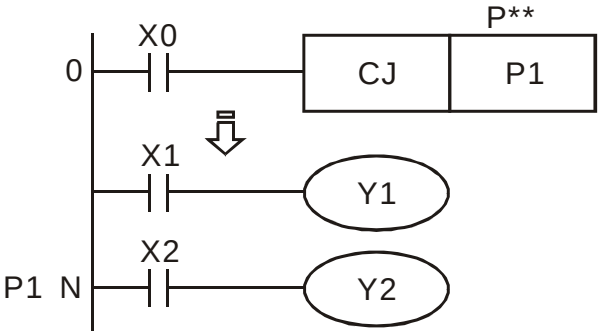
Указатель	N	Управление вложениями (мастер-контроль)		N0~N7, 8 точек	Управляющие вложениями точки
	P	Для команд CJ, CALL		P0~P255, 256 точек	Метки CJ, CALL
Указатель	I	Для прерывания	Внешний прерыватель	I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7), 8 points (01, включение по переднему фронту ⌐↑, 00, включение по заднему фронту ↓¬)	Метки подпрограмм прерывания.
			Таймер-прерыватель (прерывание по времени)	I602/I699, I702/I799, 2 точки (разрешение таймера=1мс)	
			Высокоскоростной счетчик - прерыватель	I010, I020, I030, I040, I050, I060, I070, I080, 8 точек	
			Коммуникационный прерыватель	I140(COM1: RS232), I150(COM2: RS-485), I160(COM3: RS-485), 3 точки	

Указатель уровня вложения N: применяется с командами MC и MCR. MC – стартовая команда мастер-контроля. После выполнения команды MC команды между MC и MCR выполняются в обычном режиме. Команды мастер-контроля MC-MCR управляют структурой вложений и максимально возможно 8 уровней вложений, они нумеруются как N0 … N7.

Указатель P: применяется с командами CJ, CALL и SRET.

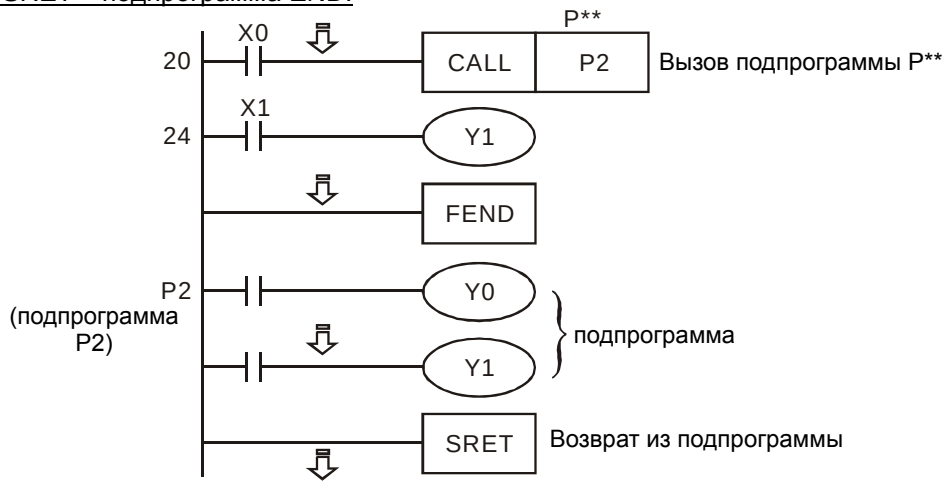
CJ – условный переход:

Когда X0 включен, программа перейдет к выполнению строки N (обозначенной указателем P1) и команды между 0 и N выполняться не будут.
Когда X0 выключен, программа выполняется обычно последовательно с 0.



CALL – подпрограмма, SRET – подпрограмма END:

Когда X0 включен, программа переходит к P2 для выполнения подпрограммы. После выполнения команды SRET программа возвращается к строке 24 и продолжает выполнение.



Указатель прерывания I: применяется с командами API 04 EI, API 05 DI, API 03 IRET. Существует 4 типа указателей прерывания. Для вставки прерывателя необходимо объединить команды EI (включение прерывания), DI (выключение прерывания) и IRET (возврат прерывания).

1. Внешний прерыватель

- Когда выходной сигнал на выходах X0~X7 срабатывает по переднему или заднему фронту, выполнение программы будет прервано и программа перейдет к подпрограмме прерывания, обозначенной указателем I000/I001(X0), I100/I101(X1), I200/I201(X2), I300/I301(X3), I400/I401(X4), I500/I501(X5), I600/I601(X6), I700/I701(X7). После выполнения команды IRET программа возвращается в точку до прерывания.
- Когда X0 (C243) работает с I100/I101 (X1), X0/X1 (C246, C248, C252) работает с I400/I401, значение C243, C246, C248, C252 будет сохранено в (D1240, D1241)
- Когда X2 (C244) работает с I300/I301 (X3), X2/X3 (C250, C254) работает с I500/I501, значение C244, C250, C254 будет сохранено в (D1242, D1243).

2. Прерывание по времени

ПЛК автоматически прерывает выполнение программы через каждый назначенный отрезок времени (2 мс~99 мс) и переходит к выполнению назначенной подпрограммы прерывания.

3. Счетчик-прерыватель

Высокоскоростной счетчик работает с командой сравнения API 53 DHSCS, прерывая выполнение программы при достижении в счетчике заданного значения, и переходит к выполнению подпрограммы прерывания, помеченной прерывателями I010, I020, I030, I040, I050 ,I060, I070, I080.

4. Коммуникационный прерыватель

I140: Команда связи RS (COM1 RS-232) отправляет запрос на прерывание при получении определенных данных. Прерыватель I140 и заданные данные (определенные данные для прерывания) хранятся в младшем байте регистра D1167.

Эта функция работает при приеме ПЛК данных различной длины. Установите специальное конечное слово в in D1167 и напишите подпрограмму прерывания I140. Когда ПЛК получит конечное слово, будет выполняться подпрограмма I140.

I150: Команда связи RS (COM2 RS-485) отправляет запрос на прерывание при получении определенных данных. Прерыватель I150 и заданные данные (определенные данные для прерывания) хранятся в младшем байте регистра D1168.
Эта функция работает при приеме ПЛК данных различной длины. Установите специальное конечное слово в in D1168 и напишите подпрограмму прерывания I150. Когда ПЛК получит конечное слово, будет выполняться подпрограмма I150.

I160: Команда связи RS (COM3 RS-485) отправляет запрос на прерывание при получении определенных данных. Прерыватель I160 и заданные данные (определенные данные для прерывания) хранятся в младшем байте регистра D1169.
Эта функция работает при приеме ПЛК данных различной длины. Установите специальное конечное слово в in D1169 и напишите подпрограмму прерывания I160. Когда ПЛК получит конечное слово, будет выполняться подпрограмма I160.

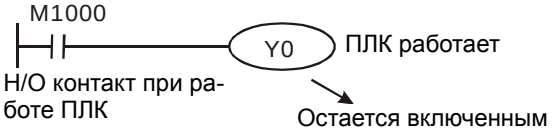
2.16 Описания специальных реле M и регистров D

Функциональная группа Флаги ПЛК

Номер M1000~M1003

Описание:
Эти реле дают информацию о ходе работы ПЛК.

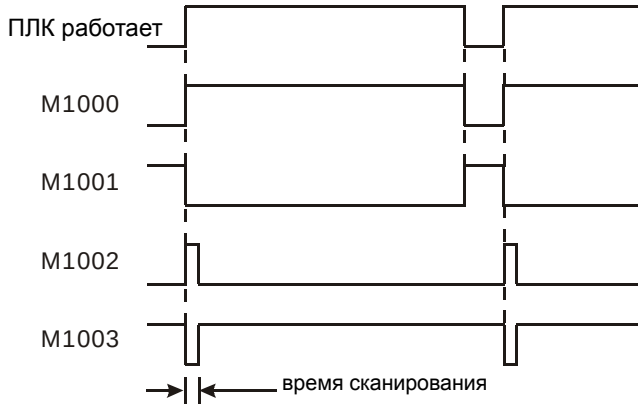
M1000:
Н/О контакт для индикации состояния ПЛК. M1000 замкнут, когда ПЛК работает.



M1001:
Н/З контакт для индикации состояния ПЛК. M1001 разомкнут, когда ПЛК работает.

M1002:
Включается один раз при первом сканировании, когда ПЛК начинает работу. Используется для инициализации регистров, выходов или счетчиков при запуске ПЛК.

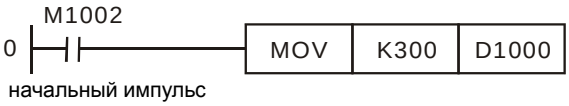
M1003:
Выключается один раз при первом сканировании, когда ПЛК начинает работу. Используется для инициализации регистров, выходов или счетчиков при запуске ПЛК.



Функциональная группа Сторожевой таймер

Номер D1000

Описание:
1. Используется для контроля времени сканирования ПЛК. Когда время сканирования превышает заданное значение (SV) в сторожевом таймере, включается красный светодиод ERROR LED и все выходы отключаются.
2. По умолчанию значение сторожевого монитора равно 200 мс. При длинной или сложной операции, команда MOV может изменять значение SV. Для примера SV = 300 мс.



3. Максимально значение SV равно 32 767 мс. Но следует соблюдать осторожность при регулировке SV. Если SV в D1000 слишком велико, поиск ошибок выполнения операций может сильно затягиваться. Рекомендуется устанавливать SV короче 200 мс.
4. Время сканирования может быть увеличено из-за сложности программы или большого количества подключенных модулей ввода/вывода. Проверьте D1010 ~ D1012, если время сканирования превышает заданное значение (SV) в D1000. Кроме изменений SV в D1000, можно применить команду WDT (API 07). Когда программа начнет выполнять команду WDT, внутренний сторожевой таймер будет сброшен и, следовательно, значение SV не будет превышено.

Функциональная группа Размер программы

Номер D1002

Описание:
Регистр содержит допустимый размер программы ПЛК.
SS2: 7 920 шагов (слова)
ES2 / EX2 / SA2 / SX2: 15 872 шага (слова)

Функциональная группа Проверка синтаксиса в программе

Номер M1004, D1004, D1137

Описание:
1. При обнаружении синтаксической ошибки, мигает светодиодный индикатор ERROR и включается специальное реле M1004.
2. Проверка синтаксиса проводится:
   a) При включении питания ПЛК.

b) Когда WPLSoft загружает программу в ПЛК.
c) При онлайн редактировании WPLSoft.

3. Бывают ошибки параметров или грамматические ошибки. Код ошибки фиксируется в D1004. Адрес, по которому возникла ошибка, фиксируется в D1137. В случае циклической ошибки значение D1137 недействительно.
4. Коды ошибок см. раздел 6.2 Таблица кодов ошибок.

Функциональная группа Сторожевой таймер

Номер M1008, D1008

Описание:

1. Когда сканирование прерывается во время исполнения, включается светодиодный индикатор ERROR и также включается M1008.
2. D1008 сохраняет шаг сканирования, на котором произошло прерывание сканирования.

Функциональная группа Мониторинг времени сканирования

Номер D1010~D1012

Описание:

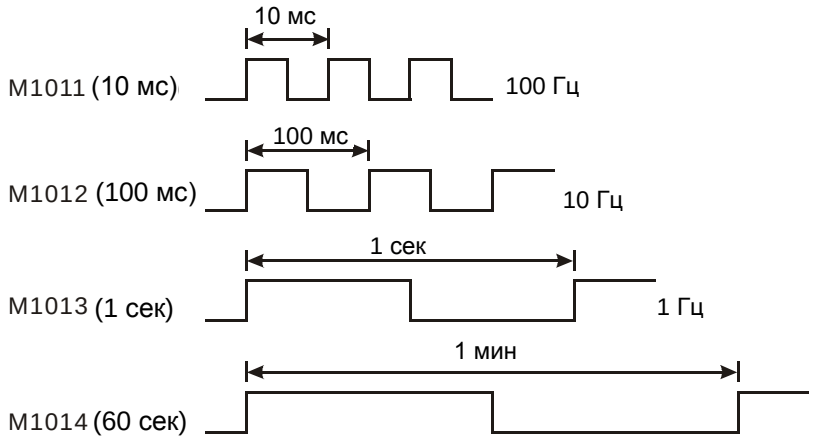
Текущее, минимальное и максимальное значение времени сканирования сохраняется в D1010 ~ D1012.
D1010: текущее, **D1011:** минимальное, **D1012:** максимальное

Функциональная группа Внутренние тактовые импульсы

Номер M1011~M1014

Описание:

1. ПЛК может выдавать в программе четыре вида тактовых импульсов. При включении ПЛК четыре тактовых импульса выдаются автоматически.



2. Тактовые импульсы выдаются даже при остановке ПЛК. При повторном запуске ПЛК тактовые импульсы не синхронизируются с повторным запуском.

Функциональная группа Высокоскоростной таймер

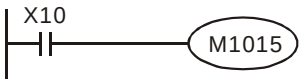
Номер M1015, D1015

Описание:

1. Когда M1015 включен, высокоскоростной таймер D1015 будет активирован при выполнении команды END в текущем скане. Минимальное значение D1015 = 100 мкс.
2. Диапазон D1015 0~32 767. При достижении значения 32 767 счет начинается с 0.
3. Когда M1015 выключен, D1015 немедленно прекращает счет времени.

Пример:

1. Когда X10 включен, M1015 включается и запускает высокоскоростной таймер, текущее значение которого сохраняется в D1015.
2. Когда X10 выключен, M1015 выключен. Таймер выключается.



Функциональная группа Часы реального времени (RTC)

Номер M1016~M1017, D1313~D1319

Описание:

1. Специальные M и D, относящиеся к RTC

Реле/регистр	Наименование	Функция
M1016	Отображение года	OFF: отображаются 2 последние цифры (напр. 12) ON: отображаются 2 последние цифры + 2 000 (напр. 2012)
M1017	Коррекция ±30 секунд (т.е. с точностью до 1 минуты)	0 ~ 29 сек: минуты остаются; секунды сбрасываются до 0 30~ 59 сек: минуты увеличиваются на 1; секунды сбрасываются до 0
D1313	Секунды	0~59
D1314	Минуты	0~59
D1315	Часы	0~23
D1316	Дни	1~31
D1317	Месяцы	1~12
D1318	Дни недели	1~7
D1319	Годы	0 ~ 99 (2 последние цифры)

2. При некорректной установке значения RTC время будет сброшено до следующих показателей: секунды→0, минуты→0, часы→0, дни→1, месяцы→1, дни недели→1, годы →0.
3. Для SS2 – функция работает при включенном питании. Память RTC энергонезависимая. RTC возобновит значение времени после повторного включения питания. Для точности RTC желательно провести калибровку после восстановления подачи питания.
4. Для SA2 V1.0 и ES2/EX2/SX2 V2.0 значение RTC сохраняется в течение 1-2 недель после выключения. Поэтому при включении позже этого срока, установите RTC заново.
5. Методы изменения RTC:
   a) Применяя команду TWR.
   b) Применяя периферийное устройство или ПО WPLSoft.

Функциональная группа π

Номер D1018~D1019

Описание:

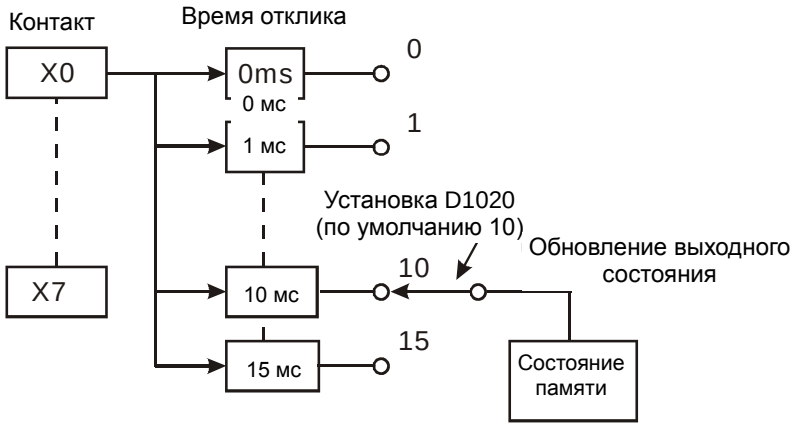
1. Число π сохраняется в двух регистрах D1018 и D1019 в виде 32-битных данных с плавающей запятой.
2. Шестнадцатеричное значение = H 40490FDB.

Функциональная группа Настройка времени отклика на входах

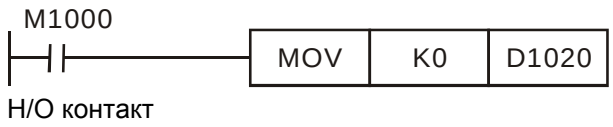
Номер D1020

Описание:

1. D1020 может быть использован для установки времени отклика на входные сигналы от X0 ~X7 для серии ES2. По умолчанию: 10 мс, 0~20 мс – регулируемый диапазон.
2. При включении ПЛК в D1020 автоматически запишется значение 10.



3. Если время отклика для X0 ~ X7 установлено как 0 мс, оно будет зависеть только от быстродействия RC-фильтров и в этом случае скорость отклика будет минимальной – 50 мкс.



H/O контакт

4. Нет необходимости корректировать время отклика, если применяются высокоскоростные счетчики и прерыватели.
5. Для корректировки времени отклика в D1020 можно использовать команду API 51 REFF.

Функциональная группа Функция обнаружения ширины импульса для X6

Номер M1083,M1084, D1023

Содержимое:

Когда M1084 включен, функция обнаружения запускается, и обнаруженная ширина импульсов сохраняется в D1023 (шаг: 0.1 мс)

M1083 вкл：обнаруживается ширина отрицательного полуимпульса (OFF→ON)

M1083 выкл：обнаруживается ширина положительного полуимпульса (ON→OFF)

Функциональная группа Код ошибки связи

Номер M1025, D1025

Описание:

В процессе связи между ПЛК и PC/HMI, M1025 включается, когда ПЛК получает некорректный запрос коммуникации во время обмена данными. Код ошибки сохраняется в D1025.

01: недопустимый код команды
02: недопустимый адрес устройства
03: запрашиваемые данные превышают допустимый диапазон
07: ошибка контрольной суммы

Функциональная группа Функция маркировки и маскировки импульсного выхода

Номер M1108, M1110, M1156, M1158, M1538, M1540, D1026, D1027, D1135, D1136, D1232, D1233, D1234, D1235, D1348, D1349

Описание:

См. описания команд API 59 PLSR / API 158 DDRVI / API 197 DCLLM.

Функциональная группа Флаги завершения выполнения

Номер M1029, M1030, M1102, M1103

Описание:

Флаги завершения выполнения:

MTR, HKY, DSW, SEGL, PR:

M1029 включается в цикле сканирования по завершении выполнения вышеперечисленных команд.

PLSY, PLSR:

1. M1029 включен, когда на выходе Y0 завершена генерация импульсов.

2. M1030 включен, когда на выходе Y1 завершена генерация импульсов.
3. M1102 включен, когда на выходе Y2 завершена генерация импульсов.
4. M1103 включен, когда на выходе Y3 завершена генерация импульсов.
5. Флаги M1029 и M1030 сбрасываются вручную.

INCD:
M1029 включается в цикле сканирования по завершении сравнения групп данных

RAMP, SORT:
1. M1029 включается по завершении выполнения команд.
2. Если команда не выполняется, флаг не включается.

DABSR:
1. M1029 включается по завершении выполнения команды.
2. При повторном выполнении команды, M1029 выключится и затем снова включится по завершении выполнения команды.

ZRN, DRVI, DRVA:
1. M1029 включится после окончания генерации импульсов на выходах Y0 и Y1. M1102 включится после окончания генерации импульсов на выходах Y2 и Y3.
2. При повторном выполнении команды, M1029/M1102 выключатся и затем снова включатся по завершении выполнения команд.

Функциональная группа Команды очистки памяти

Номер M1031, M1032

Описание:
M1031 (очистка энергозависимой памяти), M1032 (очистка энергонезависимой памяти)

Реле	Очищаемые операнды
M1031 Очистка энергозависимой памяти	Состояние контактов Y, M и S общего назначения ■ Регистры и контакты таймеров общего назначения T ■ Регистры и контакты счетчиков общего назначения C ■ Регистры данных общего назначения D
M1032 Очистка энергонезависимой памяти	Состояние контактов энергонезависимых M и S ■ Регистры и контакты аккумулятивных таймеров T ■ Регистры и контакты высокоскоростных энергонезависимых счетчиков C ■ Энергонезависимые регистры данных D

Функциональная группа Сохранение состояния выходов в режиме STOP

Номер M1033

Описание:
Когда M1033 включен, ПЛК зафиксирует состояние выходов даже при остановке работы ПЛК.

Функциональная группа Отключение всех выходов Y

Номер M1034

Описание:
Когда M1034 включен, все выходы выключены независимо от работы программы.

Функциональная группа Переключатель RUN/STOP

Номер M1035

Описание:
Когда M1035 включен, ПЛК использует вход X7 как переключатель режимов RUN/STOP.

Функциональная группа Функции COM-портов

Номер

Порт / Пункт	COM1	COM2	COM3
Формат связи	D1036	D1120	D1109
Настройки связи	M1138	M1120	M1136
ASCII/RTU- режимы	M1139	M1143	M1320
Адрес связи в Slave-режиме	D1121		D1255

Описание:
COM-порты (COM1: RS-232, COM2: RS-485, COM3: RS-485) поддерживают формат связи MODBUS в ASCII/RTU-режимах. Когда выбран RTU-формат, длина данных устанавливается как 8. COM2 и COM3 поддерживают передачу со скоростью 921 кбит/с. COM1, COM2 и COM3 могут работать одновременно.

COM1:
Может использоваться в Master или Slave-режиме. Поддерживает форматы связи ASCII/RTU, скорость передачи данных (1макс. 15 200 бит/с) и переменную длину данных (биты данных, биты четности, стоп-биты). **D1036:** COM1 (RS-232) протокол связи – Master/Slave-режим ПЛК. (b8 - b15 не используются). См. таблицу ниже.

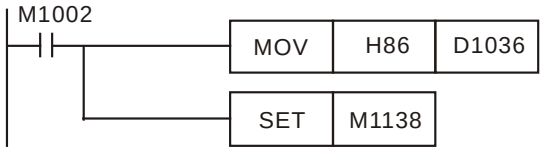
COM2:
Может использоваться в Master или Slave-режиме. Поддерживает форматы связи ASCII/RTU, скорость передачи данных (1макс. 15 200 бит/с) и переменную длину данных (биты данных, биты четности, стоп-биты). **D1120:** COM2 (RS-485) протокол связи – Master/Slave-режим ПЛК. См. таблицу ниже.

COM3:
Может использоваться в Master или Slave-режиме. Поддерживает форматы связи ASCII/RTU, скорость передачи данных (1макс. 15 200 бит/с) и переменную длину данных (биты данных, биты четности, стоп-биты). **D1109:** COM3 (RS-485) протокол связи – Master/Slave-режим ПЛК. (b8 - b15 не используются). См. таблицу ниже.

	Содержимое		
b0	Длина данных	0: 7 биты данных, 1: 8 биты данных (RTU поддерживает только 8 бит данных)	
b1 b2	Биты четности	00: None 01: нечётный 11: четный	
b3	Стоп-биты	0: 1 бит, 1: 2 бита	
b4 b5 b6 b7	Скорость передачи данных	0001(H1): 110 0010(H2): 150 0011(H3): 300 0100(H4): 600 0101(H5): 1200 0110(H6): 2400 0111(H7): 4800 1000(H8): 9600 1001(H9): 19200 1010(HA): 38400 1011(HB): 57600 1100(HC): 115200	
		1101(HD): 500000 (COM2 / COM3)	
		1110(HE): 31250 (COM2 / COM3)	
		1111(HF): 921000 (COM2 / COM3)	
b8	Выбор стартового бита	0: None	1: D1124
b9	Выбор 1ro конечного бита	0: None	1: D1125
b10	Выбор 2ro конечного бита	0: None	1: D1126
b11~b15	Не определены		

Пример 1: Изменение формата связи для COM1

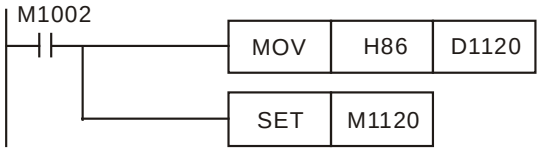
1. Применим следующие команды для изменения формата связи на COM1. Когда ПЛК включается, программа обнаруживает включение M1138 в первом цикле сканирования. Если M1138 включен, программа меняет параметры связи для COM1 в зависимости от значения, установленного в D1036.



2. Измененный формат связи для COM1 в ASCII mode, 9600bps, 7 data bits, проверка на четность, 1 стоп-бит (9600, 7, E, 1).

Пример 2: Изменение формата связи для COM2

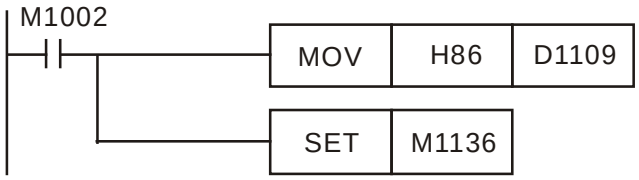
1. Применим следующие команды для изменения формата связи на COM2. Когда ПЛК включается, программа обнаруживает включение M1120 в первом цикле сканирования. Если M1120 включен, программа меняет параметры связи для COM2 в зависимости от значения, установленного в D1120.
2. Измененный формат связи для COM2 в ASCII-режиме, 9600 бит/с, 7 бит данных, проверка на четность, 1 стоп-бит (9600, 7, E, 1)



Пример 3: Изменение формата связи для COM3

1. Применим следующие команды для изменения формата связи на COM3. Когда ПЛК включается, программа обнаруживает включение M1136 в первом цикле сканирования. Если M1136 включен, программа меняет параметры связи для COM3 в зависимости от значения, установленного в D1109.

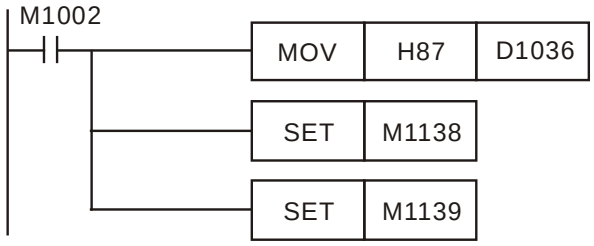
2. Измененный формат связи для COM3 в ASCI -режиме 9600 бит/с, 7 бит данных, проверка на четность, 1 стоп-бит (9600, 7, E, 1).



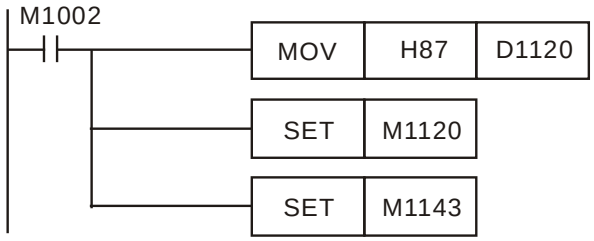
Пример 4: Установка RTU-режима для COM1, COM2, COM3

1. COM1, COM2 и COM3 поддерживают ASCII/RTU-режимы. COM1 определяется флагом M1139, COM2 определяется флагом M1143 и COM3 определяется флагом M1320. Включите эти флаги для установки RTU-режима или выключите - для ASCII-режима.
2. Измененный формат связи для COM1/COM2/COM3 в RTU-режиме, 9600 бит/с, 8 бит данных, проверка четности, 1 стоп-бит (9600, 8, E, 1).

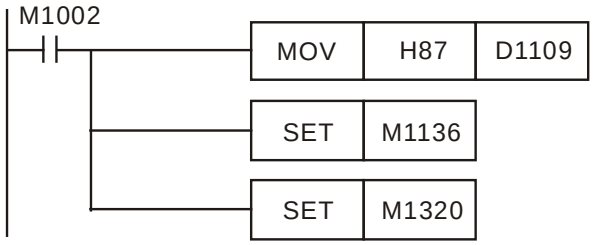
COM1:



COM2:



COM3:



Замечания:

1. Измененный формат связи останется таким же при остановке работы ПЛК.
2. При новом запуске ПЛК, измененный формат связи на COM1~COM3 будет сброшен до состояния формата по умолчанию (9600, 7, E, 1).

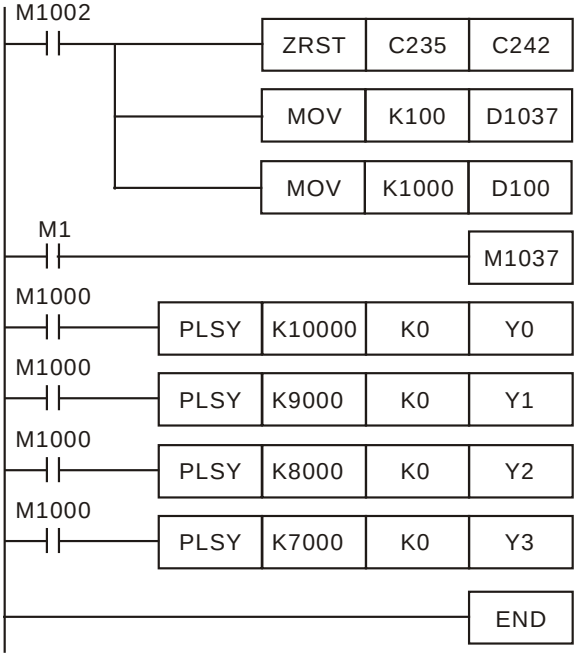
Функциональная группа Включение функции определения скорости (SPD)

Номер M1037, D1037

Описание:

1. M1037 и D1037 могут использоваться для включения 8-шаговой команды SPD. Когда M1037 включен, 8 шагов команды SPD запускаются. При выключенном M1037 функция не работает.
2. Определенная скорость будет сохранена в D1037, т.е. если D1037 = K100, пользователь должен установить значение в D100, показывающее время, необходимое для определения (мс). Кроме того, определенное значение скорости будет храниться в D101 ~ D108 по порядку.

※ Когда функция отключена, C235~C242 будут заняты и недоступны ПЛК.



Функциональная группа Задержка отклика связи

Номер D1038

Описание:

1. Данные времени задержки ответа могут быть установлены, когда ПЛК работает в Slave-режиме связи по COM2, COM3 RS-485. Шаг: 0.1 мс. Диапазон: 0~10 000.
2. При использовании PLC-Link, D1038 может устанавливать задержку выполнения следующей отправки данных. Шаг: 1 цикл сканирования. Диапазон: 0~10 000.

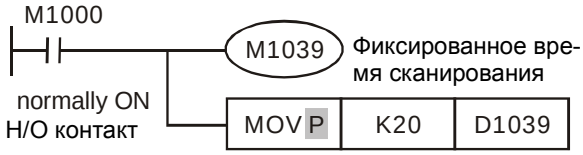
Функциональная группа Фиксированное время сканирования

Номер M1039, D1039

Описание:

1. Когда M1039 включен, время сканирования программы определяется значением в D1039.

2. После выполнения программы, следующее сканирование запустится только после достижения фиксированного времени сканирования. Если значение в D1039 меньше фактического времени сканирования, сканирование осуществится по фактическому времени.



Фиксированное время сканирования = 20 мс

3. Команды, оперирующие временем сканирования - RAMP, HKY, SEGL, ARWS и PR должны использоваться с “фиксированным временем сканирования” или “прерыванием по времени”.
4. Только для команды HKY: время сканирования должно быть не менее 20 мс.
5. Время сканирования, отображаемое в D1010~D1012 также включает фиксированное.

Функциональная группа Функции аналоговых входов/выходов ПЛК

Номер D1062, D1110~D1113, D1116~D1118

Описание:

1. Только для EX2/SX2.
2. Разрядность AD (аналоговых входных) каналов: 12 бит – 20EX2 и 20SX2; 16 бит по напряжению/току – 30EX2; 0.1 ℃ по температуре – 30EX2
3. Аналоговые входные сигналы и соответствующие им цифровые значения:

Режим \ Тип		20EX2/SX2	30EX2
Напряжение	-10 В~+10 В	-2000~+2000	-32000~+32000
	-5 В~+5 В	Не поддерживает	-32000~+32000
	+1 В~+5 В	Не поддерживает	+0~+32000
Ток	-20 мА~+20 мА	-2000~+2000	-32000~+32000
	+4 мА~+20 мА	+0~+2000	+0~+32000
Температура	PT100/PT1000 -180 ℃~+800 ℃	Не поддерживает	-1800~+8000
Температура	NI100/NI1000 -80 ℃ ~ +170 ℃	Не поддерживает	-800~+1700

4. Разрядность DA (аналоговых выходных) каналов: 12 бит
5. Аналоговые выходные сигналы и соответствующие им цифровые значения:

Режим \ Тип		20EX2/SX2	30EX2
Напряжение	-10 В~+10 В	-2000~+2000	-32000~+32000
Ток	+0 мА~+20 мА	+0~+4000	+0~+32000
	+4 мА~+20 мА	+0~+4000	+0~+32000

6. Специальные регистры для аналоговых функций:

Регистр	Функция
D1062	Среднее количество аналоговых входных сигналов через CH0~CH3 для 20EX2/SX2: 1~20, по умолчанию = K2
	Среднее количество аналоговых входных сигналов через CH0~CH2 для 30EX2: 1~15, по умолчанию = K2
D1110	Среднее значение на аналоговом входном канале 0 (AD 0) для EX2/SX2
D1111	Среднее значение на аналоговом входном канале 1 (AD 1) для EX2/SX2
D1112	Среднее значение на аналоговом входном канале 2 (AD 2) для EX2/SX2
D1113	Среднее значение на аналоговом входном канале 3 (AD 3) для 20EX2/SX2 если D1062 включен, среднее значение равно текущему.
	Отображение состояния аналогового выходного канала 30EX2
D1114	Вкл/Выкл AD каналов для 20EX2/SX2, (0: вкл (по умолчанию) / 1: выкл) bit0~bit3 определяют AD0~AD3.
	30EX2 не поддерживает функцию
D1116	Значение на аналоговом выходном канале 0 (DA 0) для EX2/SX2

Регистр	Функция
D1117	Значение на аналоговом выходном канале 1 (DA 1) для 20EX2/SX2
	30EX2 не поддерживает функцию
D1118	Для EX2/SX2: Время аналого-цифрового преобразования. Равно 2 мс при D1118≦2.

Регистр	Функция
D1115	20EX2/SX2: настройки аналогового вх/вых режима (по умолчанию=H'0) bit0~bit5: Выбор режима: напряжение/ток (0: напряжение (по умолчанию); 1: ток;) bit0~bit3: аналоговые входы (AD0~AD3) bit4~bit5: аналоговые выходы (DA0~DA1) bit8~bit 13: токовый режим bit8~bit11: AD0~AD3 (0: -20 мА~20 мА; 1: 4~20 мА) bit12~bit13: DA0~DA1 (0: 0~20 мА; 1: 4~20 мА)
	30EX2: настройки аналогового вх/вых режима (по умолчанию =H'FFFF)

Описание D1113 для 30EX2:

Биты 15~12	Биты 11~8	Биты 7~4	Биты 3~0
Резервные	Состояние аналогового входного канала (AD2)	Состояние аналогового входного канала (AD1)	Состояние аналогового входного канала (AD0)

Состояние аналогового входного канала 30EX2:

Состояние	0x0	0x1	0x2
Описание	Нормальное	Выходит за пределы верхнего/ нижнего диапазона	Температурный датчик отключен

Диапазон выходных значений для аналогового выхода 30EX2:

Режим аналогового выхода		Верхний диапазон	Нижний диапазон
Напряжение	-10~+10 В	+32384	-32384
	-5V~+5 В		
	+1 V~+5 В	+32384	-384
Ток	-20 мА~+20 мА	+32384	-32384
	+4 мА~+20 мА	+32384	-384
Температура	PT100/PT1000	+8100	-1900
	NI100/NI1000	+1800	-900

Описание D1115 для 30EX2:

Биты 15~12	Биты 11~8	Биты 7~4	Биты 3~0
Аналоговый выходной режим DA0	Аналоговый входной режим AD2	Аналоговый входной режим AD1	Аналоговый входной режим AD0

Аналоговые входные режимы для 30EX2:

Код	0x0	0x1	0x2	0x3
Описание	2-проводная PT100	3-проводная NI100	2-проводная PT1000	2-проводная NI1000
Код	0x4	0x5	0x6	0x7
Описание	3-проводная PT100	3-проводная NI100	3-проводная PT1000	3-проводная NI1000
Код	0x8	0x9	0xA	0xB
Описание	Напряжение: -10 В~+10 В	Напряжение: -5 В~+5 В	Напряжение: +1 В~+5 В	Ток: -20 мА~+20 мА
Код	0xC	0xD	0xE	0xF
Описание	Ток: +4 мА~+20 мА	Резервные		Не используются

Аналоговые выходные режимы для 30EX2:

Код	0x0	0x1	0x2	0xF
Описание	Напряжение: -10 В~+10 В	Ток: +0 мА~+20 мА	Ток: +4 мА~+20 мА	Не используются

Пример настроек D1115 для 30EX2:
Если аналоговый входной режим AD0 это 2-проводной NI100, аналоговый входной режим AD1 3-проводной 1000, аналоговый входной режим AD2 – режим по напряжению (+1 В~ +5 В) и аналоговый входной режим DA0 – режим по току (+4 мА ~ +20 мА), установленное значение в D1115 равно H'2A61.

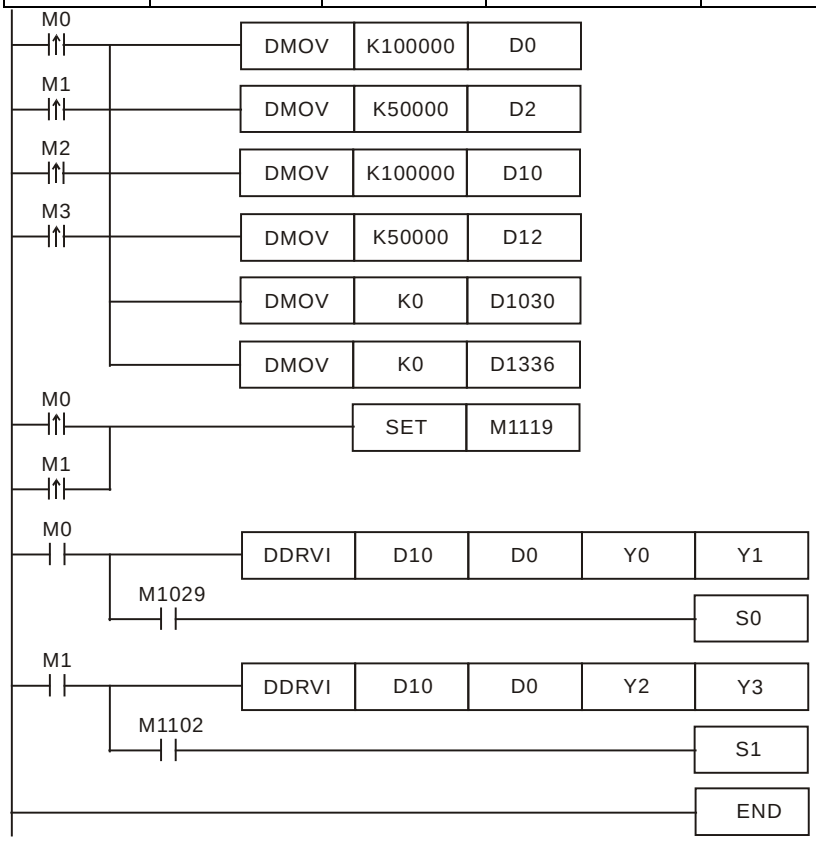
Функциональная группа Включение 2-скоростной выходной функции для команды DDRVI

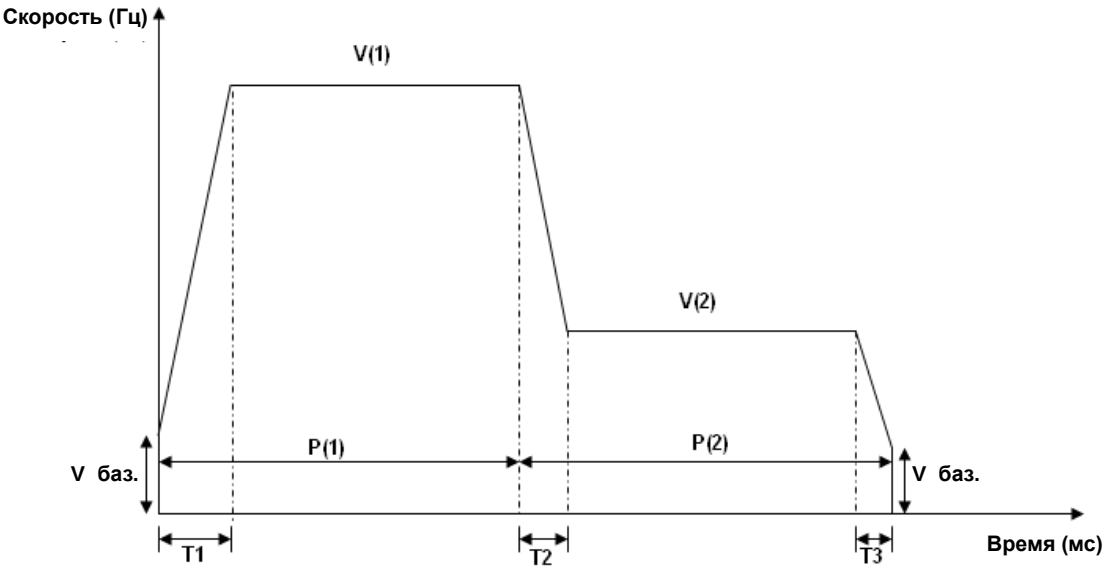
Номер M1119

Описание:
Когда M1119 включен, 2-скоростная выходная функция команды DDRVI включается.
Пример: Предположим D0 (D1) – первая скорость и D2(D3) - вторая. D10(D11) – число выходных импульсов для первой скорости и D12(D13) – для второй.

Пояснения к графику ниже:

Vbase	T1	T2+T3	P(1)	V(1)	P(2)	V(2)
Начальная частота	Время разгона	Время замедления	Позиция первой скорости	Первая скорость	Позиция второй скорости	Вторая скорость





Функциональная группа Ошибки выполнения программы

Номер M1067~M1068, D1067~D1068

Описание:

Операнд	Описание	Энергонезависимость	STOP→RUN	RUN→STOP
M1067	Ошибка выполнения программы	Нет	Очистка	Не измен.
M1068	Фиксация ошибки выполнения	Нет	Не измен.	Не измен.
D1067	Код ошибки выполнения программы	Нет	Очистка	Не измен.
D1068	Адрес ошибки выполнения программы	Нет	Не измен.	Не измен.

Описание кода ошибки:

D1067 код ошибки	Функция
0E18	Ошибка преобразования BCD
0E19	Деление на 0
0E1A	Значение выходит за пределы диапазона (включая изменение индексных регистров E, F)
0E1B	Отрицательный квадратный корень
0E1C	Ошибка команд коммуникации FROM/TO

Функциональная группа Индикация подключения модулей ввода/вывода

Номер D1140, D1142, D1143, D1145

Описание:

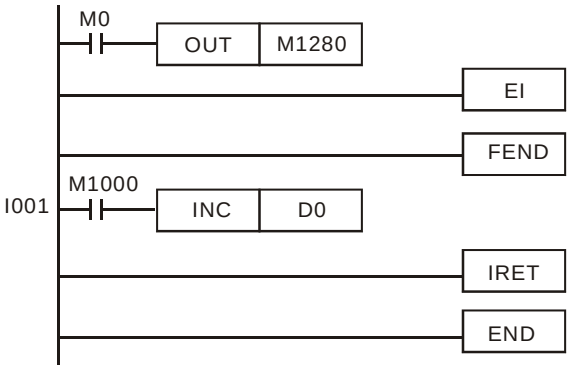
D1140: Число правосторонних модулей (AIO, PT, TC и т.д.), макс. 8 модулей.
D1142: Число точек ввода (X) от модулей.
D1143: Число точек вывода (Y) от модулей.
D1145: Число левосторонних модулей (AIO, PT, TC и т.д.), макс. 8 модулей (только для SA2/SX2).

Функциональная группа Переключение способа прерывания

Номер M1280, M1284, M1286

Описание:

1. Флаг включается перед выполнением команды EI
2. По умолчанию прерыватель I101 (X0) настроен для срабатывания по переднему фронту. Если M1280 включен и выполняется команда EI, PLC ПЛК меняет на срабатывание по заднему фронту. Направление срабатывания X1 снова будет установлено по переднему фронту после сброса M1280.
3. Когда M0 и M1280 выключены, внешний прерыватель X0 срабатывает по переднему фронту.
4. Когда M0 и M1280 включены, внешний прерыватель X0 срабатывает по заднему фронту. Пользователю не надо менять прерыватель I101 на прерыватель I000.



Функциональная группа Сохранение значения высокоскоростного счетчика при операции прерывания

Номер D1240~D1243

Описание:

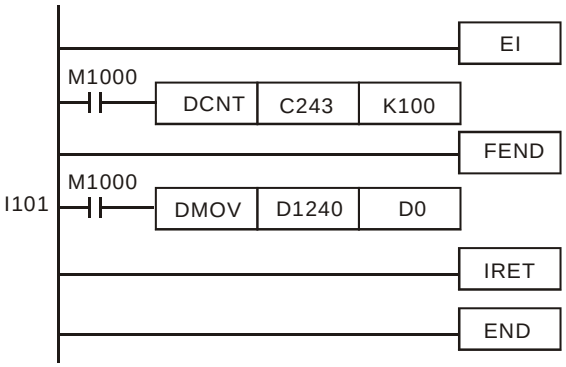
1. Если внешние прерыватели используются на входах для сброса, команды прерывания имеют приоритет на входах. ПЛК будет помещать текущие данные от счетчика в связанные с ним регистры (см. ниже), после чего показания счетчика будут сброшены.

Регистры D	D1241, D1240				D1243, D1242		
Счетчик	C243	C246	C248	C252	C244	C250	C254
Сигнал прерывания	X1(I100/I101)	X4(I400/I401)			X3(I300/I301)	X5(I500/I501)	

2. Порядок работы:
   a) Когда X0 (вход счетчика) и X1 (внешний прерыватель) соответственно работают вместе с C243 и I100/I101, ПЛК помещат значение счетчика в D1241 и D1240.
   b) Когда X0 (вход счетчика) и X4 (внешний прерыватель) соответственно работают вместе с C246, C248, C252 и I400/I401, ПЛК помещат значение счетчика в D1241 и D1240
   c) Когда X2 (вход счетчика) и X3 (внешний прерыватель) соответственно работают вместе с C244, и I300/I301, ПЛК помещат значение счетчика в D1243 и D1242.
   d) Когда X2 (вход счетчика) и X5 (внешний прерыватель) соответственно работают вместе с C250, C254 и I500/I501, ПЛК помещат значение счетчика в D1243 и D1242.

Пример:

Когда внешний прерыватель (X1, I101) срабатывает во время подсчета C243, значение счета в C243 сохраняется в (D1241, D1240) и C243 сбрасывается. После этого выполняется подпрограмма прерывания, обозначенная I101.



Функциональная группа Включение питания на входе X

Номер M1304

Описание:
Когда M1304 включен, WPLSoft или ISPSoft устанавливают состояние ВКЛ/ВЫКЛ на входе X, но связанные LED не будут реагировать на это.

Функциональная группа Заданные выходные импульсы или поиск Z-фазы при достижении нулевой точки.

Номер M1308, D1312

Описание:
Когда нулевая точка достигнута, ПЛК с помощью этой функции может вывести заданные выходные импульсы или осуществлять поиск Z-функции. Входы X2, X3 являются Z-фазой сигнала входов CH1, CH2. Когда M1308 включен, D1312 – назначенный регистр для указания дополнительных импульсов в диапазоне -30,000~30,000. При превышении диапазона, значения автоматически корректируются к минимальному или максимальному значению. Когда D1312 установлен в 0, дополнительные выходные импульсы не выдаются.

Работа других входов:
X4 → CH1 входной сигнал DOG
X5 → CH1 входной сигнал LSN
X6 → CH2 входной сигнал DOG
X7 → CH2 входной сигнал LSN

Функциональная группа ID правосторонних модулей для ES2/EX2

Номер D1320~ D1327

Описание:
При подключении правосторонних модулей к ES2/EX2, ID каждого модуля ввода/вывода сохраняется в D1320~D1327 по порядку.

ID специальных модулей:

Наименование	ID (HEX)
DVP04AD-E2	H'0080
DVP02DA-E2	H'0041
DVP04DA-E2	H'0081

Наименование	ID (HEX)
DVP06XA-E2	H'00C4
DVP04PT-E2	H'0082
DVP04TC-E2	H'0083

Функциональная группа ID левосторонних модулей для SA2/SX2

Номер D1386~D1393

Описание:
При подключении левосторонних модулей к SA2/SX2, ID каждого модуля ввода/вывода сохраняется в D1386~D1393 по порядку.

ID специальных модулей:

Наименование	ID (HEX)	Наименование	ID (HEX)
DVP04AD-SL	H'4480	DVP01HC-SL	H'4120
DVP04DA-SL	H'4441	DVP02HC-SL	H'4220
DVP04PT-SL	H'4402	DVPDNET-SL	H'4131
DVP04TC-SL	H'4403	DVPEN01-SL	H'4050
DVP06XA-SL	H'6404	DVPMDM-SL	H'4040
DVP01PU-SL	H'4110	DVPCOPM-SL	H'4133

Функциональная группа Очистка выходного сигнала после завершения команды ZRN

Номер M1346

Описание:

Когда M1346 включен, ПЛК выдает сбрасывающие сигналы после выполнения ZRN. Сбрасывающие сигналы для Y0, Y1 устанавливаются в Y4 через 20 мс, а для to Y2, Y3 устанавливаются в Y5 также через 20 мс.

Функциональная группа PLC LINK

Номер M1350-M1356, M1360-M1439, D1355-D1370, D1399, D1415-D1465, D1480-D1991

Описание:

1. PLC LINK поддерживается COM2 (RS-485) для связи с 16 Slave-устройствами и оперирует объемом до 50 слов.
2. Специальные D и M, соответствующие Slave-устройствам ID1~ID8: (M1353 выключен, доступны 16 слов)

MASTER ПЛК															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
Специальные регистры D для чтения/записи 16 слов данных (назначаются автоматически)															
D1480 | D1495	D1496 | D1511	D1512 | D1527	D1528 | D1543	D1544 | D1559	D1560 | D1575	D1576 | D1591	D1592 | D1607	D1608 | D1623	D1624 | D1639	D1640 | D1655	D1656 | D1671	D1672 | D1687	D1688 | D1703	D1704 | D1719	D1720 | D1735
Длина данных для доступа к Slave-устройству (не более 16 слов данных, нет доступа при SV = 0)															
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Стартовый операнд для доступа к Slave-устройству*															
D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
M1355 = ON, Slave-статус определяется пользователем. Установка статуса связи вручную M1360~M1375. M1355 = OFF, Slave-статус устанавливается автоматически. Статус связи можно контролировать M1360~M1375															
M1360		M1361		M1362		M1363		M1364		M1365		M1366		M1367	
Состояние обмена данными с Slave-устройствами.															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
Флаг ошибки доступа (ON = норма; OFF = ошибка)															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
Флаг "Чтение завершено"															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	
Флаг «Запись завершена»															
M1424		M1425		M1426		M1427		M1428		M1429		M1430		M1431	

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

Slave ПЛК*															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

3. Специальные D и M, соответствующие Slave-устройствам ID9~ID16: (M1353 выключен, доступны 16 слов)

MASTER ПЛК															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
Специальные регистры D для чтения/записи 16 слов данных (назначаются автоматически)															
D1736 | D1751	D1752 | D1767	D1768 | D1783	D1784 | D1799	D1800 | D1815	D1816 | D1831	D1832 | D1847	D1848 | D1863	D1864 | D1879	D1880 | D1895	D1896 | D1911	D1912 | D1927	D1928 | D1943	D1944 | D1959	D1960 | D1975	D1976 | D1991
Длина данных для доступа к Slave-устройству (не более 16 слов данных, нет доступа при SV = 0)															
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Стартовый операнд для доступа к Slave-устройству*															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
M1355 = ON, Slave-статус определяется пользователем. Установка статуса связи вручную M1360~M1375. M1355 = OFF, Slave-статус устанавливается автоматически. Статус связи можно контролировать M1360~M1375															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Состояние обмена данными с Slave-устройствами.															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
Флаг ошибки доступа (ON = норма; OFF = ошибка)															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	
Флаг “Чтение завершено”															
M1416		M1417		M1418		M1419		M1420		M1421		M1422		M1423	
Флаг «Запись завершена»															
M1432		M1433		M1434		M1435		M1436		M1437		M1438		M1439	
↓		↓		↓		↓		↓		↓		↓		↓	
Slave ПЛК*															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

4. Специальные D и M, соответствующие Slave-устройствам ID1~ID8: (M1353 включен, доступны 50 слов)

MASTER ПЛК															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
M1353 = ON, включен доступ к 50 словам. Пользователь может задать начальный регистр для хранения прочитанных/записанных данных:															
D1480	D1496	D1481	D1497	D1482	D1498	D1483	D1499	D1484	D1500	D1485	D1501	D1486	D1502	D1487	D1503
M1356 = ON, пользователь может указать номер Slave-устройства ID1~ID8 в D1900~D1907															
D1900		D1901		D1902		D1903		D1904		D1905		D1906		D1907	
Длина данных для доступа к Slave-устройству (не более 50 слов данных, нет доступа при SV = 0)															
D1434	D1450	D1435	D1451	D1436	D1452	D1437	D1453	D1438	D1454	D1439	D1455	D1440	D1456	D1441	D1457
Стартовый операнд для доступа к Slave-устройству*															

D1355	D1415	D1356	D1416	D1357	D1417	D1358	D1418	D1359	D1419	D1360	D1420	D1361	D1421	D1362	D1422
M1355 = ON, Slave-статус определяется пользователем. Установка статуса связи вручную M1368~M1375. M1355 = OFF, Slave-статус устанавливается автоматически. Статус связи можно контролировать M1368~M1375															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Состояние обмена данными с Slave-устройствами.															
M1376		M1377		M1378		M1379		M1380		M1381		M1382		M1383	
Флаг ошибки доступа (ON = норма; OFF = ошибка)															
M1392		M1393		M1394		M1395		M1396		M1397		M1398		M1399	
Флаг “Чтение завершено”															
M1408		M1409		M1410		M1411		M1412		M1413		M1414		M1415	

Флаг «Запись завершена»							
M1424	M1425	M1426	M1427	M1428	M1429	M1430	M1431
↓	↓	↓	↓	↓	↓	↓	↓

Slave ПЛК*															
SLAVE ID 1		SLAVE ID 2		SLAVE ID 3		SLAVE ID 4		SLAVE ID 5		SLAVE ID 6		SLAVE ID 7		SLAVE ID 8	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

5. Специальные D и M, соответствующие Slave-устройствам ID9~ID16: (M1353 включен, доступны 50 слов)

MASTER ПЛК															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
M1353 = ON, включен доступ к 50 словам. Пользователь может задать начальный регистр для хранения прочитанных/записанных данных:															
D1488	D1504	D1489	D1505	D1490	D1506	D1491	D1507	D1492	D1508	D1493	D1509	D1494	D1510	D1495	D1511
M1356 = ON, пользователь может указать номер Slave-устройства ID9~ID16 в D1908~D1915															
D1908		D1909		D1910		D1911		D1912		D1913		D1914		D1915	
Длина данных для доступа к Slave-устройству (не более 50 слов данных, нет доступа при SV = 0)															
D1442	D1458	D1443	D1459	D1444	D1460	D1445	D1461	D1446	D1462	D1447	D1463	D1448	D1464	D1449	D1465
Стартовый операнд для доступа к Slave-устройству*															
D1363	D1423	D1364	D1424	D1365	D1425	D1366	D1426	D1367	D1427	D1368	D1428	D1369	D1429	D1370	D1430
M1355 = ON, Slave-статус определяется пользователем. Установка статуса связи вручную M1368~M1375. M1355 = OFF, Slave-статус устанавливается автоматически. Статус связи можно контролировать M1368~M1375															
M1368		M1369		M1370		M1371		M1372		M1373		M1374		M1375	
Состояние обмена данными с Slave-устройствами															
M1384		M1385		M1386		M1387		M1388		M1389		M1390		M1391	
Флаг ошибки доступа (ON = норма; OFF = ошибка)															
M1400		M1401		M1402		M1403		M1404		M1405		M1406		M1407	
Флаг “Чтение завершено”															
M1416		M1417		M1418		M1419		M1420		M1421		M1422		M1423	

Флаг «Запись завершена»							
M1432	M1433	M1434	M1435	M1436	M1437	M1438	M1439
↓	↓	↓	↓	↓	↓	↓	↓

Slave ПЛК*															
SLAVE ID 9		SLAVE ID 10		SLAVE ID 11		SLAVE ID 12		SLAVE ID 13		SLAVE ID 14		SLAVE ID 15		SLAVE ID 16	
Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в	Чтение из	Запись в
D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215	D100 | D115	D200 | D215

*Примечание:

По умолчанию – для начала управления Slave-устройствами (DVP-PLC) для чтения: H1064 (D100), для записи: H10C8 (D200)

6. Описание:

a) PLC LINK основан на протоколе связи MODBUS

b) Скорость передачи и формат связи всех периферийных устройств, связанных с ПЛК в режиме Slave должны быть аналогичными параметрам связи, когда ПЛК находится в Master-режиме, независимо от COM-порта.

c) Когда M1356 выключен (по умолчанию), номер первого Slave-устройства (ID1) обозначен в D1399 Master ПЛК, и через PLC LINK, ПЛК автоматически назначит ID2~ID16 как номера устройств после ID1. Например, если D1399 = K3, Master ПЛК будет передавать команды связи на ID1~ID16, обозначаемые K3~K18. Номера Slave-устройств не должны совпадать с номерами Master ПЛК, установленными в D1121/D1255.

d) Когда M1353 и M1356 включены, номера устройств ID1~ID16 назначаются пользователем в D1900~D1915 Master ПЛК. Номера Slave-устройств не должны совпадать с номерами Master ПЛК и M1353 должен быть включен для реализации данной функции.

e) Функция выбора номера Slave-устройства (M1356 включен) поддерживается ES2/EX2 v1.4.2 или выше, SS2/SX2 v1.2 или выше, and SA2 v1.0 или выше.

7. Работа:

a) Установите скорость передачи и формат связи. Master ПЛК и все подсоединенные Slave ПЛК должны иметь одинаковые параметры связи. COM1_RS-232: D1036, COM2_RS-485: D1120, COM3_RS-485: D1109.

b) Установите ID Master PLC как D1121 и ID начального Slave ПЛК как D1399. Затем ID каждого Slave ПЛК. ID Master ПЛК и Slave PLC не должны совпадать.

c) Установите длину данных для обмена. (если длина не указана, ПЛК примет ее по умолчанию или предыдущее значение, см. таблицу выше по параметрам данных).

d) Установите стартовые регистры Slave ПЛК. По умолчанию для чтения read: H1064 (D100); для записи: H10C8 (D200). См. таблицу выше)

e) Шаги запуска PLC LINK:

- Включите M1354 для функции одновременного чтения/записи данных в PLC LINK.
- M1355 включен, Slave-статус устанавливается пользователем. Установка статуса связи вручную M1360~M1375. M1355 выключен, Slave-статус устанавливается автоматически. Статус связи можно контролировать M1360~M1375.
- Выберите авторежим PLC LINK M1351 или ручной режим M1352 связи (оба флага не могут быть включены одновременно) Установите время цикла опроса в D1431.
- Запустите PLC LINK (M1350 включен)

8. Работа Master ПЛК:

a) M1355 включен, Slave-статус устанавливается пользователем. Установка статуса связи вручную M1360~M1375.

b) M1355 выключен, Slave-статус устанавливается автоматически. Статус связи можно контролировать M1360~M1375.

- Включите PLC LINK (M1350 включен). Master ПЛК определяет все подключенные Slave-устройства и сохраняет их количество в D1433. Время обнаружения зависит от количества подключенных устройств и времени ожидания ответа, заданного в D1129.
- M1360~M1375 показывает статус связи для ID 1~16.
- Если Slave-устройства не обнаружены, M1350 выключается и PLC LINK останавливается.
- ПЛК определяет все подключенные Slave-устройства каждый раз после включения M1350.

- После выполнения автообнаружения, Master ПЛК получает доступ к каждому подключенному устройству. Позже подключенные устройства обнаружены не будут до проведения нового автообнаружения.

c) Функция одновременного чтения/записи (M1354) устанавливается перед включением PLC LINK. Во время работы PLC LINK запуск этой функции невозможен.

d) Когда M1354 включен, ПЛК выбирает функцию Modbus H17 (одновременные чтение/запись) для PLC LINK. Если длина данных для чтения установлена равной 0, ПЛК автоматически выбирает функцию Modbus H03 (чтение нескольких слов). Если длина данных для записи установлена равной 0, ПЛК выбирает функцию Modbus H06 (запись одного слова) или функцию H10 (запись нескольких слов) для PLC LINK.

e) Когда M1353 выключен, PLC LINK доступно не более 16 слов, данные автоматически сохраняются в соответствующих регистрах. Когда M1353 включен, доступно до 50 слов и пользоваель может задавать начальный регистр хранения считанных/записанных данных.
Например, если регистры хранения считанных/записанных данных от Slave-устройства ID1 определены как D1480 = K500, D1496 = K800, длина доступных данных D1434 = K50, D1450 = K50, Регистры Master-ПЛК D500~D549 сохраняют данные от Slave-устройсва ID1, а данные из регистров D800~D849 передаются в Slave-устройство ID1.

f) Master-ПЛК проводит чтение преред записью. Чтение и запись выполняются в диапазоне, заданном пользователем.

g) Master-ПЛК работает со Slave-ПЛК по порядку, т.е. сеанс со следующим начинается после завершения связи с пердыдущим.

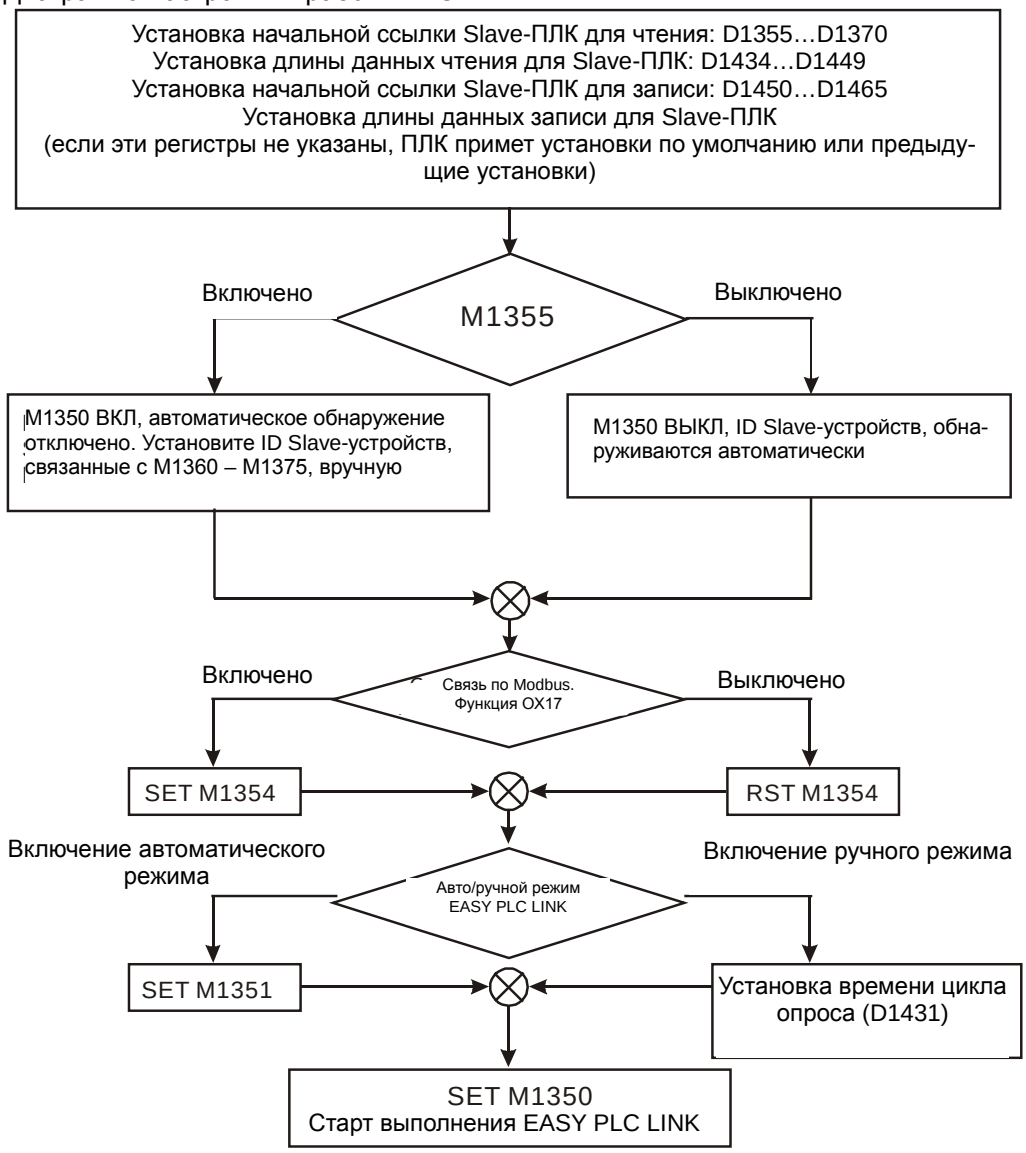
9. Автоматический и ручной режимы:

a) Автоматический режим (M1351): когда M1351 включен, Master-ПЛК работает как описано выше и останавливается при выключении M1350 или M1351.

b) Ручной режим (M1352): когда выбран ручной режим, время цикла опроса может быть установлено в D1431. Полное время опроса складывается из времени опрса всех Slave-устройств. Когда PLC LINK запущен, D1432 – стартовый регистр хранения времени опроса. Когда D1431 = D1432, PLC LINK останавливается и M1352 очищается. Когда M1352 включается снова, PLC автоматически начинает опрос по времени, заданном в D1431.

c) Заметки:
- Автоматический режим M1351 и ручной режим M1352 не могут быть запущены одновременно. Если M1351 включить после включения M1352, PLC LINK прекратит работу и M1350 сбрасывается.
- Время ожидания связи может быть изменено в D1129 в диапазоне 200 $\leqq$ D1129 $\leqq$ 3000. В случае выхода за пределы диапазона ПЛК примет соответствующе крайнее значение. D1129 настраивается до включения M1350.
- Функция PLC LINK работает на скорости выше 1200 бит/с. Когда скорость меньше 9600 бит/с, устанавливайте время ожидания не менее 1 сек.
- Связь не осуществляется при длине данных, равной 0.
- Доступ к 32-битным высокоскоростным счетчикам (C200~C255) не поддерживается.
- Диапазон D1399: 1 ~ 230. В случае выхода за пределы диапазона ПЛК примет соответствующе крайнее значение.
- D1399 может применяться для увеличения числа Slave-устройств в многоуровневых сетях.

10. Диаграмма настройки и работы PLC LINK:



11. Пример 1: соединение 1 Master и 2 Slave-устройств по RS-485 и обмен 16 пакетами данных через PLC LINK

a) Запишите лестничную диаграмму в Master ПЛК (ID#17)



b) Когда X1 включен, происходит обмен данными между Master и 2-мя Slaves-устройствами автоматически при помощи PLC LINK. Данные в D100 ~ D115 (2 Slave-устройства) читаются в D1480 ~ D1495 и D1512 ~ D1527 регистры Master-ПЛК, а данные из D1496 ~ D1511 и D1528 ~ D1543 передаются в D200 ~ D215 регистры 2-х Slave-устройств.

Master ПЛК *1		Slave ПЛК*2
D1480 ~ D1495	Чтение ←	D100 ~ D115 (Slave-устр. ID#1)
D1496 ~ D1511	Запись →	D200 ~ D215 (Slave-устр. ID#1)
D1512 ~ D1527	Чтение ←	D100 ~ D115 (Slave-устр. ID#2)
D1528 ~ D1543	Запись →	D200 ~ D215 (Slave-устр. ID#2)

c) Предположим, что данные для обмена перед запуском PLC LINK (M1350 выключен) следующие:

Master ПЛК	Предустановка	Slave ПЛК	Предустановка
D1480 ~ D1495	K0	D100 ~ D115 (Slave-устр. ID#1)	K5 000
D1496 ~ D1511	K1 000	D200 ~ D215 (Slave-устр. ID#1)	K0
D1512 ~ D1527	K0	D100 ~ D115 (Slave-устр. ID#2)	K6 000
D1528 ~ D1543	K2 000	D200 ~ D215 (Slave-устр. ID#2)	K0

После запуска PLC LINK (M1350 включен) данные в регистрах будут:

Master ПЛК	Предустановка	Slave ПЛК	Предустановка
D1480 ~ D1495	K5 000	D100 ~ D115 (Slave-устр. ID#1)	K5 000
D1496 ~ D1511	K1 000	D200 ~ D215 (Slave-устр. ID#1)	K1 000
D1512 ~ D1527	K6 000	D100 ~ D115 (Slave-устр. ID#2)	K6 000
D1528 ~ D1543	K2 000	D200 ~ D215 (Slave-устр. ID#2)	K2 000

d) Через PLC LINK можетбыть доступно до 16 Slave-устройств. Для обозначения D100 ~ D115 и D200 ~ D215 кождого Slave-ПЛК, см. таблицу специальных M иl D выше.

12. Пример 2: Свяжем ПЛК DVP с ПЧ VFD-M и применяем операции RUN, STOP, Движение вперед, Движение назад через PLC LINK.

a) Запишите лестничную диаграмму в Master ПЛК (ID#17)



b) M1355 включен. Установим вручную для Slave-устройств M1360~M1375. Включим M1360 для связи со Slave-устройством ID#1.

c) Параметры ПЧ H2100-H2105 передаются в регистры ПЛК D1480-D1485, когда X1 включается.

d) Параметры ПЧ H2000-H2001 передаются в регистры ПЛК D1496-D1497 of PLC, когда X1 включается.

e) Команды VFD выбираются изменением значения в D1496, например, D1496 = H12=>VFD – движение вперед; D1496 = H1=> VFD останавливается.

f) Частота VFD выбираются изменением значения в D1497, например, D1497 = K5000, установленное значение частоты VFD равно 50 кГц.

g) По протоколу MODBUS в качестве Slave-устройств можно применять как ПЧ VFD, так и температурные контроллеры DTA/DTB и ервоприводы ASDA, до 16 устройств.

13. D1354 показывает время цикла сканирования в PLC LINK с шагом 1 мс и максимальным значением K32000. D1354 = K0 когда PLC LINK останавливается или после завершения первого цикла сканирования.

Система команд для программирования контроллеров DVP

В данном разделе приводится подробное описание инструкций контроллеров DVP и детальное описание правил их применения

3.1 Базовые команды (без API номеров)

Команда	Функция	Операнды	Скорость выполнения (мкс)	К-во шагов
LD	Нормально-открытый (Н/О) контакт	X, Y, M, S, T, C	0.76	1~3
LDI	Нормально-закрытый (Н/З) контакт	X, Y, M, S, T, C	0.78	1~3
AND	Последовательный Н/О контакт	X, Y, M, S, T, C	0.54	1~3
ANI	Последовательный Н/З контакт	X, Y, M, S, T, C	0.56	1~3
OR	Параллельный Н/О контакт	X, Y, M, S, T, C	0.54	1~3
ORI	Параллельный Н/З контакт	X, Y, M, S, T, C	0.56	1~3
ANB	Последовательно блоки	X, Y, M, S, T, C	0.68	1
ORB	Параллельно блоки	X, Y, M, S, T, C	0.76	1
MPS	Смещение вниз по стеку	X, Y, M, S, T, C	0.74	1
MRD	Считать значение стека	X, Y, M, S, T, C	0.64	1
MPP	Выход из стека	X, Y, M, S, T, C	0.64	1
OUT	Присвоение выходу результата предыдущего логического выражения	Y, M, S	0.88	1~3
SET	Фиксированное включение операнда (установка логической "1")	Y, M, S	0.76	1~3
RST	Сброс контактов в исходное, очистка регистров от содержимого	Y, M, S, T, C, D, E, F	2.2	3
MC	Начало исключаемого участка программы	N0 ~ N7	1	3
MCR	Конец исключаемого участка программы	N0 ~ N7	1	3
END	Конец программы	нет	1	1
NOP	Пустая строка	нет	0.4	1
P	Указатель	P0 ~ P255	0.4	1
I	Указатель прерывания	I□□□	0.4	1
STL	Начало пошагового управления	S	2.2	1
RET	Конец пошагового управления	нет	1.6	1
NP	Отрицательный контакт - положительный контакт	N/A	1.66	1
PN	Положительный контакт - отрицательный контакт	N/A	1.62	1

Примечание: Скорость выполнения определена для основных тестовых программ, фактическое время выполнения команд может быть выше в более сложных программах, например, программа содержит несколько прерываний или высокая скорость ввода/вывода.

3.2 Описание базовых команд

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
LD	X, Y, M, S, T, C	Нормально-открытый контакт	1~3				

Описание:

1. Команда LD используется в качестве нормально-открытого контакта для программирования начала логических цепочек. В контактных схемах команда всегда расположена слева и соединяется непосредственно с шиной питания или начинает новый блок программ, соединенных последовательно или параллельно.
2. ПЛК серий DVP-ES2/EX2 версии прошивки 3.20, DVP-SS2 версии 3.00, DVP-SA2 версии 2.60, DVP-SE версии 1.20, DVP-SX2 версии 2.40 и выше поддерживают операнды X, Y, M и S. Эти операнды определяются как E или F. Используйте ПО WPLSoft версии 2.31 и выше, ISPSoft версии 2.01 и выше.

Применение:

Релейно-контактная схема:

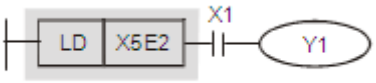
LD	**X0**	Выход Н/О контакта X0
AND	X1	Последовательно Н/О контакт X1
OUT	Y1	Катушка Y1



Команда "нормально-открытый контакт X0" открывает последовательную логическую связь. Если на входах X0 и X1 одновременно будет сигнал "1", тогда и выход Y1 установится в состояние "1".

Релейно-контактная схема:

LD	**X5E2**	Выход Н/О контакта X3 **(принимаем E2=K-2)**
AND	X1	Последовательно Н/О контакт X1
OUT	Y1	Катушка Y1



Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
LDI	X, Y, M, S, T, C	Нормально-закрытый контакт	1~3				

Описание:

Команда LDI используется в качестве нормально-открытого контакта для программирования начала логических цепочек. В контактных схемах команда всегда расположена слева и соединяется непосредственно с шиной питания или начинает новый блок программ, соединенных последовательно или параллельно

Применение:

Релейно-контактная схема:



Команда:	Операция:
LDI X0	Выход Н/З контакта X0
AND X1	Последовательно Н/О контакт X1
OUT Y1	Катушка Y1

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
AND	X, Y, M, S, T, C	Последовательный нормально-открытый контакт (логическое И)	1~3				

Описание:

Команда AND используется в качестве последовательного нормально-открытого контакта для программирования операции логического умножения (И).

Применение:

Релейно-контактная схема:



Команда:		Операция:
LDI	X1	Выход Н/З контакта X1
AND	**X0**	Последовательно Н/О контакт X0
OUT	Y1	Катушка Y1

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
ANI	X, Y, M, S, T, C	Последовательный нормально-закрытый контакт (И-НЕ)	1~3				

Описание:

Команда ANI используется в качестве последовательного нормально-закрытого контакта для программирования операции И-НЕ.

Применение:

Релейно-контактная схема:



Команда:		Операция:
LD	X1	Выход Н/О контакта X1
ANI	**X0**	Последовательно Н/З контакт X0
OUT	Y1	Катушка Y1

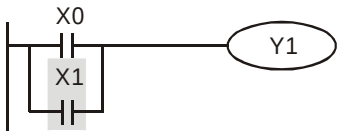
Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
OR	X, Y, M, S, T, C	Параллельный нормально-открытый контакт (логическое ИЛИ)	1~3				

Описание:

Команда OR используется в качестве параллельного нормально-открытого контакта для программирования операции логического сложения (ИЛИ).

Применение:

Релейно-контактная схема:



Команда:		Операция:
LD	X0	Выход Н/О контакта X0
OR	**X1**	Параллельно Н/О контакт X1
OUT	Y1	Катушка Y1

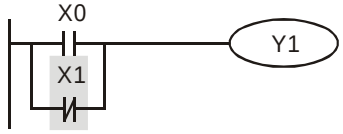
Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
ORI	X, Y, M, S, T, C	Параллельный нормально-закрытый контакт (ИЛИ-НЕ)	1~3				

Описание:

Команда ORI используется в качестве параллельного нормально-закрытого контакта для программирования логической операции ИЛИ-НЕ.

Применение:

Релейно-контактная схема:

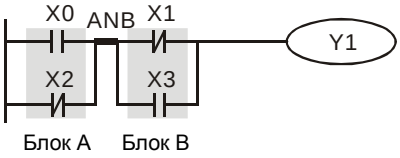


Команда:		Операция:
LD	X0	Выход Н/О контакта X0
RI	**X1**	Параллельно Н/З контакт X1
OUT	Y1	Катушка Y1

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/ SE	SX2
ANB	«И» блок: последовательное включение блоков контактов	1				

Описание:
Команда ANB используется для последовательного соединения цепочек из двух групп контактов. Отдельные блоки, параллельно включенных элементов, заносятся в программу раздельно. Чтобы эти блоки соединить последовательно, после каждого блока программируется ANB инструкция.

Применение:
Релейно-контактная схема:

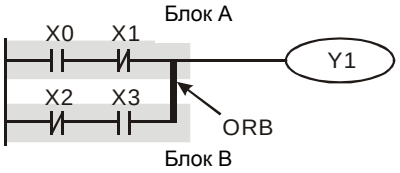


Команда:		Операция:
LD	X0	Выход Н/О контакта X0
ORI	X2	Параллельно Н/З контакт X2
LDI	X1	Выход Н/З контакта X1
OR	X3	Параллельно Н/О контакт X3
ANB		Последовательно блоки
OUT	Y1	Катушка Y1

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/ SE	SX2
ORB	«ИЛИ» блок: параллельное включение блоков контактов	1				

Описание:
Команда ORB используется для параллельного соединения групп контактов. Если несколько последовательных блоков включаются параллельно, то нужно после программирования каждого отдельного блока вводить ORB-инструкцию.

Применение:
Релейно-контактная схема:



Команда:		Операция:
LD	X0	Выход Н/О контакта X0
ANI	X1	Последовательно Н/З контакт X1
LDI	X2	Выход Н/З контакта X2
AND	X3	Последовательно Н/О контакт X3
ORB		Параллельно блоки
OUT	Y1	Катушка Y1

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/ SE	SX2
MPS	Точка начала разветвления с одним входным условием для всего разветвления	1				

Описание:
В начале ветки MPS сохраняет текущий результат выполнения программы в точке разветвления

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/ SE	SX2
MRD	Промежуточная точка разветвления с одним входным условием для всего разветвления (ответвление)	1				

Описание:

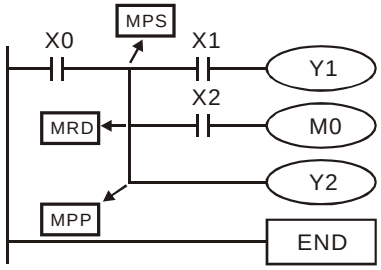
MRD читает текущий результат с предыдущей точки MPS и оперирует с контактом после MRD.

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/ SE	SX2
MPP	Точка конца разветвления с одним входным условием для всего разветвления	1				

Описание:
Инструкции MPS, MRD, MPP служат для того, чтобы создавать уровни логических связей – разветвлений. Например, после одного начального логического выражения создать несколько логических выражений на выходе, т.е. включать несколько выходов-катушек от одного входа. С программной точки зрения данные команды представляют собой точки, обладающие памятью результата предыдущих логических операций.

Примечание:
1. Каждая инструкция MPS не может быть применена без соответствующей ей инструкции MPP
2. Максимальное число пар MPS-MPP не должно быть более 8.

Применение:
Релейно-контактная схема:



Инструкция:		Операция:
LD	X0	Выход Н/О контакта X0
MPS		Хранение текущего состояния
AND	X1	Последовательно Н/О контакт X1
OUT	Y1	Катушка Y1
MRD		Чтение сохраненного состояния
AND	X2	Последовательно Н/О контакт X2
OUT	M0	Катушка M0
MPP		Чтение сохраненного состояния и сброс
OUT	Y2	Катушка Y2
END		Конец программы

Примечание: При компиляции в WPLSoft программы виде релейной логики инструкции, MPS, MRD и MPP будут после компиляции автоматически добавляться в результат в формате листа инструкций. Если программа создаётся в формате листа инструкций, то пользователю необходимо внести эти команды самостоятельно.

3

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
OUT	Y, M, S	Выход	1~3				

Применение:
Команда OUT служит для включения или отключения выхода в зависимости от результата логических связей (результата обработки центральным процессором логического выражения).

Состояние выхода

Результат	Инструкция OUT		
	Выход	Присоединённые контакты	
		нормально-открытый	нормально-закрытый
FALSE	OFF	Ток прерван	Ток течёт
TRUE	ON	Ток течёт	Ток прерван

Применение:
Релейно-контактная схема:



Инструкция:		Операция:
LDI	X0	Выход Н/О контакта X0
AND	X1	Последовательно Н/О контакт X1
OUT	**Y1**	Катушка Y1

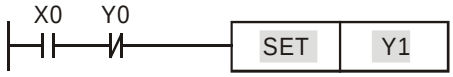
При условии: X0=0 и X1=1 – команда OUT Y1 установит выход контроллера Y1 в состояние "1".

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
SET	Y, M, S	Включение выхода с фиксацией	1~3				

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. С помощью SET могут устанавливаться в "1" (включаться) операнды Y, M или S.

Применение:

Релейно-контактная схема:



Команда		Операция
LD	X0	Выход Н/О контакта X0
ANI	Y0	Последовательно Н/З контакт Y0
SET	**Y1**	Катушка Y1 и фиксация состояния

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
RST	Y, M, S, T, C, D, E, F	Сброс состояния операнда	3				

Описание:

Состояние устройств при выполнении команды :

Устройство	Состояние
S, Y, M	Выход и контакт переходят в состояние OFF.
T, C	Очистка текущего значения. Связанные с ними контакты и катушки сбрасываются
D, E, F	Содержимое равно 0.

До выполнения инструкции состояние не меняется.

Применение:

Релейно-контактная схема:



Команда		Операция
LD	X0	Выход Н/О контакта X0
RST	**Y5**	Сброс контакта Y5

Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
MC/MCR	N0~N7	Мастер управления "старт/сброс"	3				

Описание:

Инструкции MC/MCR позволяют временно отключать определенные участки программы (мастер-контроль).

Инструкция MC обозначает начало исключаемого участка. По своему назначению данная инструкция является аналогичной главному контакту питающей шины в релейно-контактных схемах, отключающему участок схемы от источника питания.

Инструкция MCR обозначает конец исключаемого участка программы. Перед инструкцией MCR не должно стоять никаких контактов и условий.

Инструкции MC/MCR поддерживают до 8 уровней вложенности, но последовательность нумерации должна быть строго по возрастанию: от N0 до N7.

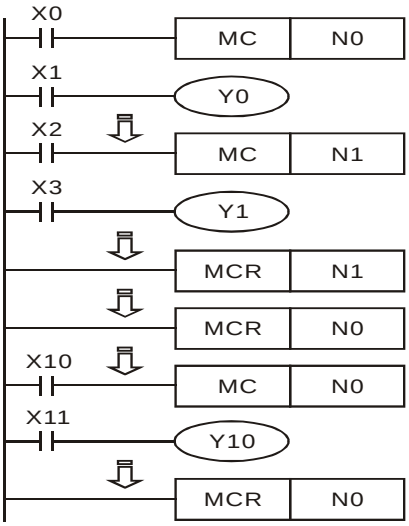
Если условие включения инструкции MC выполняется, то участок программы между MC и MCR соответствующего номера вложенности будет выполняться без каких-либо ограничений.

Если условие включения инструкции MC не выполняется, то участок программы до соответствующей инструкции MCR не будет выполняться и операнды примут следующие состояния:

Операнд	Описание
Таймеры общего назначения	Текущее значение = 0. Контакты не реагируют, выходы отключены.
Аккумулятивный таймер	Выход отключен, текущее значение и состояние контактов не меняется.
Таймер подпрограмм	Выход отключен, текущее значение и состояние контактов не меняется.
Счетчики	Выходы отключены, текущее значение и состояние контактов не меняется.
Выходы, управляемые инструкцией OUT	Все выключены.
Выходы, управляемые инструкциями SET и RST	Остаются без изменения.
Прикладные инструкции	Все выключены. Циклы FOR-NEXT выполняться заданное количество раз, но операнды внутри них будут вести себя как между MC и MCR.

Применение:

Релейно-контактная схема:



Команда:		Операция:
LD	X0	Выход Н/О контакта X0
MC	**N0**	Включение разъема N0
LD	X1	Выход Н/О контакта X1
OUT	Y0	Катушка Y1
LD	X2	Выход Н/О контакта X2
MC	**N1**	Включение разъема N1
LD	X3	Выход Н/О контакта X3
OUT	Y1	Катушка Y1
MCR	**N1**	Сброс разъема N1
MCR	**N0**	Сброс разъема N0
LD	X10	Выход Н/О контакта X10
MC	**N0**	Включение разъема N0
LD	X11	Выход Н/О контакта X11
OUT	Y10	Катушка Y10
MCR	**N0**	Сброс разъема N0

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/SE	SX2
END	Конец программы	1				

Описание:

Любая программа для контроллера (ступенчатая диаграмма или список инструкций) должна заканчиваться командой END. Контроллер осуществляет сканирование программы с шага «0» до команды END, а затем снова возвращается к шагу «0».

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/SE	SX2
NOP	Пустая строка в программе (нет операции)	1				

Описание:

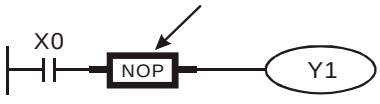
Инструкция NOP не осуществляет в программе никаких действий. Таким образом, после ее выполнения сохраняются логические состояния всех предыдущих шагов программы. Инструкцию NOP применяют в тех случаях, когда необходимо удалить какую-либо действующую инструкцию, сохранив при этом длину программы, или зарезервировать место под какое-либо действие. Применяется в языке «список инструкций».

Количество NOP инструкций в программе не ограничено. После завершения отладки программы инструкции NOP желательно удалить, так как они бесполезно удлиняют программу, увеличивая время цикла.

Применение:

Релейно-контактная схема:

Команды NOP опускаются в лестничных диаграммах



Команда:		Операция:
LD	X0	Выход Н/О контакта X0
NOP		Нет операции
OUT	Y1	Катушка Y1

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/SE	SX2
NP	Формирование импульса длительностью в один скан при переходе логического состояния предшествующих элементов в состояние ON	1				

Описание

Когда предшествующие этой инструкции элементы формируют переход из состояния «Ложное» в состояние «Истина», то в течение одного скана NP команда (работает как контакт

A) формирует однократный сигнал ON. В следующих сканах он переходит в состояние OFF.

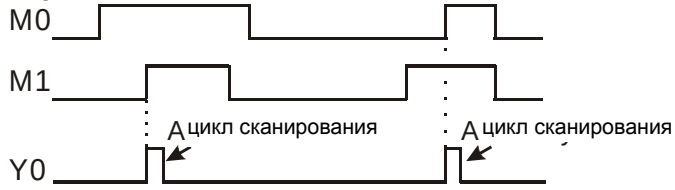
Применение:

Релейно-контактная схема:



Команда:		Операция:
LD	M0	Выход Н/О контакта M0
AND	M1	Последовательно Н/О контакт M1
NP		Отрицательный контакт - положительный контакт
OUT	Y0	Катушка Y0

Временная диаграмма:



Команда	Функция	Кол-во шагов	Контроллеры			
			ES2/EX2	SS2	SA2/ SE	SX2
PN	Формирование импульса длительностью в один скан при переходе логического состояния предшествующих элементов в состояние OFF	1				

Описание

Когда предшествующие этой инструкции элементы формируют переход из состояния «Истина» в состояние «Ложное», то в течение одного скана команда PN (работает как контакт
A) формирует однократный сигнал ON. В следующих сканах он переходит в состояние OFF.

Применение:

Релейно-контактная схема:



Команда:		Операция:
LD	M0	Выход Н/О контакта M0
AND	M1	Последовательно Н/О контакт M1
PN		Положительный контакт - отрицательный контакт
OUT	Y0	Катушка Y0

Временная диаграмма:



3.3 Точки перехода

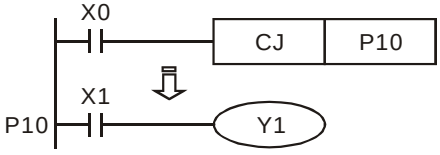
Команда	Операнд	Функция	Кол-во шагов	Контроллер			
				ES2/EX2	SS2	SA2/ SE	SX2
P	P0~P255	Указатель точки перехода	1				

Описание:

P-инструкция служит для указания точки перехода для команд CJ, CALL.
Номер точки в программе не должен повторяться
Дополнительная информация по этой инструкции приведена в разделе 2.12 .

Применение:

Релейно-контактная схема:



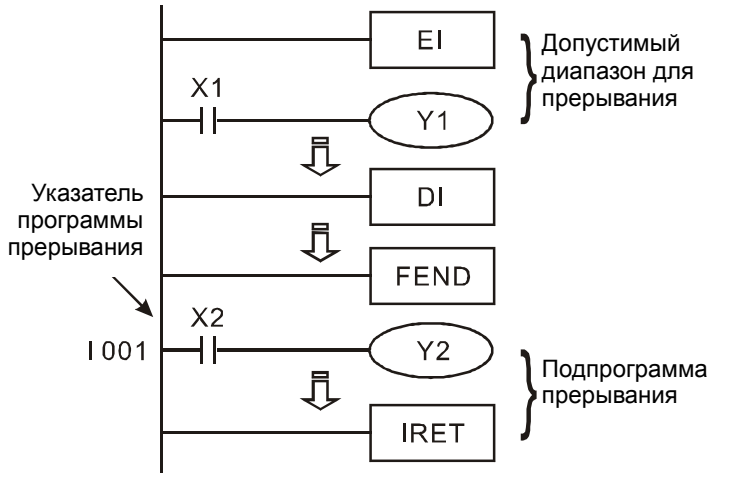
Команда:		Операция:
LD	X0	Выход Н/О контакта X0
CJ	P10	Перейти к P10
:		
P10		Точка P10
LD	X1	Выход Н/О контакта X1
OUT	Y1	Катушка Y1

3.4 Точки прерывания

Команда	Функция	Кол-во шагов	Контроллер			
			ES2/EX2	SS2	SA2/ SE	SX2
I	Указатель точки прерывания	1				

Описание:
Программа прерывания должна начать работу с точки прерывания (I□□□) и закончить командой API 03 IRET. Инструкция должна использоваться с API 03 IRET, API 04 EI, and API 05 DI. Дополнительная информация по этой инструкции приведена в разделе 2.

Применение:
Релейно-контактная схема:



Команда:		Операция:
EI		Включение прерывания
LD	X1	Выход Н/О контакта X1
OUT	Y1	Катушка Y1
:		
DI		Отключение прерывания
:		
FEND		Завершение основной программы
I001		Указатель прерывания
LD	X2	Выход Н/О контакта X2
OUT	Y2	Катушка Y2
:		
IRET		Возврат прерывания

Внешние прерывания:
ES2 поддерживает 8 внешних входов прерываний: (I000/I001, X0), (I100/I101, X1), (I200/I201, X2), (I300/I301, X3), (I400/I401, X4), (I500/I501, X5), (I600/I601, X6) и (I700/I701, X7). (01, по переднему фронту ⌐↑, 00, по заднему фронту ↓¬)

Прерывания по таймеру:
ES2 поддерживает 2 прерывания по таймеру: I602~I699, I702~I799, (Разрешение: 1мс)

Коммуникационные прерывания:
ES2 поддерживает 3 коммуникационных прерывания: I140, I150 и I160.

Прерывания по счётчику:
ES2 поддерживает прерывания по 8 высокоскоростным счётчикам: I010, I020, I030, I040, I050, I060, I070 и I080.

3.5 Прикладные инструкции программирования

1. Для упрощения запоминания применяется мнемоническое обозначение команд, облегчающее их запоминание. Ниже приводится пример API команды 12, обозначение которой MOV и реализуемая функция Move.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2	SX2
12	D	MOV	P	(S) (D)	Движение				

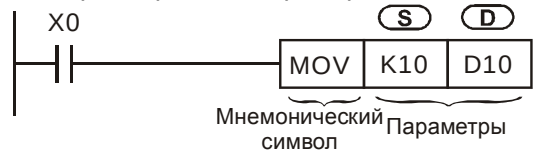
Тип / Опер.	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	MOV, MOVP: 5 шагов
D								*	*	*	*	*	*	*	*	DMOV, DMOVP: 9 шагов

ИМПУЛЬС				16-bit				32-bit			
ES2/EX2	SS2	SA2	SX2	ES2/EX2	SS2	SA2	SX2	ES2/EX2	SS2	SA2	SX2

2. Поле ‘Operands’ определяет параметры (операнды) необходимые для её применения. Идентификационная буква соответствует выполняемым функциям, например. D-получатель результата, S-источник данных, n, m-номера устройств. Дополнительно, суффиксы применяют при использовании операндов с одинаковыми функциями, например. S1, S2.
3. При использовании программы WPLSoft для написания пользовательской программы, нет необходимости запоминать инструкции, так как для выбора требуемой API имеется выпадающее меню.
4. Выбранные модели контроллеров отображаются в нижнем поле справа, там же внизу указываются особенности применения: Pulse (в импульсном виде), 16-bit или 32-bit (в непрерывном).
5. Импульсные инструкции требует прибавление после мнемонического названия буквы ‘P’. 32 разрядные инструкции требую прибавления в начале мнемонического названия буквы ‘D’, в случае применения 32 разрядной импульсной инструкции название будет выглядеть как “D***P”, где *** базовое обозначение.

Составление команд

Каждому номеру инструкции соответствует её мнемоническое изображение и для выбора необходимой инструкции в программе программирования WPLSoft пользователю необходимо только кликнуть по мнемоническому названию, например MOV. После этого, в строку будет вставлена выбранная инструкция.
Команды состоят либо только из команд, либо команды сопровождаются операндам для задания параметров. Для примера возьмем команду MOV:



Мнемоника : Имя и функция инструкции
Операнд : Параметры

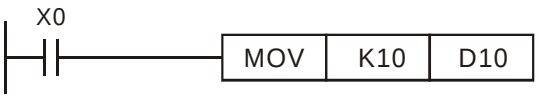
Параметры имеют следующие обозначения:

S	Параметр – источник данных. Если источников несколько, то они нумеруются последовательно нижним индексом: $\mathbf{S_1}$, $\mathbf{S_2}$, …
D	Параметр – получатель результата. Если получателей результата несколько, то они нумеруются последовательно нижним индексом: $\mathbf{D_1}$, $\mathbf{D_2}$, …
Если параметр задается только константой K/H или регистром, то он обозначается как **m**, $\mathbf{m_1}$, $\mathbf{m_2}$, **n**, $\mathbf{n_1}$, $\mathbf{n_2}$, ….	

Длина операндов (16-битные или 32-битные инструкции)

Операнды подразделяются на 2 группы: 16-битные или 32-битные для работы с данными разной длины. Префикс ”D” означает 32-битную инструкцию.

Команда MOV в формате 16 бит



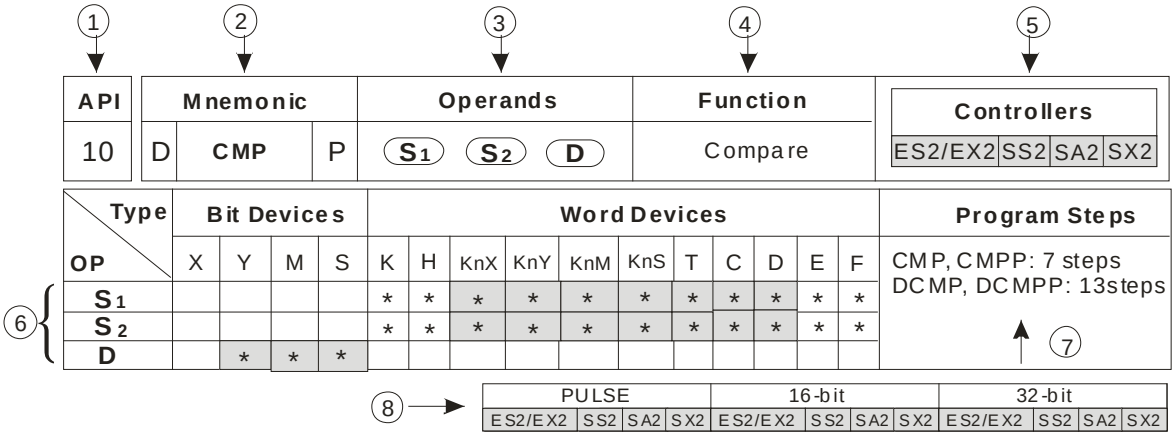
Когда X0 = 1, K10 будет переслано в D10.

Команда DMOV в формате 32 бит



Когда X1 = 1, содержимое регистров (D11, D10) будет переслано в регистры (D21, D20).

Описание формата прикладной команды



① Номер прикладной инструкции

② Мнемоническое обозначение инструкции: префикс “D” означает 32-битную инструкцию, суффикс “P“ означает импульсную инструкцию

③ Формат операндов прикладной инструкции

④ Функция инструкции. Кратко описывается функция, выполняемая данной инструкцией

⑤ Типы контроллеров, поддерживающих выполнение данной инструкции

⑥ Ячейки, помеченные значком «*» и выделенные серым цветом, показывают, что операнд может использовать индексный регистр. Например, D операнда S1 может использоваться с индексными регистрами E и F.

Ячейки, помеченные значком «*», показывают, что данный операнд применим для задания исходных данных соответствующему параметру инструкции.

⑦ Количество шагов, требующееся для выполнения инструкции в формате 16 бит, 32 бит и импульсном режиме.

⑧ Типы контроллеров, поддерживающих данную инструкцию в формате 16 бит, 32 бит и импульсном режиме. Соответствующий тип контроллера выделяется серым цветом.

Непрерывное и импульсное выполнение команд:

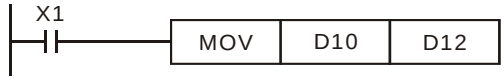
1. Существует два типа выполнения команд - непрерывное выполнение и импульсное выполнение. Время цикла программы меньше, когда команды не выполняются. Таким образом, с помощью импульсного выполнения можно уменьшить время цикла программы.
2. Импульсное выполнение позволяет включить соответствующую команду по переднему (поднимающемся) фронту и выполнить ее в одном цикле программы. Кроме того, пока управляющий вход остается включенным, соответствующая команда не будет выполняться повторно. Для повторного выполнения управляющий вход должен быть переключен с OFF на ON еще раз.

Импульсное выполнение



Когда X0 переходит из состояния OFF в ON, команда MOVP 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.

Операнды:

1. Битовые операнды X, Y, M и S в прикладных командах можно объединять в слова, храня состояние битовых параметров в регистрах данных 16 бит в виде KnX, KnY, KnM и KnS.
2. Словные операнды: регистры D, таймеры T, счетчики C и индексные регистры E, F в прикладных инструкциях обозначаются одноименными операндами без каких-либо изменений.

3. В прикладных командах операнд, как правило, имеет длину 16 бит и занимает стандартный регистр D. Если операнд в прикладной инструкции используется в формате 32 бит, то он займет 2 последовательных регистра D.
4. Если в прикладной команде, работающей только в формате 32 бит, использовать регистр данных 16 бит, например D0, то он все равно займет 2 последовательных регистра D0 и D1. В данном случае D1 – это старшие 16 бит, а D0 – младшие 16 бит. То же самое касается и использования 16-ти битных таймеров и счетчиков C0 ~ C199 в 32-х битных командах.
5. Использование 32-х разрядных счетчиков C200 ~ C255 возможно только с 32-х разрядными командами, в т.ч. и при использовании данных счетчиков в качестве регистров данных.

Формат данных операндов:

1. Операнды X, Y, M и S могут иметь только два состояния (0 или 1), поэтому именуются битовыми параметрами.
2. Операнды D, C, и T и индексы E, F имеют формат словных регистров 16 или 32 бит, поэтому именуются словными операндами.
3. Перед битовыми операндами X, Y, M и S можно поставить коэффициент Kn, при помощи которого битовые операнды можно последовательно объединять в слова (регистры) для дальнейшей обработки. n=1 соответствует 4 битам. Для 16-ти битовых инструкций n = K1 ~ K4, а для 32-х битовых K1 ~ K8. Например: K2M0 соответствует 8 битам M0 ~ M7.



Когда X0 = 1, содержимое M0 ~ M7 будет переслано в биты 0 ~ 7 регистра D10, а биты 8 ~ 15 будет равны 0.

Величины Kn

16-ти битовая команда	
Диапазон значений: K-32,768 ~ K32,767	
Предельные значения при K1 ~ K4	
K1 (4 бит)	0 ~ 15
K2 (8 бит)	0 ~ 255
K3 (12 бит)	0 ~ 4095
K4 (16 бит)	-32768 ~ +32767

32-х битовая команда	
Диапазон значений: K-2,147,483,648 ~ K2,147,483,647	
Предельные значения при K1 ~ K8	
K1 (4 бит)	0 ~ 15
K2 (8 бит)	0 ~ 255
K3 (12 бит)	0 ~ 4095
K4 (16 бит)	0 ~ 65535
K5 (20 бит)	0 ~ 1 048 575
K6 (24 бит)	0 ~ 167 772 165
K7 (28 бит)	0 ~ 268 435 455
K8 (32 бит)	-2 147 483 648 ~ +2 147 483 647

Флаги

При обработке некоторых прикладных команд контроллер автоматически включает или отключает различные флаги (специальные реле). Используемый флаг показывает определенное состояние выполнения команды или программы в целом. Флаг каждый раз включается или отключается, если в программе активизируется соответствующая команда. По своему назначению флаги подразделяются на общие, флаги ошибок и флаги расширения функций.

Общие флаги:

M1020 – флаг нуля. Включается, если результат сложения или вычитания равен нулю

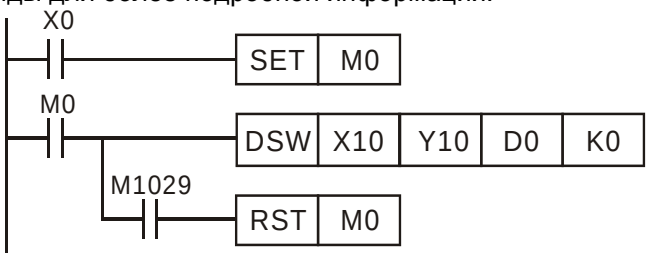
M1021 – флаг заимствования (Borrow). Включается, если результат вычитания меньше самого малого значения

M1022 – флаг переноса (Carry). Включается при передаче значения числа, при суммировании или при передаче данных, при выполнении команды сдвига

M1029 – флаг завершения выполнения команды

Например, результаты выполнения команд ADD/SUB/MUL/DVI влияют на состояние M1020 - M1022. Когда команда не выполняется, будет произведено изменение состояния флага с ON

на OFF. Состояние четырех флагов связано со многими командами. См. соответствующие команды для более подробной информации.



Когда X0 = 1, DSW активируется.

Когда X0 = 0, M0 выключится только, когда команда DSW закончит цикл и M1029 = 1.

Флаги ошибок:

Ошибки при выполнении команд могут возникать при неправильной комбинации команд, или когда какой-либо из операндов выходит за допустимый диапазон. В данных случаях активируются нижеприведенные реле и регистры.

Операнд	Комментарии
M1067 D1067 D1069	M1067 включается при появлении ошибки. В регистре D1067 отображается код ошибки, а в D1069 шаг программы, в котором возникла ошибка. При появлении новых ошибок, содержимое D1067 и D1069 будет автоматически обновляться. Флаг M1067 выключится после устранения ошибки.
M1068 D1068	M1068 включается при появлении ошибки. В регистре D1068 отображается шаг программы, где возникла ошибка. Появление новых ошибок не будет влиять на содержимое D1068 до тех пор, пока флаг M1068 не будет принудительно сброшено командой RST.

Флаги расширения функций:

Некоторые команды могут иметь специальные флаги, включающие или отключающие дополнительные возможности данных функций. Например, у инструкции RS есть реле M1161, которое переключает между режимами 8 или 16 бит.

3

Ограничения по использованию прикладных команд

Количество применений некоторых команд в программе ограничено.
Такие команды могут с помощью индексных регистров расширять свои возможности.

1. Команды, допускающие только однократное использование в программе:
   API 60 (IST) | API 155 (DABSR)
2. Команды, допускающие двукратное использование в программе:
   API 77 (PR)
3. Команды, которые могут быть использованы в программе не более 8-ми раз:
   API 64 (TTMR)
4. Для счётчиков C232~C242: максимально 6 раз могут использоваться DHSCS, DHSCR и DHSZ. Только DHSZ может использоваться менее 6 раз.
5. Для счётчиков C243, C245~C248, C251, C252: максимально 4 раза могут использоваться DHSCS, DHSCR и DHSZ. Только DHSZ может использоваться менее 2 раз
6. Для счётчиков C244, C249, C250, C253, C254: максимально 4 раза могут использоваться DHSCS, DHSCR и DHSZ . Только DHSZ может использоваться менее 2 раз

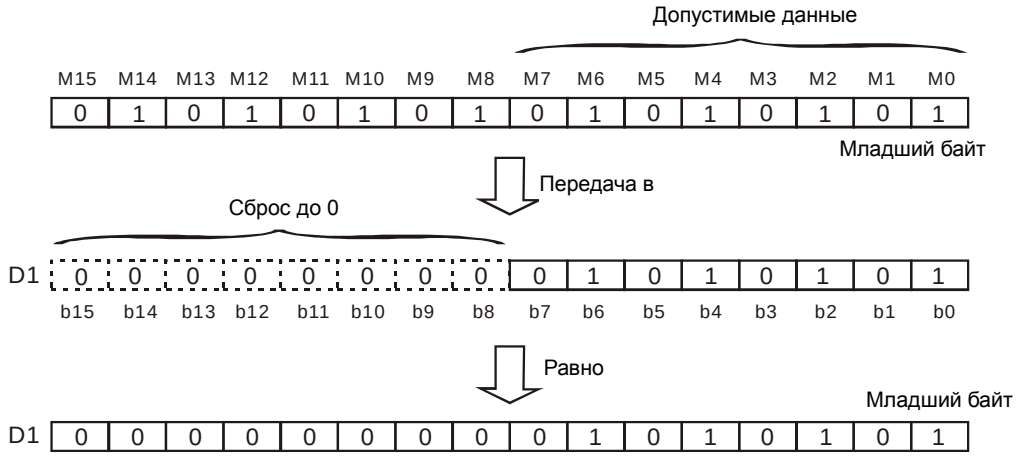
Ограничения по одновременному выполнению команд

Некоторые команды можно использовать неограниченное число раз в одной программе, однако ограничено число их одновременного выполнения:

1. Не более одной инструкции одновременно: API 52 MTR, API 69 SORT, API 70 TKY, API 71 HKY, API 72 DSW, API 74 SEGL, API 75 ARWS.
2. Не более четырех инструкции одновременно: API 56 SPD, API 169 HOUR.
3. Число применений команд высокоскоростного выхода API 57 PLSY, API 58 PWM, API 59 PLSR, API 156DZRN, API 158 DDRVI, API 159 DDRVA и API 195 DPTPO в программе неограничено, но в одном цикле они может применяться только одна из них.
4. На применение в программе коммуникационных команд API 80 RS, API 100 MODRD, API 101 MODWR, API 102 FWD, API 103 REV, API 104 STOP, API 105 RDST, API 106 RSTEF , API 150 MODRW нет ограничений, но в одном цикле для одного порта может использоваться только одна из них.

Числовые значения:

1. Такие операнды, как X, Y, M и S, имеющие состояние ON или OFF называются битовыми.
2. Такие операнды, как T, C, D, E и F, применяемые для хранения значений, называются словными. Битовые операнды, сохраняющие значения ON/OFF, могут также использоваться в словных операндах некоторых команд, если тип данных операнда, указывается, как Kn перед битовым.
3. Для 16-битных данных применяется K1~K4, для 32-битных данных-K1~K8. Например, K2M0 обозначает 8-битовое число, скомпонованное из разрядов M0 ~ M7.



4. Запись K1M0, K2M0, K3M0 в 16-битный регистр. Будут переданы только значащие биты, а старшие биты 16-битного регистра будут заполнены 0. Такое же правило действует при записи K1M0, K2M0, K3M0, K4M0, K5M0, K6M0, K7M0 в 32-битный регистр.
5. Когда значение Kn определяется как K1~K3 (K4~K7) для 16-битной (32-битной) операции, пустые старшие биты заполняются 0, поэтому результат операции положителен, так как знаковый регистр равен 0.



Данные, получающиеся при переносе битовых параметров X0 ~ X7, имеют изначально формат BCD (двоично-десятичный). Для преобразования в двоичный формат используется инструкция BIN, и в регистр D0 данные уже попадут в двоичном формате.

Нумерация битовых параметров

При переносе битовых параметров в словный регистр в качестве начального битового параметра можно использовать любой адрес. Однако, во избежание ошибок лучше начинать с нулевых адресов для X и Y – X0, X10, X20 (восьмеричная система) и т.д., а для M и S первый адрес должен быть кратен восьми (хотя начинать с нулевых адресов и здесь будет лучшим вариантом). Данные рекомендации продемонстрированы в таблице:

K1X0	K1X4	K1X10	K1X14…
K2Y0	K2Y10	K2Y20	Y2X30…
K3M0	K3M12	K3M24	K3M36…
K4S0	K4S16	K4S32	K4S48…

Также, если использовать операцию K4Y0 в 32-х битной инструкции, старшие 16 бит останутся не заполненными, поэтому лучше использовать операцию K8Y0.

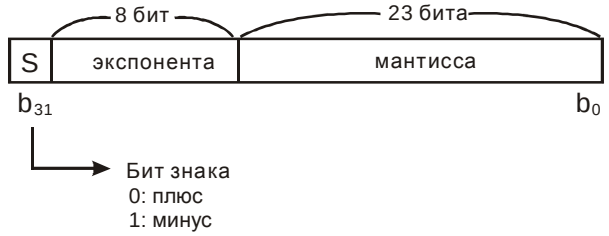
Операции с плавающей запятой

Контроллеры Delta DVP осуществляют расчеты в двоичном формате целого числа. Следовательно, при операциях с числами дробная часть отбрасывается. Например, при делении 40 на 3 получается 13 целых и 3 десятых, которые будут отброшены. При извлечении квадратного корня дробная часть также отбрасывается. Если необходимо осуществлять расчеты с точностью до знаков после запятой, то необходимо использовать специальные инструкции, перечисленные ниже:

FLT	DECMP	DEZCP	DMOVR	DRAD
DDEG	DEBCD	DEBIN	DEADD	DESUB
DEMUL	DEDIV	DEXP	DLN	DLOG
DESQR	DPOW	INT	DSIN	DCOS
DTAN	DASIN	DACOS	DATAN	DADDR
DSUBR	DMULR	DDIVR		

Двоичный формат чисел с плавающей запятой

В контроллерах Delta DVP операции с плавающей запятой осуществляются в соответствии со стандартом IEEE754 по следующей схеме:



Мантисса – часть числа с плавающей запятой, содержащая числа после запятой.
Экспонента – показатель степени числа, в которую нужно возвести основание системы счисления, чтобы получить данное число.

Для двоичной системы пересчет числа в число с плавающей запятой осуществляется по следующей формуле:

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

Где S – знак числа, 2 – основание системы счисления (двоичная), M – мантисса, E – экспонента, B – константа равная числу 127.

Получается следующий диапазон 32-х разрядного числа с плавающей запятой:

$\pm 2^{-128} \sim \pm 2^{+128}$, или в десятичном формате $\pm 1.1755 \times 10^{-38} \sim \pm 3.4028 \times 10^{+38}$

Пример 1: В качестве примера переведем десятичное число +23,0 в 32-х разрядное число с плавающей запятой.

Шаг 1. Преобразуем десятичное число 23 в двоичное: 23,0 = 10111

Шаг 2. Нормализация: $10111 = 1.0111 \times 2^4$, где 0111 мантисса, а 4 экспонента

Шаг 3. Получаем экспоненту для числа с плавающей запятой:
∵ E – B = 4 → E – 127 = 4 ∴ E = 131 = 10000011_2

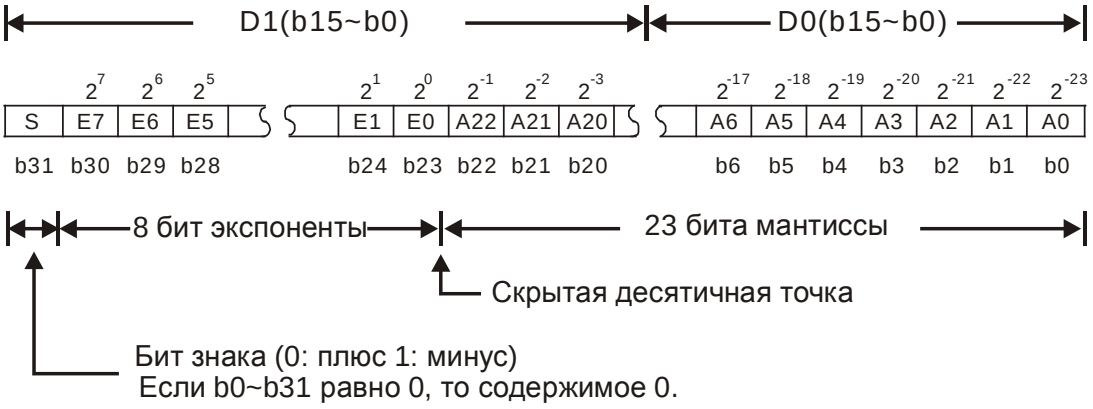
Шаг 4. Соединяем вместе бит знака, экспоненту и мантиссу:

0 100**0001**1 011**1000**0000**0000**0000**0000**$_2$ = $41B80000_{16}$

Пример 2: Переведем "-23.0" в 32-разрядное число с плавающей запятой
Если нужно преобразовать отрицательное число -23,0, то для этого необходимо произвести все те же действия, но в бит знака записать «1».

1 100**0001**1 011**1000**0000**0000**0000**0000**$_2$=$C1B80000_{16}$

Для хранения числа с плавающей запятой используются два последовательных регистра, например (D1, D0). Распределение битов на примере данных двух регистров показано ниже:



Десятичный формат чисел с плавающей запятой
Десятичное число с плавающей запятой занимает два последовательных регистра, например (D1, D0), и будет иметь следующий вид:
Десятичное число с плавающей запятой = [мантисса D0] × 10 [экспонента D1]
Мантисса D0 = ±1000 ~ ±9999
Экспонента D1 = - 41 ~ +35
Диапазон десятичного числа с плавающей запятой: $\pm 1175 \times 10^{-41} \sim \pm 3402 \times 10^{+35}$.
Мантисса 100 не существует в регистре D0, так как 100 представляется как 1000×10^{-1}.

Инструкции для работы в десятичном формате с плавающей запятой:

- D EBCD: Конвертация двоичного числа с плавающей запятой в десятичное число с плавающей запятой
- D EBIN: Конвертация десятичного числа с плавающей запятой в двоичное число с плавающей запятой

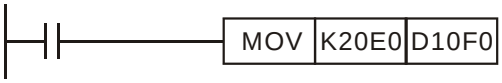
В операциях с плавающей запятой используются следующие флаги:

- Флаг нуля: M1020 = 1, если результат операции равен «0».
- Флаг заимствования: M1021 = 1, если результат операции превосходит минимальное значение.
- Флаг переноса: M1022 = 1, если абсолютное значение результата операции выходит за допустимый диапазон.

Правила работы с индексными регистрами E и F

Индексные регистры служат для динамического изменения адреса какого-либо операнда путем прибавления значения индексного регистра к значению операнда.
Индексные регистры имеют разрядность 16 бит. Если необходимо использовать индекс с разрядностью 32 бит, то индексы E и индекс F используются совместно. В индексе E будут храниться младшие 16 бит, а в индексе F будут храниться старшие 16 бит. Само 32-х разрядное значение записывается в индекс E, который при этом перекроет индекс F с таким же номером. В данном случае соответствующий индекс F будет уже не доступен. Комбинации 32-х разрядных индексных регистров будут следующие: (E0, F0), (E1, F1), (E2, F2), … (E7, F7). Рекомендуется использовать команду MOVP для сброса регистров D в 0 при включении контроллера





Например, E0 = 8 и K20E0 означает константу K28 (20 + 8). Если условие выполняется (маркер=1) то константа K28 передаётся в регистр D24.

E0 = 8 F0 = 14

20 + 8 = 28 10 + 14 = 24

Передача K28 → D24

Индексные регистры E и F могут изменять адреса операндов, M: P, X, Y, M, S, KnX, KnY, KnM, KnS, T, C, D, но не могут изменять себя, использоваться отдельно и изменять коэффициент Kn. Операция K4M0E0 является допустимой, а K0E0M0 недопустимой. В начале описания каждой прикладной команды приводится сводная таблица, где серым цветом выделены ячейки с операндами, которые могут использоваться с индексными регистрами.
Для изменения регистров P, I, X, Y, M, S, KnX, KnY, KnM, KnS, T, C и D с использованием E, F, необходимо выбрать 16 битовый регистр, то есть E или F.

3.6 Перечень команд (классификация по функциям)

Контроль цикла

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
00	CJ	-	✓	Переход к заданной строке	✓	✓	✓	✓	3	-
01	CALL	-	✓	Переход к подпрограмме	✓	✓	✓	✓	3	-
02	SRET	-	-	Конец подпрограммы	✓	✓	✓	✓	1	-
03	IRET	-	-	Конец обработки прерывания	✓	✓	✓	✓	1	-
04	EI	-	-	Разрешение прерывания	✓	✓	✓	✓	1	-
05	DI	-	-	Запрет прерывания	✓	✓	✓	✓	1	-
06	FEND	-	-	Завершение основной программы (первое завершение)	✓	✓	✓	✓	1	-
07	WDT	-	✓	Сброс сторожевого таймера	✓	✓	✓	✓	1	-
08	FOR	-	-	Начало цикла FOR-NEXT	✓	✓	✓	✓	3	-
09	NEXT	-	-	Конец цикла FOR-NEXT	✓	✓	✓	✓	1	-

Пересылка и сравнение

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	16 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
10	CMP	DCMP	✓	Сравнение данных	✓	✓	✓	✓	7	13
11	ZCP	DZCP	✓	Зонное сравнения данных	✓	✓	✓	✓	9	17
12	MOV	DMOV	✓	Пересылка данных	✓	✓	✓	✓	5	9
13	SMOV	-	✓	Пересылка со смещением	✓	✓	✓	✓	11	-
14	CML	DCML	✓	Пересылка с инвертированием данных	✓	✓	✓	✓	5	9
15	BMOV	-	✓	Пересылка блока данных	✓	✓	✓	✓	7	-
16	FMOV	DFMOV	✓	Пересылка в несколько адресов	✓	✓	✓	✓	7	13
17	XCH	DXCH	✓	Обмен данными	✓	✓	✓	✓	5	9
18	BCD	DBCD	✓	Преобразование двоичных чисел в десятичные	✓	✓	✓	✓	5	9
19	BIN	DBIN	✓	Преобразование десятичных чисел в двоичные	✓	✓	✓	✓	5	9

Арифметические действия

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	16 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
20	ADD	DADD	✓	Сложение	✓	✓	✓	✓	7	13
21	SUB	DSUB	✓	Вычитание	✓	✓	✓	✓	7	13
22	MUL	DMUL	✓	Умножение	✓	✓	✓	✓	7	13
23	DIV	DDIV	✓	Деление	✓	✓	✓	✓	7	13
24	INC	DINC	✓	Увеличение на 1	✓	✓	✓	✓	3	5
25	DEC	DDEC	✓	Уменьшение на 1	✓	✓	✓	✓	3	5
26	WAND	DAND	✓	Логическое «И»	✓	✓	✓	✓	7	13
27	WOR	DOR	✓	Логическое «ИЛИ»	✓	✓	✓	✓	7	13
28	WXOR	DXOR	✓	Логическое «ИСКЛ. ИЛИ»	✓	✓	✓	✓	7	13
29	NEG	DNEG	✓	Логическое отрицание	✓	✓	✓	✓	3	5

Сдвиг и перемещение

API	Команды		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
30	ROR	DROR	✓	Кольцевой сдвиг вправо	✓	✓	✓	✓	5	9
31	ROL	DROL	✓	Кольцевой сдвиг влево	✓	✓	✓	✓	5	9

API	Команды		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
32	RCR	DRCR	✓	Кольцевой сдвиг вправо с переносом	✓	✓	✓	✓	5	9
33	RCL	DRCL	✓	Кольцевой сдвиг влево с переносом	✓	✓	✓	✓	5	9
34	SFTR	-	✓	Сдвиг значений битовых операндов вправо	✓	✓	✓	✓	9	-
35	SFTL	-	✓	Сдвиг значений битовых операндов влево	✓	✓	✓	✓	9	-
36	WSFR	-	✓	Пословный сдвиг значений регистра вправо	✓	✓	✓	✓	9	-
37	WSFL	-	✓	Пословный сдвиг значений регистра влево	✓	✓	✓	✓	9	-
38	SFWR	-	✓	Запись данных в стек	✓	✓	✓	✓	7	-
39	SFRD	-	✓	Чтение данных из стека	✓	✓	✓	✓	7	-
40	ZRST	-	✓	Сброс операндов в заданном диапазоне	✓	✓	✓	✓	5	-
41	DECO	-	✓	Дешифратор 8 – 256 бит	✓	✓	✓	✓	7	-
42	ENCO	-	✓	Шифратор 256 – 8 бит	✓	✓	✓	✓	7	-
43	SUM	DSUM	✓	Сумма активных битов	✓	✓	✓	✓	5	9
44	BON	DBON	✓	Проверка состояния битов	✓	✓	✓	✓	7	13
45	MEAN	DMEAN	✓	Среднее арифметическое	✓	✓	✓	✓	7	13
46	ANS	-	-	Тревожная сигнализация с задержкой на включение	✓	✓	✓	✓	7	-
47	ANR	-	✓	Сброс тревожной сигнализации	✓	✓	✓	✓	1	-
48	SQR	DSQR	✓	Квадратный корень	✓	✓	✓	✓	5	9
49	FLT	DFLT	✓	Преобразование целого в число с плавающей запятой	✓	✓	✓	✓	5	9

Операции с данным
Высокоскоростные команды

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
50	REF	-	✓	Обновление состояния входов/выходов	✓	✓	✓	✓	5	-
51	REFF	-	✓	Изменение времени входного фильтра	✓	✓	✓	✓	3	-
52	MTR	-	-	Матричный ввод	✓	✓	✓	✓	9	-
53	-	DHSCS	-	Включение выхода при скоростном счете	✓	✓	✓	✓	-	13
54	-	DHSCR	-	Выключение выхода при скоростном счете	✓	✓	✓	✓	-	13
55	-	DHSZ	-	Зонное сравнение при скоростном счете	✓	✓	✓	✓	-	17
56	SPD	-	-	Вычисление скорости	✓	✓	✓	✓	7	-
57	PLSY	DPLSY	-	Выдача определенного числа импульсов	✓	✓	✓	✓	7	13
58	PWM	-	-	Выдача импульсов с ШИМ	✓	✓	✓	✓	7	-
59	PLSR	DPLSR	-	Импульсный выход с ускорением/замедлением	✓	✓	✓	✓	9	17

Пользовательские команды

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
60	IST	-	-	Ручное/автоматическое управление	✓	✓	✓	✓	7	-
61	SER	DSER	✓	Поиск данных стека	-	✓	✓	✓	9	17
62	ABSD	DABSD	-	Абсолютный многоустановочный счетчик	-	✓	✓	✓	9	17
63	INCD	-	-	Инкрементный многоустановочный счетчик	-	✓	✓	✓	9	-

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
64	TTMR	-	-	Обучающийся таймер	-	✓	✓	✓	5	-
65	STMR	-	-	Специальный таймер	-	✓	✓	✓	7	-
66	ALT	-	✓	Импульсное реле (T-триггер)	✓	✓	✓	✓	3	-
67	RAMP	DRAMP	-	Линейное изменение сигнала (RAMP)	-	✓	✓	✓	9	17
68	DTM	-	✓	Изменение и перемещение данных	-	✓	✓	✓	9	-
69	SORT	DSORT	-	Сортировка данных	-	✓	✓	✓	11	21

Команды ввода/вывода

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
70	TKY	DTKY	-	Ввод с 10-ти кнопочной клавиатуры	-	✓	✓	✓	7	13
71	HKY	DHKY	-	Ввод с 16-ти кнопочной клавиатуры	-	✓	✓	✓	9	17
72	DSW	-	-	Ввод с цифрового переключателя	-	✓	✓	✓	9	-
73	SEGD	-	✓	Дешифратор для 7-ми сегментного индикатора	✓	✓	✓	✓	5	-
74	SEGL	-	-	Вывод на 7-ми сегментный индикатор	✓	✓	✓	✓	7	-
75	ARWS	-	-	Ввод со стрелочной клавиатуры	-	✓	✓	✓	9	-
76	ASC	-	-	ASCII - конвертирование	-	✓	✓	✓	11	-
77	PR	-	-	Выдача ASCII-знаков на выходы	-	✓	✓	✓	5	-

Команды последовательного ввода/вывода

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
78	FROM	DFROM	✓	Чтение данных из модулей аналогового ввода/вывода (CR-регистры)	✓	✓	✓	✓	9	17
79	TO	DTO	✓	Запись данных в модули аналогового ввода/вывода (CR-регистры)	✓	✓	✓	✓	9	17
80	RS	-	-	Передача/прием данных по RS-435	✓	✓	✓	✓	9	-
81	PRUN	DPRUN	✓	Пересылка данных восьмиричном формате	-	✓	✓	✓	5	9
82	ASCII	-	✓	Перевод HEX в ASCII	✓	✓	✓	✓	7	-
83	HEX	-	✓	Перевод ASCII в HEX	✓	✓	✓	✓	7	-
84	CCD	-	✓	Контрольная сумма	-	✓	✓	✓	7	-
85	VRRD	-	✓	Чтение значения, заданного с потенциометра	-	-	✓	✓	5	-
86	VRSC	-	✓	Масштаб значения потенциометра	-	-	✓	✓	5	-
87	ABS	DABS	✓	Абсолютное значение	✓	✓	✓	✓	3	5
88	PID	DPID	-	ПИД-регулятор	✓	✓	✓	✓	9	17

Базовые команды

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
89	PLS	-	-	Создание импульса по переднему фронту	✓	✓	✓	✓	3	-
90	LDP	-	-	Логическое выражение с опросом по переднему фронту	✓	✓	✓	✓	3	-
91	LDF	-	-	Логическое выражение с опросом по заднему фронту	✓	✓	✓	✓	3	-
92	ANDP	-	-	«И» с опросом по переднему фронту	✓	✓	✓	✓	3	-

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
93	ANDF	-	-	«И» с опросом по заднему фронту	✓	✓	✓	✓	3	-
94	ORP	-	-	«ИЛИ» с опросом по переднему фронту	✓	✓	✓	✓	3	-
95	ORF	-	-	«ИЛИ» с опросом по заднему фронту	✓	✓	✓	✓	3	-
96	TMR	-	-	Таймер	✓	✓	✓	✓	4	-
97	CNT	DCNT	-	Счетчик	✓	✓	✓	✓	4	6
98	INV	-	-	Инверсия	✓	✓	✓	✓	1	-
99	PLF	-	-	Создание импульса по заднему фронту	✓	✓	✓	✓	3	-

Команды MODBUS

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
100	MODRD	-	-	Чтение данных Modbus	✓	✓	✓	✓	7	-
101	MODWR	-	-	Запись данных Modbus	✓	✓	✓	✓	7	-
102	FWD	-	-	Пуск вперед для VFD	✓	✓	✓	✓	7	–
103	REV	-	-	Пуск назад для VFD	✓	✓	✓	✓	7	–
104	STOP	-	-	Стоп для VFD	✓	✓	✓	✓	7	–
105	RDST	-	-	Чтение текущего состояния VFD	✓	✓	✓	✓	5	–
106	RSTEF	-	-	Команда сброс для VFD	✓	✓	✓	✓	5	–
107	LRC	-	✓	Контрольная сумма LRC	✓	✓	✓	✓	7	-
108	CRC	-	✓	Контрольная сумма CRC	✓	✓	✓	✓	7	-
150	MODRW	-	-	MODBUS чтение/запись	✓	✓	✓	✓	11	-
206	ASDRW	-	-	Чтение/запись для сервопривода ASDA	-	✓	✓	✓	7	-

Команды для чисел с плавающей запятой

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
110	-	DECMP	✓	Сравнение чисел с плавающей запятой	✓	✓	✓	✓	-	13
111	-	DEZCP	✓	Зонное сравнение чисел с плавающей запятой	✓	✓	✓	✓	-	17
112		DMOVR	✓	Перемещение числа с плавающей запятой	✓	✓	✓	✓		9
116	-	DRAD	✓	Перевод градусов в радианы	✓	✓	✓	✓	-	9
117	-	DDEG	✓	Перевод радианов в градусы	✓	✓	✓	✓	-	9
118	-	DEBCD	✓	Перевод двоичного в десятичное	✓	✓	✓	✓	-	9
119	-	DEBIN	✓	Перевод десятичного в двоичное	✓	✓	✓	✓	-	9
120	-	DEADD	✓	Сложение чисел с плавающей запятой	✓	✓	✓	✓	-	13
121	-	DESUB	✓	Вычитание чисел с плавающей запятой	✓	✓	✓	✓	-	13
122	-	DEMUL	✓	Умножение чисел с плавающей запятой	✓	✓	✓	✓	-	13
123	-	DEDIV	✓	Деление чисел с плавающей запятой	✓	✓	✓	✓	-	13
124	-	DEXP	✓	Вычисление экспоненты в формате с плавающей запятой	✓	✓	✓	✓	-	9
125	-	DLN	✓	Вычисление натурального логарифма в формате с плавающей запятой	✓	✓	✓	✓	-	9
126	-	DLOG	✓	Вычисление логарифма в формате с плавающей запятой	✓	✓	✓	✓	-	13
127	-	DESQR	✓	Вычисление квадратного корня в формате с плавающей запятой	✓	✓	✓	✓	-	9
128	-	DPOW	✓	Возведение числа в степень в формате с плавающей запятой	✓	✓	✓	✓	-	13

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
129	INT	DINT	✓	Преобразование числа с плавающей запятой в целое	✓	✓	✓	✓	5	9
130	-	DSIN	✓	Вычисление синуса	✓	✓	✓	✓	-	9
131	-	DCOS	✓	Вычисление косинуса	✓	✓	✓	✓	-	9
132	-	DTAN	✓	Вычисление тангенса	✓	✓	✓	✓	-	9
133	-	DASIN	✓	Вычисление арксинуса	✓	✓	✓	✓	-	9
134	-	DACOS	✓	Вычисление арккосинуса	✓	✓	✓	✓	-	9
135	-	DATAN	✓	Вычисление арктангенса	✓	✓	✓	✓	-	9
172	-	DADDR	✓	Сложение с использованием регистра	✓	✓	✓	✓	-	13
173	-	DSUBR	✓	Вычитание с использованием регистра	✓	✓	✓	✓	-	13
174	-	DMULR	✓	Умножение с использованием регистра	✓	✓	✓	✓	-	13
175	-	DDIVR	✓	Деление с использованием регистра	✓	✓	✓	✓	-	13

Дополнительные команды

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
143	DELAY	-	✓	Задержка выполнения	✓	✓	✓	✓	3	-
144	GPWM	-	-	Общая команда генерации импульсов ШИМ	✓	✓	✓	✓	7	-
147	SWAP	DSWAP	✓	Перестановка старшего и младшего бита в регистре	✓	✓	✓	✓	3	5
148	MEMR	-	✓	Чтение данных из файлового регистра	✓	-	✓/-	✓	7	-
149	MEMW	-	✓	Запись данных в файловый регистр	✓	-	✓/-	✓	7	-
151	PWD	-	-	Определение длительности входного импульса (для EH2/EH3 и SV/SV2)	-	-	-	-	5	-
154	RAND	DRAND	✓	Генератор случайных чисел	✓	✓	✓	✓	7	13
168	MVM	DMVM	✓	Маска и комбинация заданных битов	✓	✓	✓	✓	7	13
176	MMOV	–	✓	Перевод 16 бит в 32 бит	✓	✓	✓	✓	5	–
177	GPS	-	-	Прием GPS-данных	✓	✓	✓	✓	5	-
178	-	DSPA	-	Установка солнечных батарей	✓	✓	✓	✓	–	9
179	WSUM	DWSUM	✓	Суммирование данных с нескольких устройств	✓	✓	✓	✓	7	13
202	SCAL	-	✓	Расчет пропорционального значения	✓	✓	✓	✓	9	-
203	SCLP	DSCLP	✓	Параметры расчета пропорционального значения	✓	✓	✓	✓	9	13
205	CMPT	-	✓	Таблица сравнения	✓	✓	✓	✓	9	-
207	CSFO	-	-	Определение частоты сигнала на входе и масштабирование ее на выходе	✓	✓	✓	✓	7	-
258	ATMR	-	-	Таймер контактного типа	✓	✓	✓	✓	5	-

Команды позиционирования

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
155	-	DABSR	-	Чтение абсолютного текущего положения	✓	✓	✓	✓	-	13
156	-	DZRN	-	Возврат в исходную позицию	✓	✓	✓	✓	-	17
157	-	DPLSV		Импульсный выход с заданием частоты и направления вращения серводвигателя	✓	✓	✓	✓	-	13
158	-	DDRVI	-	Управление положением в относительных координатах	✓	✓	✓	✓	-	17
159	-	DDRVA	-	Управление положением в абсолютных координатах	✓	✓	✓	✓	-	17

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
191	-	DPPMR	-	Двухосевая относительная синхронизация по двум точкам	✓	-	✓	✓	-	17
192	-	DPPMA	-	Двухосевая абсолютная синхронизация по двум точкам	✓	-	✓	✓	-	17
193	-	DCIMR	-	Двухосевая относительная дуговая интерполяция	✓	-	✓	✓	-	17
194	-	DCIMA	-	Двухосевая абсолютная дуговая интерполяция	✓	-	✓	✓	-	17
195	-	DPTPO	-	Таблица одноосевого импульсного выхода	✓	✓	✓	✓	-	13
197	-	DCLLM	-	Управление замкнутой системой позиционирования	✓	✓	✓	✓	-	17
198	-	DVSPO	-	Переменная скорость на импульсном выходе	✓	✓	✓	✓	-	17
199	-	DICF	✓	Мгновенное изменение частоты	✓	✓	✓	✓	-	13

Команды реального времени

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
160	TCMP	-	✓	Сравнение времени	✓	✓	✓	✓	11	-
161	TZCP	-	✓	Сравнение времени в заданном диапазоне	✓	✓	✓	✓	9	-
162	TADD	-	✓	Сложение времени	✓	✓	✓	✓	7	-
163	TSUB	-	✓	Вычитание времени	✓	✓	✓	✓	7	-
166	TRD	-	✓	Чтение текущего значения реального времени	✓	✓	✓	✓	3	-
167	TWR	-	✓	Изменение значения реального времени	✓	✓	✓	✓	3	-
169	HOUR	DHOUR	-	Счетчик времени наработки	✓	✓	✓	✓	7	13

Код Грея

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
170	GRY	DGRY	✓	Преобразование числа в код Грея	✓	✓	✓	✓	5	9
171	GBIN	DGBIN	✓	Преобразование кода Грея в число	✓	✓	✓	✓	5	9

Операции с матрицами

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
180	MAND	-	✓	Логическое И для матриц	✓	✓	✓	✓	9	-
181	MOR	-	✓	Логическое ИЛИ для матриц	✓	✓	✓	✓	9	-
182	MXOR	-	✓	Исключающее ИЛИ для матриц	✓	✓	✓	✓	9	-
183	MXNR	-	✓	Исключающее НЕ-ИЛИ для матриц	✓	✓	✓	✓	9	-
184	MINV	-	✓	Инверсия матрицы	✓	✓	✓	✓	7	-
185	MCMP	-	✓	Сравнение матриц	✓	✓	✓	✓	9	-
186	MBRD	-	✓	Чтение битов в матрице	✓	✓	✓	✓	7	-
187	MBWR	-	✓	Запись битов в матрице	✓	✓	✓	✓	7	-
188	MBS	-	✓	Сдвиг битов в матрице	✓	✓	✓	✓	7	-
189	MBR	-	✓	Кольцевой сдвиг битов в матрице	✓	✓	✓	✓	7	-
190	MBC	-	✓	Счетчик битов	✓	✓	✓	✓	7	-

Логические операции контактного Типа

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
215	LD&	DLD&	-	Контакт замкнут, если S_1 & $S_2 \neq 0$	✓	✓	✓	✓	5	9
216	LD|	DLD|	-	Контакт замкнут, если S_1 | $S_2 \neq 0$	✓	✓	✓	✓	5	9
217	LD^	DLD^	-	Контакт замкнут, если S_1 ^ $S_2 \neq 0$	✓	✓	✓	✓	5	9
218	AND&	DAND&	-	Последовательный контакт замкнут, если S_1 & $S_2 \neq 0$	✓	✓	✓	✓	5	9
219	AND|	DAND|	-	Последовательный контакт замкнут, если S_1 | $S_2 \neq 0$	✓	✓	✓	✓	5	9
220	AND^	DAND^	-	Последовательный контакт замкнут, если S_1 ^ $S_2 \neq 0$	✓	✓	✓	✓	5	9
221	OR&	DOR&	-	Параллельный контакт замкнут, если S_1 & $S_2 \neq 0$	✓	✓	✓	✓	5	9
222	OR|	DOR|	-	Параллельный контакт замкнут, если S_1 | $S_2 \neq 0$	✓	✓	✓	✓	5	9
223	OR^	DOR^	-	Параллельный контакт замкнут, если S_1 ^ $S_2 \neq 0$	✓	✓	✓	✓	5	9

Операции сравнения контактов

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
224	LD=	DLD=	-	Контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	5	9
225	LD>	DLD>	-	Контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	5	9
226	LD<	DLD<	-	Контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	5	9
228	LD<>	DLD<>	-	Контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	5	9
229	LD<=	DLD<=	-	Контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	5	9
230	LD>=	DLD>=	-	Контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	5	9
232	AND=	DAND=	-	Последовательный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	5	9
233	AND>	DAND>	-	Последовательный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	5	9
234	AND<	DAND<	-	Последовательный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	5	9
236	AND<>	DAND<>	-	Последовательный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	5	9
237	AND<=	DAND<=	-	Последовательный контакт замкнут, если $S1 \leqq S2$	✓	✓	✓	✓	5	9
238	AND>=	DAND>=	-	Последовательный контакт замкнут, если $S1 \geqq S2$	✓	✓	✓	✓	5	9
240	OR=	DOR=	-	Параллельный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	5	9
241	OR>	DOR>	-	Параллельный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	5	9
242	OR<	DOR<	-	Параллельный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	5	9
244	OR<>	DOR<>	-	Параллельный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	5	9
245	OR<=	DOR<=	-	Параллельный контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	5	9
246	OR>=	DOR>=	-	Параллельный контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	5	9

Управление битами

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
266	BOUT	DBOUT	-	Переключение заданного бита в слове	✓	✓	✓	✓	5	9
267	BSET	DBSET	-	Переключение заданного бита в слове с фиксацией	✓	✓	✓	✓	5	9
268	BRST	DBRST	-	Сброс заданного бита в слове	✓	✓	✓	✓	5	9
269	BLD	DBLD	-	Установка H/O контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
270	BLDI	DBLDI	-	Установка Н/З контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
271	BAND	DBAND	-	Последовательное подключение Н/О контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
272	BANI	DBANI	-	Последовательное подключение Н/З контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
273	BOR	DBOR	-	Параллельное подключение Н/О контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
274	BORI	DBORI	-	Параллельное подключение Н/З контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9

Операции сравнения контактов с плавающей запятой

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
275	-	FLD=	-	Контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	-	9
276	-	FLD>	-	Контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	-	9
277	-	FLD<	-	Контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	-	9
278	-	FLD<>	-	Контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	-	9
279	-	FLD<=	-	Контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	-	9
280	-	FLD>=	-	Контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	-	9
281	-	FAND=	-	Последовательный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	-	9
282	-	FAND>	-	Последовательный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	-	9
283	-	FAND<	-	Последовательный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	-	9
284	-	FAND<>	-	Последовательный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	-	9
285	-	FAND<=	-	Последовательный контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	-	9
286	-	FAND>=	-	Последовательный контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	-	9
287	-	FOR=	-	Параллельный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	-	9
288	-	FOR>	-	Параллельный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	-	9
289	-	FOR<	-	Параллельный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	-	9
290	-	FOR<>	-	Параллельный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	-	9
291	-	FOR<=	-	Параллельный контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	-	9
292	-	FOR>=	-	Параллельный контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	-	9
296	LDZ＞	DLDZ＞	-	Контакт замкнут, если $\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
297	LDZ＞＝	DLDZ＞＝	-	Контакт замкнут, если $\lvert S_1 - S_2 \rvert \geq \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
298	LDZ<	DLDZ＜	-	Контакт замкнут, если $\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
299	LDZ＜＝	DLDZ＜＝	-	Контакт замкнут, если $\lvert S_1 - S_2 \rvert \leq \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
300	LDZ＝	DLDZ＝	-	Контакт замкнут, если $\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
301	LDZ＜＞	DLDZ＜＞	-	Контакт замкнут, если $\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
302	ANDZ＞	DANDZ＞	-	Последовательный контакт замкнут, если $\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
303	ANDZ＞＝	DANDZ＞＝	-	Последовательный контакт замкнут, если $\lvert S_1 - S_2 \rvert \geq \lvert S_3 \rvert$	✓	✓	✓	✓	7	13
304	ANDZ＜	DANDZ＜	-	Последовательный контакт замкнут, если $\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	✓	✓	✓	✓	7	13

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
305	ANDZ<=	DANDZ<=	-	Последовательный контакт замкнут, если | S_1 - S_2 | ≤| S_3 |	✓	✓	✓	✓	7	13
306	ANDZ=	DANDZ=	-	Последовательный контакт замкнут, если | S_1 - S_2 |=| S_3 |	✓	✓	✓	✓	7	13
307	ANDZ<>	DANDZ<>	-	Последовательный контакт замкнут, если | S_1 - S_2 | ≠ | S_3 |	✓	✓	✓	✓	7	13
308	ORZ >	DORZ >	-	Параллельный контакт замкнут, если | S_1 - S_2 | > | S_3 |	✓	✓	✓	✓	7	13
309	ORZ > =	DORZ > =	-	Параллельный контакт замкнут, если | S_1 - S_2 | ≧ | S_3 |	✓	✓	✓	✓	7	13
310	ORZ <	DORZ <	-	Параллельный контакт замкнут, если | S_1 - S_2 | < | S_3 |	✓	✓	✓	✓	7	13
311	ORZ < =	DORZ < =	-	Параллельный контакт замкнут, если | S_1 - S_2 | ≦ | S_3 |	✓	✓	✓	✓	7	13
312	ORZ =	DORZ =	-	Параллельный контакт замкнут, если | S_1 - S_2 | = | S_3 |	✓	✓	✓	✓	7	13
313	ORZ < >	DORZ < >	-	Параллельный контакт замкнут, если | S_1 - S_2 | ≠ | S_3 |	✓	✓	✓	✓	7	13

3.7 Перечень команд (в алфавитном порядке)

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
87	ABS	DABS	✓	Абсолютное значение	✓	✓	✓	✓	3	5
62	ABSD	DABSD	-	Абсолютный многоустановочный счетчик	-	✓	✓	✓	9	17
20	ADD	DADD	✓	Сложение	✓	✓	✓	✓	7	13
66	ALT	-	✓	Импульсное реле (Т-триггер)	✓	✓	✓	✓	3	-
218	AND&	DAND&	-	Последовательный контакт замкнут, если S_1 & S_2 ≠ 0	✓	✓	✓	✓	5	9
220	AND^	DAND^	-	Последовательный контакт замкнут, если S_1 ^ S_2 ≠ 0	✓	✓	✓	✓	5	9
219	AND|	DAND|	-	Последовательный контакт замкнут, если S_1 | S_2 ≠ 0	✓	✓	✓	✓	5	9
234	AND<	DAND<	-	Последовательный контакт замкнут, если S_1 < S_2	✓	✓	✓	✓	5	9
237	AND<=	DAND<=	-	Последовательный контакт замкнут, если S_1 ≦ S_2	✓	✓	✓	✓	5	9
236	AND<>	DAND<>	-	Последовательный контакт замкнут, если S_1 ≠ S_2	✓	✓	✓	✓	5	9
232	AND=	DAND=	-	Последовательный контакт замкнут, если S_1 = S_2	✓	✓	✓	✓	5	9
233	AND>	DAND>	-	Последовательный контакт замкнут, если S_1 > S_2	✓	✓	✓	✓	5	9
238	AND>=	DAND>=	-	Последовательный контакт замкнут, если S_1 ≧ S_2	✓	✓	✓	✓	5	9
93	ANDF	-	-	«И» с опросом по заднему фронту	✓	✓	✓	✓	3	-
92	ANDP	-	-	«И» с опросом по переднему фронту	✓	✓	✓	✓	3	-
47	ANR	-	✓	Сброс тревожной сигнализации	✓	✓	✓	✓	1	-
302	ANDZ>	DANDZ>	-	Последовательный контакт замкнут, если | S_1 - S_2 |>| S_3 |	✓	✓	✓	✓	7	13
303	ANDZ>=	DANDZ>=	-	Последовательный контакт замкнут, если | S_1 - S_2 | ≥| S_3 |	✓	✓	✓	✓	7	13
304	ANDZ<	DANDZ<	-	Последовательный контакт замкнут, если | S_1 - S_2 |<| S_3 |	✓	✓	✓	✓	7	13
305	ANDZ<=	DANDZ<=	-	Последовательный контакт замкнут, если | S_1 - S_2 | ≤| S_3 |	✓	✓	✓	✓	7	13
306	ANDZ=	DANDZ=	-	Последовательный контакт замкнут, если | S_1 - S_2 |=| S_3 |	✓	✓	✓	✓	7	13
307	ANDZ<>	DANDZ<>	-	Последовательный контакт замкнут, если | S_1 - S_2 | ≠ | S_3 |	✓	✓	✓	✓	7	13
46	ANS	-	-	Тревожная сигнализация с задержкой на включение	✓	✓	✓	✓	7	-

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
75	ARWS	-	-	Ввод со стрелочной клавиатуры	-	✓	✓	✓	9	-
76	ASC	-	-	ASCII - конвертирование	-	✓	✓	✓	11	-
82	ASCII	-	✓	Перевод HEX в ASCII	✓	✓	✓	✓	7	-
206	ASDRW	-	-	Чтение/запись для сервопривода ASDA	✓	✓	✓	✓	7	-
258	ATMR	-	-	Таймер контактного типа	✓	✓	✓	✓	5	-
271	BAND	DBAND	-	Последовательное подключение Н/О контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
272	BANI	DBANI	-	Последовательное подключение Н/З контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
18	BCD	DBCD	✓	Преобразование двоичных чисел в десятичные	✓	✓	✓	✓	5	9
19	BIN	DBIN	✓	Преобразование десятичных чисел в двоичные	✓	✓	✓	✓	5	9
269	BLD	DBLD	-	Установка Н/О контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
270	BLDI	DBLDI	-	Установка Н/З контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
15	BMOV	-	✓	Пересылка блока данных	✓	✓	✓	✓	7	-
44	BON	DBON	✓	Проверка состояния битов	✓	✓	✓	✓	7	13
273	BOR	DBOR	-	Параллельное подключение Н/О контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
274	BORI	DBORI	-	Параллельное подключение Н/З контакта с состоянием по заданному биту	✓	✓	✓	✓	5	9
266	BOUT	DBOUT	-	Переключение заданного бита в слове	✓	✓	✓	✓	5	9
268	BRST	DBRST	-	Сброс заданного бита в слове	✓	✓	✓	✓	5	9
267	BSET	DBSET	-	Переключение заданного бита в слове с фиксацией	✓	✓	✓	✓	5	9
01	CALL	-	✓	Переход к подпрограмме	✓	✓	✓	✓	3	-
84	CCD	-	✓	Контрольная сумма	-	✓	✓	✓	7	-
00	CJ	-	✓	Переход к заданной строке	✓	✓	✓	✓	3	-
14	CML	DCML	✓	Пересылка с инвертированием данных	✓	✓	✓	✓	5	9
10	CMP	DCMP	✓	Сравнение данных	✓	✓	✓	✓	7	13
205	CMPT	-	✓	Таблица сравнения	✓	✓	✓	✓	9	-
97	CNT	DCNT	-	Счетчик	✓	✓	✓	✓	4	6
108	CRC	-	✓	Контрольная сумма CRC	✓	✓	✓	✓	7	-
207	CSFO	-	-	Определение частоты сигнала на входе и масштабирование ее на выходе	✓	✓	✓	✓	7	-
25	DEC	DDEC	✓	Уменьшение на 1	✓	✓	✓	✓	3	5
41	DECO	-	✓	Дешифратор 8 – 256 бит	✓	✓	✓	✓	7	-
143	DELAY	-	✓	Задержка выполнения	✓	✓	✓	✓	3	-
05	DI	-	-	Запрет прерывания	✓	✓	✓	✓	1	-
23	DIV	DDIV	✓	Деление	✓	✓	✓	✓	7	13
72	DSW	-	-	Ввод с цифрового переключателя	-	✓	✓	✓	9	-
68	DTM	-	✓	Изменение и перемещение данных	-	✓	✓	✓	9	-
04	EI	-	-	Разрешение прерывания	✓	✓	✓	✓	1	-
42	ENCO	-	✓	Шифратор 256 – 8 бит	✓	✓	✓	✓	7	-
06	FEND	-	-	Завершение основной программы (Первое завершение)	✓	✓	✓	✓	1	-
49	FLT	DFLT	✓	Преобразование целого в число с плавающей запятой	✓	✓	✓	✓	5	9
16	FMOV	DFMOV	✓	Пересылка в несколько адресов	✓	✓	✓	✓	7	13
08	FOR	-	-	Начало цикла FOR-NEXT	✓	✓	✓	✓	3	-
78	FROM	DFROM	✓	Чтение данных из модулей	✓	✓	✓	✓	9	17

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
				аналогового ввода/вывода (CR-регистры)						
102	FWD	-	-	Пуск вперед для VFD	✓	✓	✓	✓	7	–
171	GBIN	DGBIN	✓	Преобразование кода Грея в число	✓	✓	✓	✓	5	9
177	GPS	-	-	Прием GPS-данных	✓	✓	✓	✓	5	-
144	GPWM	-	-	Общая команда генерации импульсов ШИМ	✓	✓	✓	✓	7	-
170	GRY	DGRY	✓	Преобразование числа в код Грея	✓	✓	✓	✓	5	9
83	HEX	-	✓	Перевод ASCII в HEX	✓	✓	✓	✓	7	-
71	HKY	DHKY	-	Ввод с 16-ти кнопочной клавиатуры	-	✓	✓	✓	9	17
169	HOUR	DHOUR	-	Счетчик времени наработки	✓	✓	✓	✓	7	13
24	INC	DINC	✓	Увеличение на 1	✓	✓	✓	✓	3	5
63	INCD	-	-	Инкрементный многоустановочный счетчик	-	✓	✓	✓	9	-
129	INT	DINT	✓	Преобразование числа с плавающей точкой в целое	✓	✓	✓	✓	5	9
98	INV	-	-	Инверсия	✓	✓	✓	✓	1	-
03	IRET	-	-	Конец обработки прерывания	✓	✓	✓	✓	1	-
60	IST	-	-	Ручное/автоматическое управление	✓	✓	✓	✓	7	-
215	LD&	DLD&	-	Контакт замкнут, если S_1 & $S_2 \neq 0$	✓	✓	✓	✓	5	9
217	LD^	DLD^	-	Контакт замкнут, если S_1 ^ $S_2 \neq 0$	✓	✓	✓	✓	5	9
216	LD|	DLD|	-	Контакт замкнут, если S_1 | $S_2 \neq 0$	✓	✓	✓	✓	5	9
226	LD<	DLD<	-	Контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	5	9
229	LD<=	DLD<=	-	Контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	5	9
228	LD<>	DLD<>	-	Контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	5	9
224	LD=	DLD=	-	Контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	5	9
225	LD>	DLD>	-	Контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	5	9
230	LD>=	DLD>=	-	Контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	5	9
91	LDF	-	-	Логическое выражение с опросом по заднему фронту	✓	✓	✓	✓	3	-
90	LDP	-	-	Логическое выражение с опросом по переднему фронту	✓	✓	✓	✓	3	-
296	LDZ>	DLDZ>	-	Контакт замкнут, если | S_1 - S_2 |>| S_3 |	✓	✓	✓	✓	7	13
297	LDZ>=	DLDZ>=	-	Контакт замкнут, если | S_1 - S_2 | ≥ | S_3 |	✓	✓	✓	✓	7	13
298	LDZ<	DLDZ<	-	Контакт замкнут, если | S_1 - S_2 |<| S_3 |	✓	✓	✓	✓	7	13
299	LDZ<=	DLDZ<=	-	Контакт замкнут, если | S_1 - S_2 | ≤ | S_3 |	✓	✓	✓	✓	7	13
300	LDZ=	DLDZ=	-	Контакт замкнут, если | S_1 - S_2 |=| S_3 |	✓	✓	✓	✓	7	13
301	LDZ<>	DLDZ<>	-	Контакт замкнут, если | S_1 - S_2 | ≠ | S_3 |	✓	✓	✓	✓	7	13
107	LRC	-	✓	Контрольная сумма LRC	✓	✓	✓	✓	7	-
180	MAND	-	✓	Логическое И для матриц	✓	✓	✓	✓	9	-
190	MBC	-	✓	Счетчик битов	✓	✓	✓	✓	7	-
189	MBR	-	✓	Кольцевой сдвиг битов в матрице	✓	✓	✓	✓	7	-
186	MBRD	-	✓	Чтение битов в матрице	✓	✓	✓	✓	7	-
188	MBS	-	✓	Сдвиг битов в матрице	✓	✓	✓	✓	7	-
187	MBWR	-	✓	Запись битов в матрице	✓	✓	✓	✓	7	-
185	MCMP	-	✓	Сравнение матриц	✓	✓	✓	✓	9	-
45	MEAN	DMEAN	✓	Среднее арифметическое	✓	✓	✓	✓	7	13
148	MEMR	-	✓	Чтение данных из файлового регистра	✓	-	✓/-	✓	7	-
149	MEMW	-	✓	Запись данных в файловый регистр	✓	-	✓/-	✓	7	-

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
184	MINV	-	✓	Инверсия матрицы	✓	✓	✓	✓	7	-
176	MMOV	-	✓	Перевод 16 бит в 32 бит	✓	✓	✓	✓	5	-
100	MODRD	-	-	Чтение данных Modbus	✓	✓	✓	✓	7	-
150	MODRW	-	-	MODBUS чтение/запись	✓	✓	✓	✓	11	-
101	MODWR	-	-	Запись данных Modbus	✓	✓	✓	✓	7	-
181	MOR	-	✓	Логическое ИЛИ для матриц	✓	✓	✓	✓	9	-
12	MOV	DMOV	✓	Пересылка данных	✓	✓	✓	✓	5	9
52	MTR	-	-	Матричный ввод	✓	✓	✓	✓	9	-
22	MUL	DMUL	✓	Умножение	✓	✓	✓	✓	7	13
168	MVM	DMVM	✓	Маска и комбинация заданных битов	✓	✓	✓	✓	7	13
183	MXNR	-	✓	Исключающее НЕ-ИЛИ для матриц	✓	✓	✓	✓	9	-
182	MXOR	-	✓	Исключающее ИЛИ для матриц	✓	✓	✓	✓	9	-
29	NEG	DNEG	✓	Отрицание	✓	✓	✓	✓	3	5
09	NEXT	-	-	Конец цикла FOR-NEXT	✓	✓	✓	✓	1	-
221	OR&	DOR&	-	Параллельный контакт замкнут, если S_1 & $S_2 \neq 0$	✓	✓	✓	✓	5	9
223	OR^	DOR^	-	Параллельный контакт замкнут, если S_1 ^ $S_2 \neq 0$	✓	✓	✓	✓	5	9
222	OR|	DOR|	-	Параллельный контакт замкнут, если S_1 | $S_2 \neq 0$	✓	✓	✓	✓	5	9
242	OR<	DOR<	-	Параллельный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	5	9
245	OR<=	DOR<=	-	Параллельный контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	5	9
244	OR<>	DOR<>	-	Параллельный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	5	9
240	OR=	DOR=	-	Параллельный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	5	9
241	OR>	DOR>	-	Параллельный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	5	9
246	OR>=	DOR>=	-	Параллельный контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	5	9
95	ORF	-	-	«ИЛИ» с опросом по заднему фронту	✓	✓	✓	✓	3	-
94	ORP	-	-	«ИЛИ» с опросом по переднему фронту	✓	✓	✓	✓	3	-
308	ORZ >	DORZ＞	-	Параллельный контакт замкнут, если | S_1 - S_2 | > | S_3 |	✓	✓	✓	✓	7	13
309	ORZ > =	DORZ＞＝	-	Параллельный контакт замкнут, если | S_1 - S_2 | ≧ | S_3 |	✓	✓	✓	✓	7	13
310	ORZ <	DORZ＜	-	Параллельный контакт замкнут, если | S_1 - S_2 | < | S_3 |	✓	✓	✓	✓	7	13
311	ORZ < =	DORZ＜＝	-	Параллельный контакт замкнут, если | S_1 - S_2 | ≦ | S_3 |	✓	✓	✓	✓	7	13
312	ORZ =	DORZ＝	-	Параллельный контакт замкнут, если | S_1 - S_2 | = | S_3 |	✓	✓	✓	✓	7	13
313	ORZ < >	DORZ＜＞	-	Параллельный контакт замкнут, если | S_1 - S_2 | ≠ | S_3 |	✓	✓	✓	✓	7	13
88	PID	DPID	-	ПИД-регулятор	✓	✓	✓	✓	9	17
99	PLF	-	-	Создание импульса по заднему фронту	✓	✓	✓	✓	3	-
89	PLS	-	-	Создание импульса по переднему фронту	✓	✓	✓	✓	3	-
59	PLSR	DPLSR	-	Импульсный выход с ускорением/замедлением	✓	✓	✓	✓	9	17
57	PLSY	DPLSY	-	Выдача определенного числа импульсов	✓	✓	✓	✓	7	13
77	PR	-	-	Выдача ASCII-знаков на выходы	-	✓	✓	✓	5	-
81	PRUN	DPRUN	✓	Пересылка данных в восьмиричном формате	-	✓	✓	✓	5	9
151	PWD	-	-	Определение длительности входного импульса	-	-	-	-	5	-

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
58	PWM	-	-	Выдача импульсов с ШИМ	✓	✓	✓	✓	7	-
67	RAMP	DRAMP	-	Линейное изменение сигнала (RAMP)	-	✓	✓	✓	9	17
154	RAND	DRAND	✓	Генератор случайных чисел	✓	✓	✓	✓	7	13
33	RCL	DRCL	✓	Кольцевой сдвиг влево с переносом	✓	✓	✓	✓	5	9
32	RCR	DRCR	✓	Кольцевой сдвиг вправо с переносом	✓	✓	✓	✓	5	9
105	RDST	-	-	Чтение текущего состояния VFD	✓	✓	✓	✓	5	–
50	REF	-	✓	Обновление состояния входов/выходов	✓	✓	✓	✓	5	-
51	REFF	-	✓	Изменение времени входного фильтра	✓	✓	✓	✓	3	-
103	REV	-	-	Пуск назад для VFD	✓	✓	✓	✓	7	–
31	ROL	DROL	✓	Кольцевой сдвиг влево	✓	✓	✓	✓	5	9
30	ROR	DROR	✓	Кольцевой сдвиг вправо	✓	✓	✓	✓	5	9
80	RS	-	-	Передача/прием данных по RS-435	✓	✓	✓	✓	9	-
106	RSTEF	-	-	Команда сброс для VFD	✓	✓	✓	✓	5	–
202	SCAL	-	✓	Расчет пропорционального значения	✓	✓	✓	✓	9	-
203	SCLP	DSCLP	✓	Параметры расчета пропорционального значения	✓	✓	✓	✓	7	13
73	SEGD	-	✓	Дешифратор для 7-ми сегментного индикатора	✓	✓	✓	✓	5	-
74	SEGL	-	-	Вывод на 7-ми сегментный индикатор	✓	✓	✓	✓	7	-
61	SER	DSER	✓	Поиск данных стека	-	✓	✓	✓	9	17
39	SFRD	-	✓	Чтение данных из стека	✓	✓	✓	✓	7	-
35	SFTL	-	✓	Сдвиг значений битовых операндов влево	✓	✓	✓	✓	9	-
34	SFTR	-	✓	Сдвиг значений битовых операндов вправо	✓	✓	✓	✓	9	-
38	SFWR	-	✓	Запись данных в стек	✓	✓	✓	✓	7	-
13	SMOV	-	✓	Пересылка со смещением	✓	✓	✓	✓	11	-
69	SORT	DSORT	-	Сортировка данных	-	✓	✓	✓	11	21
56	SPD	-	-	Вычисление скорости	✓	✓	✓	✓	7	-
48	SQR	DSQR	✓	Квадратный корень	✓	✓	✓	✓	5	9
02	SRET	-	-	Конец подпрограммы	✓	✓	✓	✓	1	-
65	STMR	-	-	Специальный таймер	-	✓	✓	✓	7	-
104	STOP	-	-	Стоп для VFD	✓	✓	✓	✓	7	–
21	SUB	DSUB	✓	Вычитание	✓	✓	✓	✓	7	13
43	SUM	DSUM	✓	Сумма битов в регистре	✓	✓	✓	✓	5	9
147	SWAP	DSWAP	✓	Перестановка старшего и младшего бита в регистре	✓	✓	✓	✓	3	5
162	TADD	-	✓	Сложение времени	✓	✓	✓	✓	7	-
160	TCMP	-	✓	Сравнение времени	✓	✓	✓	✓	11	-
70	TKY	DTKY	-	Ввод с 10-ти кнопочной клавиатуры	-	✓	✓	✓	7	13
96	TMR	-	-	Таймер	✓	✓	✓	✓	4	-
79	TO	DTO	✓	Запись данных в модули аналогового ввода/вывода (CR-регистры)	✓	✓	✓	✓	9	17
166	TRD	-	✓	Чтение текущего значения реального времени	✓	✓	✓	✓	3	-
163	TSUB	-	✓	Вычитание времени	✓	✓	✓	✓	7	-
64	TTMR	-	-	Обучающий таймер	-	✓	✓	✓	5	-
167	TWR	-	✓	Изменение значения реального времени	✓	✓	✓	✓	3	-
161	TZCP	-	✓	Сравнение времени в заданном диапазоне	✓	✓	✓	✓	9	-

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
85	VRRD	-	✓	Чтение значения, заданного с потенциометра	-	-	✓	✓	5	-
86	VRSC	-	✓	Масштаб значения потенциометра	-	-	✓	✓	5	-
26	WAND	DAND	✓	Логическое «И»	✓	✓	✓	✓	7	13
07	WDT	-	✓	Сброс сторожевого таймера	✓	✓	✓	✓	1	-
27	WOR	DOR	✓	Логическое «ИЛИ»	✓	✓	✓	✓	7	13
37	WSFL	-	✓	Сдвиг значений регистра влево	✓	✓	✓	✓	9	-
36	WSFR	-	✓	Сдвиг значений регистра вправо	✓	✓	✓	✓	9	-
179	WSUM	DWSUM	✓	Суммирование данных с нескольких устройств	✓	✓	✓	✓	7	13
28	WXOR	DXOR	✓	Логическое «ИСКЛ.ИЛИ»	✓	✓	✓	✓	7	13
17	XCH	DXCH	✓	Обмен данными	✓	✓	✓	✓	5	9
11	ZCP	DZCP	✓	Зонное сравнения данных	✓	✓	✓	✓	9	17
40	ZRST	-	✓	Сброс операндов в заданном диапазоне	✓	✓	✓	✓	5	-
155	-	DABSR	-	Чтение абсолютного текущего положения	✓	✓	✓	✓	-	13
134	-	DACOS	✓	Вычисление арккосинуса	✓	✓	✓	✓	-	9
172	-	DADDR	✓	Сложение с использованием регистра	✓	✓	✓	✓	-	13
133	-	DASIN	✓	Вычисление арксинуса	✓	✓	✓	✓	-	9
135	-	DATAN	✓	Вычисление арктангенса	✓	✓	✓	✓	-	9
194	-	DCIMA	-	Двухосевая абсолютная дуговая интерполяция	✓	-	✓	✓	-	17
193	-	DCIMR	-	Двухосевая относительная дуговая интерполяция	✓	-	✓	✓	-	17
197	-	DCLLM	-	Управление замкнутой системой позиционирования	✓	✓	✓	✓	-	17
131	-	DCOS	✓	Вычисление косинуса	✓	✓	✓	✓	-	9
117	-	DDEG	✓	Перевод радианов в градусы	✓	✓	✓	✓	-	9
175	-	DDIVR	✓	Деление с использованием регистра	✓	✓	✓	✓	-	13
159	-	DDRVA	-	Управление положением в абсолютных координатах	✓	✓	✓	✓	-	17
158	-	DDRVI	-	Управление положением в относительных координатах	✓	✓	✓	✓	-	17
120	-	DEADD	✓	Сложение чисел с плавающей запятой	✓	✓	✓	✓	-	13
118	-	DEBCD	✓	Перевод двоичного в десятичное	✓	✓	✓	✓	-	9
119	-	DEBIN	✓	Перевод десятичного в двоичное	✓	✓	✓	✓	-	9
110	-	DECMP	✓	Сравнение двух чисел с плавающей запятой	✓	✓	✓	✓	-	13
123	-	DEDIV	✓	Деление чисел с плавающей запятой	✓	✓	✓	✓	-	13
122	-	DEMUL	✓	Умножение чисел с плавающей запятой	✓	✓	✓	✓	-	13
127	-	DESQR	✓	Вычисление квадратного корня в формате с плавающей запятой	✓	✓	✓	✓	-	9
121	-	DESUB	✓	Вычитание чисел с плавающей запятой	✓	✓	✓	✓	-	13
124	-	DEXP	✓	Вычисление экспоненты в формате с плавающей запятой	✓	✓	✓	✓	-	9
111	-	DEZCP	✓	Зонное сравнение двух чисел с плавающей запятой	✓	✓	✓	✓	-	17
54	-	DHSCR	-	Сброс состояния выхода при скоростном счете	✓	✓	✓	✓	-	13
53	-	DHSCS	-	Установка состояния выхода при скоростном счете	✓	✓	✓	✓	-	13
55	-	DHSZ	-	Зонное сравнение при скоростном счете	✓	✓	✓	✓	-	17
199	-	DICF	✓	Мгновенное изменение частоты	✓	✓	✓	✓	-	13
125	-	DLN	✓	Вычисление натурального логарифма в формате с плавающей запятой	✓	✓	✓	✓	-	9
126	-	DLOG	✓	Вычисление логарифма в	✓	✓	✓	✓	-	13

API	Команда		Импульс	Функция	Применимо к контроллеру				Число шагов	
	16 бит	32 бит			ES2 EX2	SS2	SA2 SE	SX2	16-бит	32-бит
				формате с плавающей запятой						
112	-	DMOVR	✓	Перемещение данных с плавающей запятой	✓	✓	✓	✓	-	9
174	-	DMULR	✓	Умножение с использованием регистра	✓	✓	✓	✓	-	13
157	-	DPLSV	-	Импульсный выход с заданием частоты и направления вращения серводвигателя	✓	✓	✓	✓	-	13
128	-	DPOW	✓	Возведение числа в степень в формате с плавающей точкой	✓	✓	✓	✓	-	13
192	-	DPPMA	-	Двухосевая абсолютная синхронизация по двум точкам	✓	-	✓	✓	-	17
191	-	DPPMR	-	Двухосевая относительная синхронизация по двум точкам	✓	-	✓	✓	-	17
195	-	DPTPO	-	Таблица одноосевого импульсного выхода	✓	✓	✓	✓	-	13
116	-	DRAD	✓	Перевод градусов в радианы	✓	✓	✓	✓	-	9
130	-	DSIN	✓	Вычисление синуса	✓	✓	✓	✓	-	9
178	-	DSPA	-	Установка солнечных батарей	✓	✓	✓	✓	–	9
173	-	DSUBR	✓	Вычитание с использованием регистра	✓	✓	✓	✓	-	13
132	-	DTAN	✓	Вычисление тангенса	✓	✓	✓	✓	-	9
198	-	DVSPO	-	Переменная скорость на импульсном выходе	✓	✓	✓	✓	-	17
156	-	DZRN	-	Возврат в исходную позицию	✓	✓	✓	✓	-	17
283	-	FAND<	-	Последовательный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	-	9
285	-	FAND<=	-	Последовательный контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	-	9
284	-	FAND<>	-	Последовательный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	-	9
280	-	FAND=	-	Последовательный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	-	9
282	-	FAND>	-	Последовательный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	-	9
286	-	FAND>=	-	Последовательный контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	-	9
277	-	FLD<	-	Контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	-	9
279	-	FLD<=	-	Контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	-	9
278	-	FLD<>	-	Контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	-	9
275	-	FLD=	-	Контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	-	9
276	-	FLD>	-	Контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	-	9
280	-	FLD>=	-	Контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	-	9
289	-	FOR<	-	Параллельный контакт замкнут, если $S_1 < S_2$	✓	✓	✓	✓	-	9
291	-	FOR<=	-	Параллельный контакт замкнут, если $S_1 \leqq S_2$	✓	✓	✓	✓	-	9
290	-	FOR<>	-	Параллельный контакт замкнут, если $S_1 \neq S_2$	✓	✓	✓	✓	-	9
287	-	FOR=	-	Параллельный контакт замкнут, если $S_1 = S_2$	✓	✓	✓	✓	-	9
288	-	FOR>	-	Параллельный контакт замкнут, если $S_1 > S_2$	✓	✓	✓	✓	-	9
292	-	FOR>=	-	Параллельный контакт замкнут, если $S_1 \geqq S_2$	✓	✓	✓	✓	-	9

3.8 Детальное описание команд

API	Команда		Операнды	Функция	Контроллеры			
00	CJ	P	(S)	Условный переход	ES2/EX2	SS2	SA2/ SE	SX2

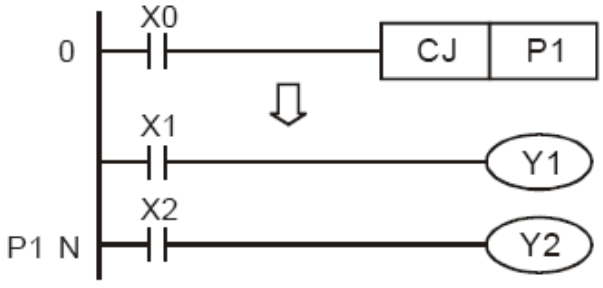
Операнд	Диапазон	Шаги программы
(S)	P0~P255	CJ, CJP: 3 шага

PULSE				16-bit				32-bit			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Указатель точки условного перехода P.

Описание:
1. С помощью CJ-инструкции может пропускаться часть программы. При применении этой инструкции время выполнения программы может уменьшаться.
2. Цель (конец) перехода определяется установкой указателя (точки перехода) в программе.
3. Указание адреса конца перехода (Адреса точки) определяет, к какому указателю должен выполняться переход. CJ-инструкции могут использовать один адрес перехода многократно в программе, однако использование одного и того же указателя инструкциями CJ и CALL недопустимо. Это вызовет ошибку в программе.
4. Обратный переход (вверх программы) также может выполняться внутри программы. Если входной сигнал для CJ-инструкции держится больше 200 мс, то появляется ошибка времени работы (Watchdog Timer).
5. Имеется возможность дублирования записи выхода. При дублировании записи выходов следите за тем, чтобы оба выхода никогда не были активными в одно и то же время. Это может привести к ошибочной отработке программы.
6. Изменение состояния устройств внутри перехода CJ:
   6.1. Y, M, S сохранят свое состояние, которое было до перехода CJ;
   6.2. Выполнение таймеров (100мс, 10 мс) прекратится (накопленное время сохраняется);
   6.3. Таймеры T192-199 и их рабочие контакты будут продолжать работать;
   6.4. Высокоскоростные счетчики и их рабочие контакты будут продолжать работать;
   6.5. Обычные счетчики прекратят работу (накопленное значение сохраняется);
   6.6. Если внутри подпрограммы перехода программируется инструкция сброса (отключения) для аккумулятивного таймера, то процесс сброса (стирание накопленного значения) имеет место тогда, когда перепрыгивается цепь схемы катушки счетчика;
   6.7. Прикладные инструкции (кроме DHSCS, DHSCR, DHSZ, SPD, PLSY, PWM, PLSR, PLSV, DRVI, DRVA,) внутри перехода выполняться не будут.

Пример применения 1:

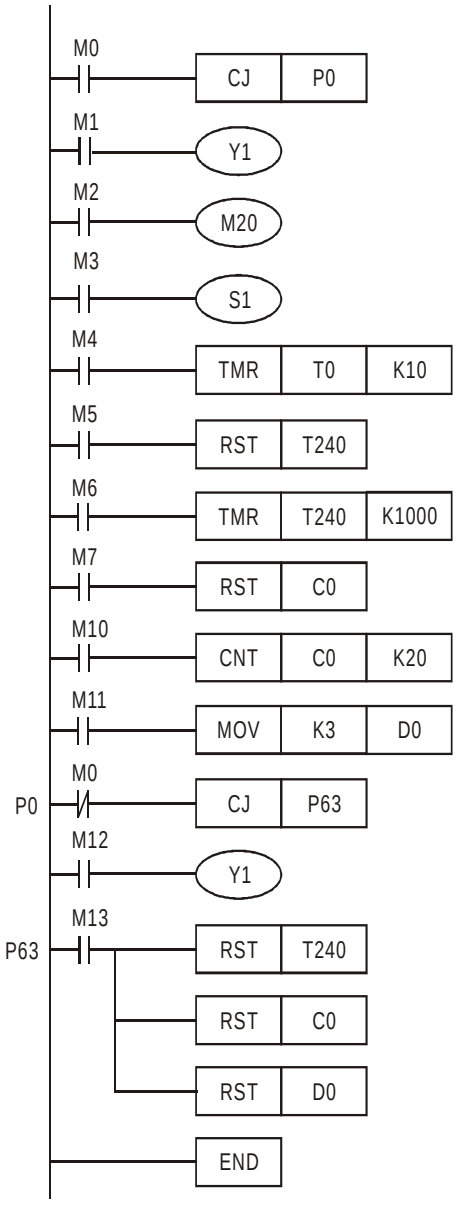


Маркировка точки выполняется при программировании на языке IL (Список инструкций) непосредственно перед цепью схемы (перед инструкциями LD и LDI). При программировании на языке контактной схемы маркировка точки указывается слева от цепи схемы.

Если включается X0, то выполняется переход к точке P1 и строки программы между 0 и N выполняться не будут. Если X0 выключено, то программа выполняется полностью.

Пример применения 2:
Состояния устройств поясняют выполнение инструкции.

Операнды	Состояние контакта до выполнения CJ	Состояние контакта в ходе выполнения CJ	Состояние выходной катушки в ходе выполнения CJ
Y, M, S	M1, M2, M3 ВЫКЛ.	M1, M2, M3 ВЫКЛ.→ВКЛ.	Y1 *1, M20, S1 ВЫКЛ.
	M1, M2, M3 ВКЛ.	M1, M2, M3 ВКЛ.→ВЫКЛ.	Y1 *1, M20, S1 ВКЛ.
Таймер*2 10мс, 100мс	M4 ВЫКЛ.	M4 ВЫКЛ.→ВКЛ.	Таймер не активируется
	M4 ВКЛ.	M4 ВКЛ.→ВЫКЛ.	Таймер T0 останавливается, его состояние запоминается. Когда M0 ВКЛ. → ВЫКЛ., T0 таймер сбрасывается
Аккумулятивный таймер 1мс,10мс, 100мс	M6 ВЫКЛ.	M6 ВЫКЛ.→ВКЛ.	Таймер T240 не активируется
	M6 ВКЛ.	M6 ВКЛ.→ВЫКЛ.	Таймер T240 останавливается, его состояние запоминается. Когда M0 ВКЛ. → ВЫКЛ., его состояние сохраняется
C0~C234 *3	M7, M10 ВЫКЛ.	M10 переключится ВКЛ./ВЫКЛ.	СчётчикC0 останавливается
	M7 ВЫКЛ., M10 переключится ВКЛ./ВЫКЛ.	M10 переключится ВКЛ./ВЫКЛ.	СчётчикC0 останавливается, его значение запоминается. Когда M0 -.> ВЫКЛ., то C0 продолжает счёт
Выполнение команд	M11 ВЫКЛ.	M11 ВЫКЛ.→ВКЛ.	Инструкции не будут выполняться
	M11 ВКЛ.	M11 ВКЛ.→ВЫКЛ.	Пропущенные инструкции не будут выполняться, но API 53~59, API 157~159 будут продолжать выполнение



*1: Y1 включается в двух случаях: когда M0 = ВЫКЛ., M1 включает Y1, когда M0 = ВКЛ.,
M12 включает Y1

*2: Когда используемый подпрограммой таймер (T184~T199) сначала запускается, а затем выполняется инструкция CJ, то таймер продолжает счёт и, при достижении значения уставки, происходит замыкание выходного контакта

*3: Когда сначала высокоскоростной счётчик запускается (C235~C254) а затем выполняется CJ инструкция, счёт продолжается, и состояние его выхода не меняется.

API	Команда		Операнды	Функция	Контроллеры			
01	CALL	P	(S)	Вызов подпрограммы	ES2/EX2	SS2	SA2/ SE	SX2

Операнд	Диапазон	Шаги программы
(S)	P0~P255	CALL, CALLP: 3 шага

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Указатель точки перехода к подпрограмме.

Описание:
1. Если активируется CALL-команда, то выполняется переход к указанной точке маркировки.
2. CALL команда должна использоваться вместе с командами FEND (API06) и SRET (API02).
3. Подпрограмма программируется после FEND-команды и перед END-командой. После отработки SRET-команды выполняется обратный переход в главную программу к команде, следующей за CALL-командой.

Заметки:
1. Подпрограмма размещается после FEND команды.
3. Подпрограмма заканчивается SRET командой
4. Использование одинаковых указателей командами CJ и CALL недопустимо. Это вызовет ошибку в программе.
5. CALL может обращаться к одной и той же подпрограмме любое число раз.
6. Внутри подпрограммы могут вызываться другие подпрограммы. Возможно максимум 5 уровней вложенности, включая начальную подпрограмму.

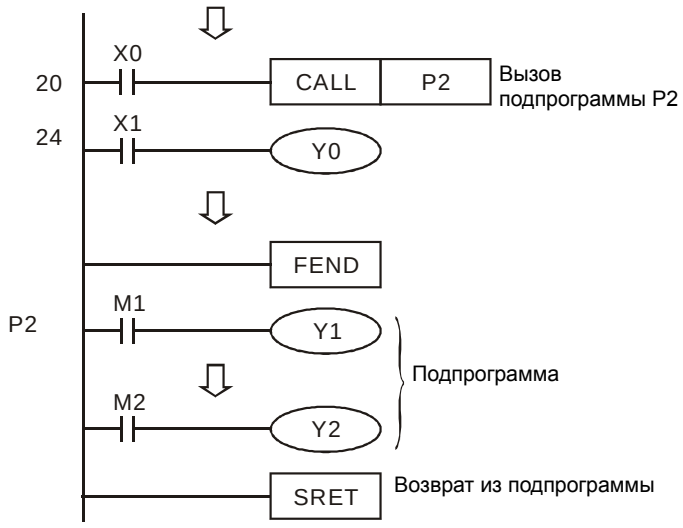
API	Команда	Функция	Контроллеры			
02	SRET	Возврат в основную программу	ES2/EX2	SS2	SA2/ SE	SX2

Операнд	Описание	Шаги программы
НЕТ	Контакт для условия выполнения команды не требуется Автоматический возврат программы после выполнения по адресу, следующему за командой CALL в O100.	SRET: 1 шаг

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Описание:
SRET определяет конец подпрограммы. После выполнения подпрограммы происходит возврат в основную программу и выполнение её с адреса, следующего за командой CALL.

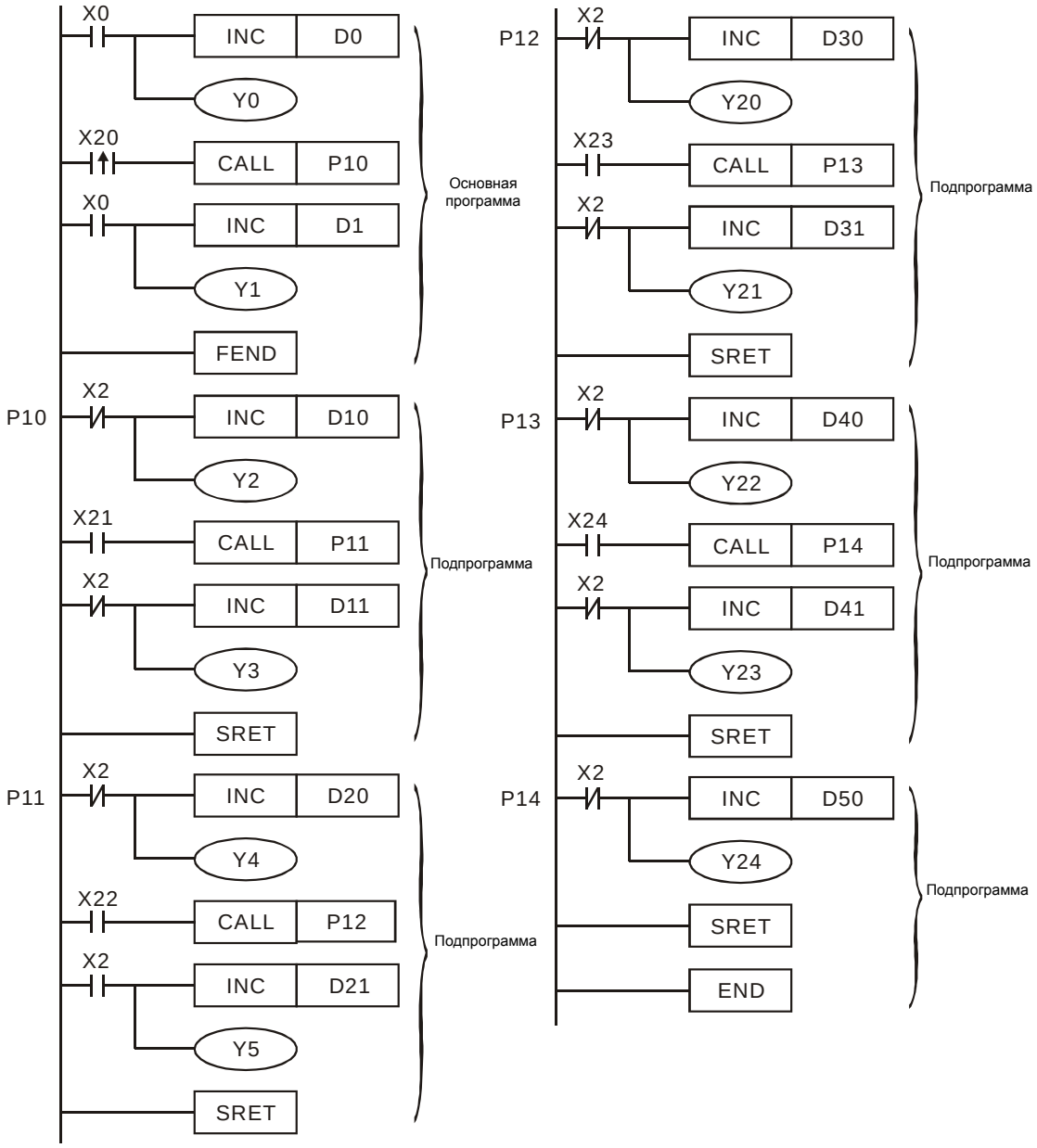
Пример 1:



Когда X0 = ВКЛ., команда CALL прерывает ход основной программы и запускает выполнение подпрограммы P2.
После её выполнения команда SRET обеспечивает переход по адресу 24 и продолжается выполнение основной программы.

Пример 2:

1. По переднему фронту при переключении X20, вызывается команда CALL P10, и управление передаётся подпрограмме P10.
2. Когда X21 =ВКЛ., выполняется команда CALL P11, происходит переход к подпрограмме P11 и её выполнение.
3. Когда X22 = ВКЛ., выполняется команда CALL P12, происходит переход к подпрограмме P12 и её выполнение.
4. Когда X23 = ВКЛ., выполняется команда CALL P13, происходит переход к подпрограмме P13 и её выполнение.
5. Когда X24 = ВКЛ., выполняется команда CALL P14, происходит переход к подпрограмме P13 и её выполнение.
6. При достижении команды SRET происходит возврат к предыдущей подпрограмме P для завершения цикла.
7. Выполнение подпрограммы будет производиться до самого верхнего уровня, пока команда SRET в подпрограмме P10 не будет выполнена. Далее выполнение программы вернётся к основной программе.



API	Команда	Функция	Контроллеры			
03	IRET	Конец обработки прерывания	ES2/EX2	SS2	SA2	SX2

Операнд	Описание	Шаги программы
НЕТ	Контакт для условия выполнения команды не требуется. IRET завершает исполнение подпрограммы прерывания и выполняет возврат в основную программу.	IRET: 1 шаг

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2	SX2	ES2/EX2	SS2	SA2	SX2	ES2/EX2	SS2	SA2	SX2

API	Команда	Функция	Контроллеры			
04	EI	Разрешение прерывания	ES2/EX2	SS2	SA2/ SE	SX2

Операнд	Описание	Шаги программы
НЕТ	Контакт для условия выполнения команды не требуется. Разрешение прерывания, описание этой команды совпадает с описание DI (команда запрета прерывания), см. команду DI для получения дополнительной информации. M1050~M1059	EI: 1 шаг

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

API	Команда	Функция	Контроллеры			
05	DI	Запрет прерывания	ES2/EX2	SS2	SA2/ SE	SX2

Операнд	Описание	Шаги программы
НЕТ	Контакт для условия выполнения команды не требуется. Команда DI запрещает контроллеру обрабатывать запросы на прерывание. Когда специальные внутренние маркеры M1050 ~ M1059, служащие для запрета прерывания, включены, то соответствующие запросы на прерывания не будут выполняться, даже в случае разрешения прерываний.	DI: 1 шаг

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Описание:

1. Команда EI разрешает выполнение подпрограмм прерывания таких, как внешнее прерывание, прерывание по таймеру и по высокоскоростному счётчику.
2. При выполнении программы запуск подпрограмм прерывания будет разрешен между командами EI и DI. Если в вашей программе нет частей, требующих запрета на обработку прерываний, то команда DI может быть удалена.
3. Подпрограммы прерываний должны быть размещены после команды FEND.
4. Во время выполнения подпрограммы прерывания другие прерывания не выполняются. При наличии нескольких запросов на прерывание приоритет получает первый пришедший. Если происходит ещё несколько запросов на прерывание одновременно, то приоритет имеет то, которое имеет меньший адрес точки прерывания.
5. Запросы на прерывание приходящие между командами DI и EI не будут выполняться сразу, а запомнятся и будут выполняться после следующей команды EI.
6. При использовании указателя точки прерывания нельзя последовательно с высокоскоростным счётчиком использовать опрос соответствующего дискретного входа X.
7. Если во время выполнения подпрограммы прерывания требуется немедленно обеспечить ввод или вывод, то необходимо в программе использовать команду REF для обновления статуса входов или выходов

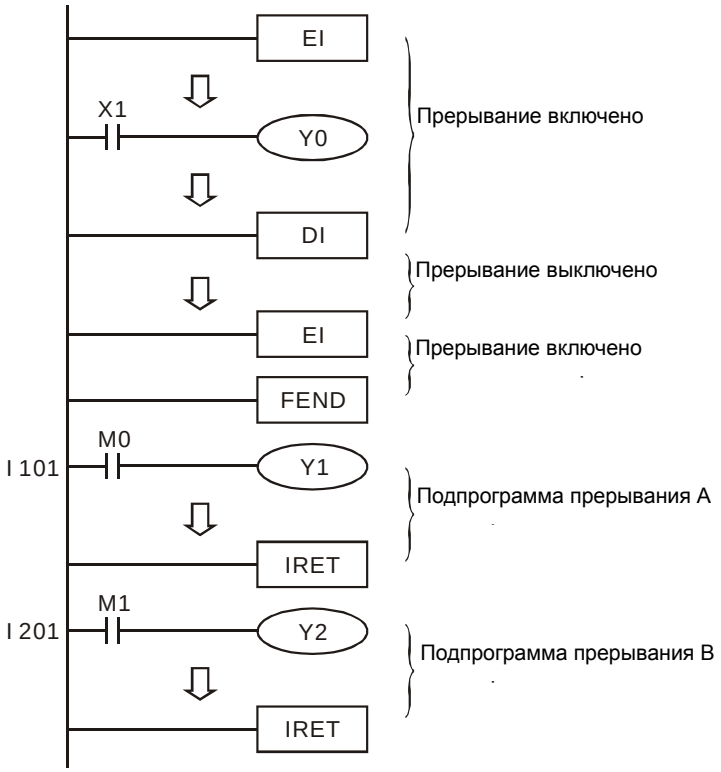
Примечания:
Точки прерывания (I):

a) Внешние прерывания: 8 точек, включая (I000/I001, X0), (I100/I101, X1), (I200/I201, X2), (I300/I301, X3), (I400/I401, X4), (I500/I501, X5), (I600/I601, X6) и (I700/I701, X7)
(00 прерывание по заднему фронту, 01 прерывание по переднему фронту)
b) Прерывания по таймеру: 2 точки, включая I605~I699 и I705~I799 (дискретность 1мс)
c) Прерывания по высокоскоростному счётчику: 8 точек, включая I010, I020, I030, I040, I050, I060, I070 и I080 (используются с командой API 53 DHSCS для генерирования сигналов прерывания).
d) Коммуникационные прерывания: 3 точки I140, I150 и I160
e) Используемые флаги:

Флаг	Функция
M1050	Запрет внешнего прерывания I000 / I001
M1051	Запрет внешнего прерывания I100 / I101
M1052	Запрет внешнего прерывания I200 / I201
M1053	Запрет внешнего прерывания I300 / I301
M1054	Запрет внешнего прерывания I400 / I401
M1055	Запрет внешнего прерывания I500 / I501, I600 / I601, I700 / I701
M1056	Запрет прерывания от таймера I605~I699
M1057	Запрет прерывания от таймера I705~I799
M1059	Запрет прерываний от высокоскоростного счётчика I010~I080
M1280	I000/I001 Инвертирование логики привязки к фронту (передний/задний).
M1284	I400/I401 Инвертирование логики привязки к фронту (передний/задний).
M1286	I600/I601 Инвертирование логики привязки к фронту (передний/задний).

Внимание:
По умолчанию прерывание I000(X0) привязано заднему фронту. Когда M1280=ВКЛ. и выполняется команда EI, контроллер изменит функцию входа X0 так, что привязка будет производиться по переднему фронту.
При необходимости повторного изменения привязки, сначала надо перевести флаг M1280 в состояние ВЫКЛ. и выполнить команду DI. После этого, при повторном выполнении команды EI прерывание I000(X0) вновь будет привязано к заднему фронту.

Пример программы:
В ходе работы ПЛК программа сканирует команды между EI и DI. Если X1 или X2 включены, подпрограммы A или B будут прерваны. Когда IRET будет достигнут, выполнение основной программы продолжится.



API	Команда	Функция	Контроллеры			
06	FEND	Конец основной программы (Первое завершение)	ES2/EX2	SS2	SA2/ SE	SX2

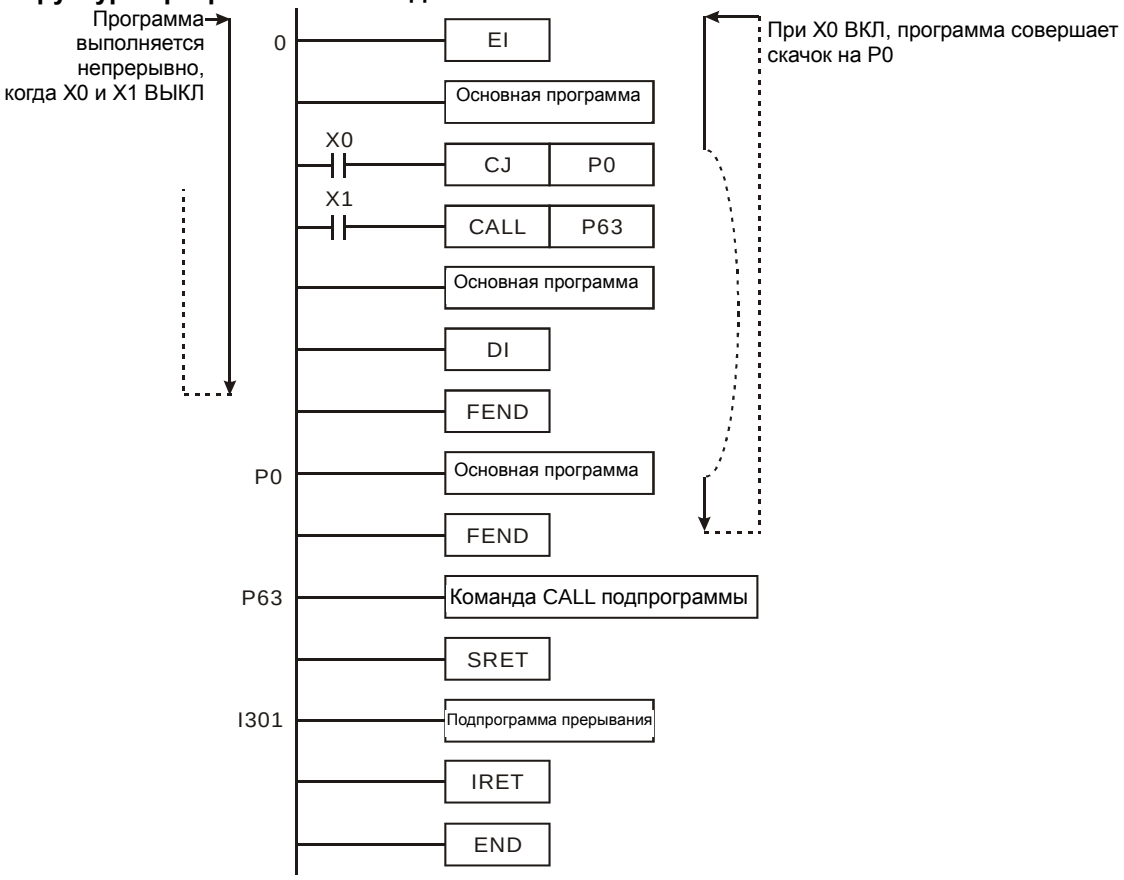
Операнд	Описание	Шаги программы
НЕТ	Контакт для условия выполнения команды не требуется.	FEND: 1 шаг

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

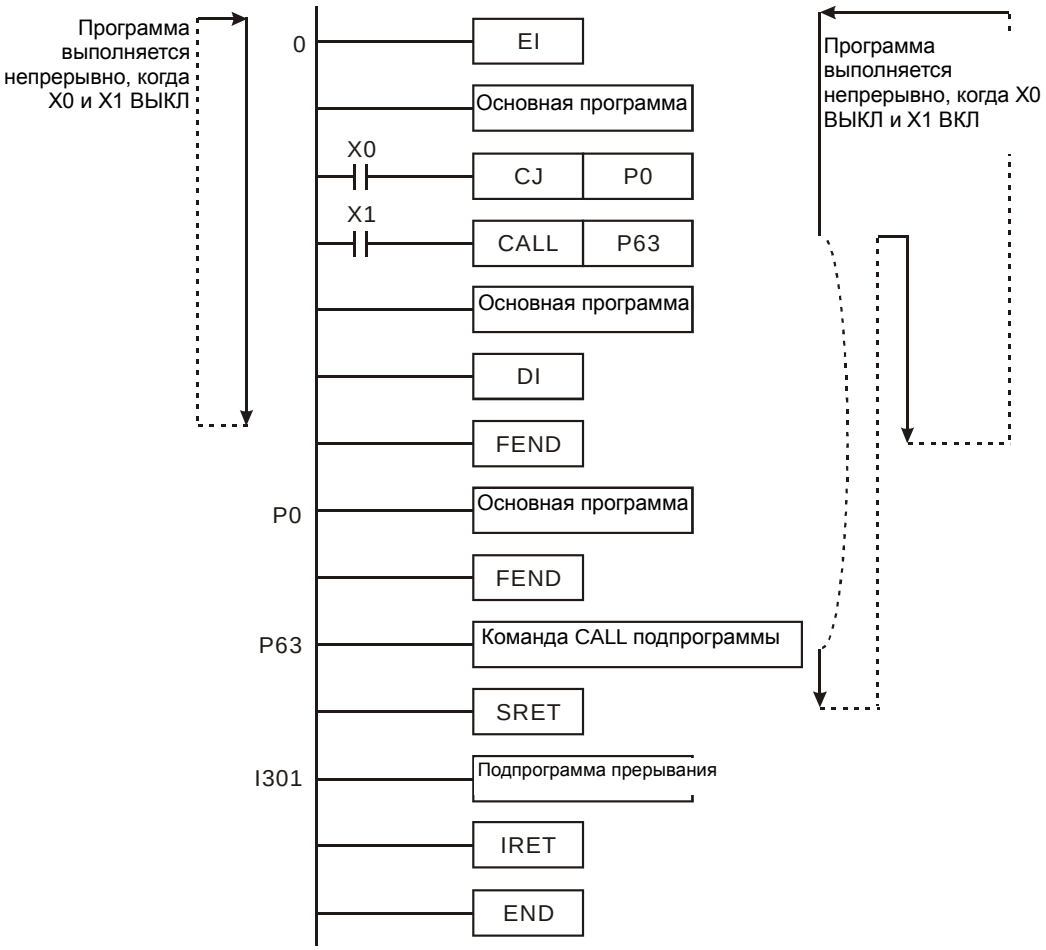
Описание:

1. Команда FEND применяется, когда в программе имеются обращения к подпрограммам (команды CALL и прерывания). В остальных случаях команда END должна завершать программу.
2. Функции команды FEND полностью аналогичны функциям команды END.
3. В основной программе подпрограммы должны размещаться после команды FEND, причём каждая подпрограмма должна завершаться командой SRET.
4. В основной программе подпрограммы прерываний должны размещаться после команды FEND, причём каждая подпрограммам должна завершаться командой IRET.
5. При использовании команд FEND команда END также необходима. Но она должна быть размещена после всех подпрограмм в конце всей программы.
6. В том случае, когда имеется несколько команд FEND подпрограммы и программы обработки прерываний должны размещаться между последней командой FEND и командой END.
7. При выполнении команды CALL появление команды FEND до команды SRET приведёт к ошибке.
8. При выполнении команды FOR появление команды FEND до команды NEXT приведёт к ошибке.

Структура программы с командой CJ .



Структура программы с командой CALL



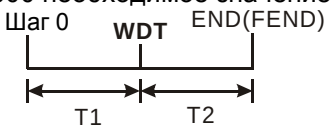
API	Команда		Функция	Контроллеры			
07	WDT	P	Сброс сторожевого таймера	ES2/EX2	SS2	SA2/SE	SX2

Операнд	Описание	Шаги программы
НЕТ		WDT, WDTP: 1 шаг

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Описание:

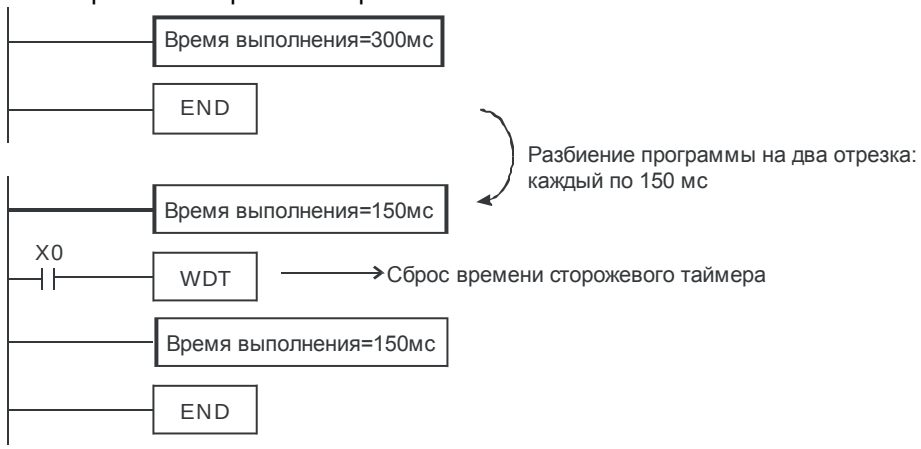
1. Команда WDT может использоваться для сброса сторожевого таймера Watch Dog Timer. Если время цикла превысит 200мс (от адреса 0 до команды END или FEND), то будет мигать светодиод ERROR LED. В этом случае пользователю необходимо выключить питание и повторно включить его для сброса ошибки. Контроллер определит статус RUN/STOP в соответствии с состоянием переключателя RUN/STOP. При его отсутствии контроллер перейдёт в состояние STOP автоматически.
2. Использование сброса сторожевого таймера WDT:
   a) При появлении ошибок в контроллере.
   b) Когда время скана превысит значение WDT в регистре D1000.
   c) Его задать можно двумя способами:используя команду WDT или записав в регистр D1000 необходимое значение (по умолчанию установлена величина 200 мс).



Замечание:

При использовании команды WDT происходит её выполнение в каждом скане в соответствии с условиями работы программы. Если необходимо обеспечить её выполнение в одном скане, то следует использовать импульсный вид команды WDT, то есть команду WDTP.

Пример программы:
При времени скана программы более 200 мс, например, 300 мс, пользователь может разделить программу на две части. Установите команду WDT в теле программы так, чтобы время сканирования первой и второй части менее 200 мс



API	Команда	Операнд	Функция	Контроллеры			
				ES2/EX2	SS2	SA2 /SE	SX2
08	FOR	(S)	Начало цикла FOR-NEXT				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	FOR: 3 шага

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнд:
S: Число повторений цикла.

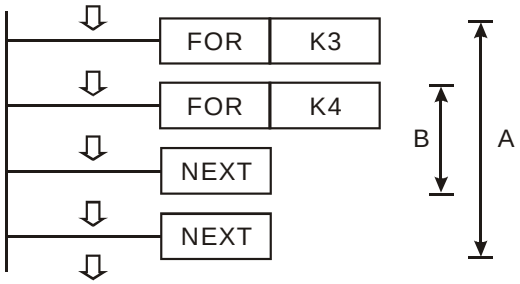
API	Команда	Функция	Контроллеры			
			ES2/EX2	SS2	SA2/ SE	SX2
09	NEXT	Конец цикла FOR-NEXT				

Операнд	Описание	Шаги программы
НЕТ	Контакт для условия выполнения команды не требуется.	NEXT: 1 шаг

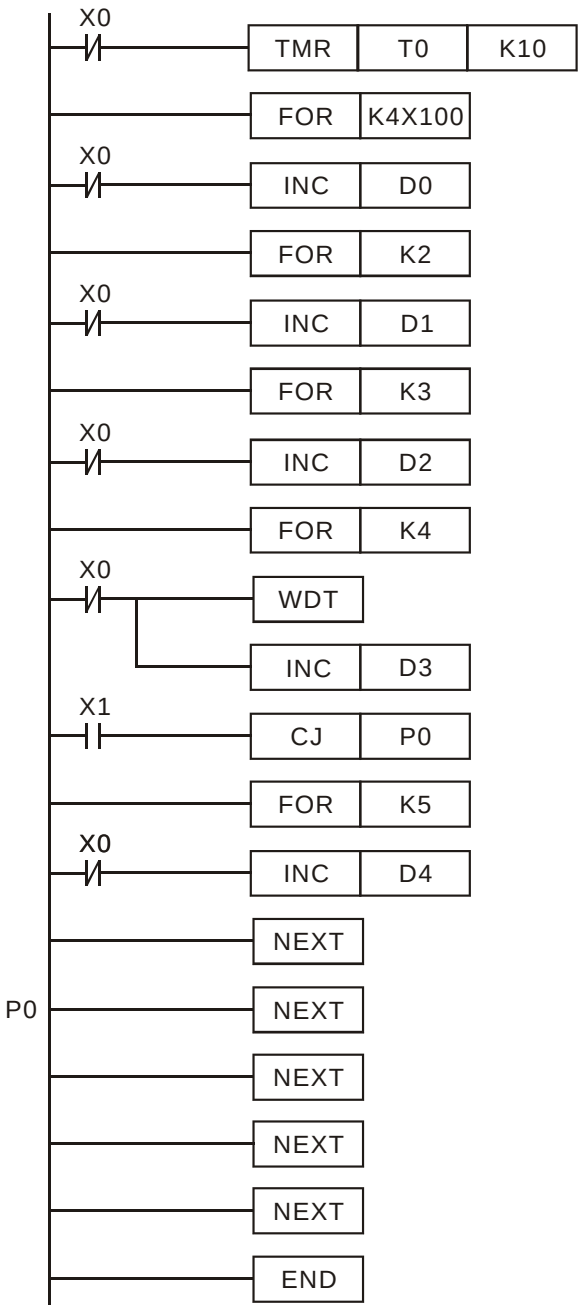
ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Описание:
1. Команды FOR/NEXT используются для программирования циклических повторений частей программы (цикл программы).
2. Часть программы между FOR- и NEXT-инструкциями повторяется "n" раз, после чего будет выполнен шаг программы после NEXT-инструкции.
3. Значение "n" может находиться внутри следующей области: "n": от +1 до +32 767. Если для "n" указано значение между 0 и -32 767, то цикл FOR-NEXT отрабатывается только один раз.
4. В следующих случаях будет выдана ошибка в работе программы:
   - NEXT-команда стоит перед FOR-командой.
   - NEXT-команда стоит после FEND-команды или END-командой.
   - Количество команд NEXT не соответствует количеству команд FOR.
   - Большое число повторений "n" может значительно увеличить время выполнения программы, и сторожевой таймер может вызвать ошибку. Используйте WDT-команду.

Пример программы 1:
В примере запрограммированы два входящие друг в друга FOR~ NEXT-цикла.
Отрезок программы A обрабатывается три раза (здесь K3 константа 3).
При каждом выполнении отрезка A отрезок программы B обрабатывается четыре раза (здесь K4 константа 4).
Поэтому отрезок B обрабатывается 3 x 4 = 12 раз.

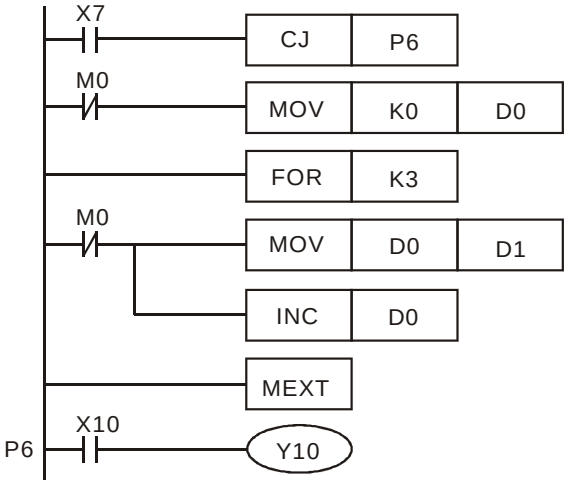


Пример программы 2:



Если вход X1 включен, то вложенный FOR-NEXT-цикл пропускается с помощью команды CJ.
Пользователь может задать, чтобы команда CJ пропускала указанные FOR-NEXT-циклы. Если X1=ВКЛ., команда CJ пропустит самый внутренний в структуре FOR-NEXT-цикл.

Пример программы 3:



При X7 = ВЫКЛ. контроллер будет выполнять часть программы FOR ~ NEXT. При X7 = ВКЛ., по команде CJ произойдёт переход к подпрограмме P6, выполнение части программы между FOR…NEXT не произойдёт.

API	Команда			Операнд	Функция	Контроллеры
10	D	CMP	P	S_1 S_2 D	Сравнение числовых данных	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	CMP, CMPP: 7 шагов DCMP, DCMPP: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

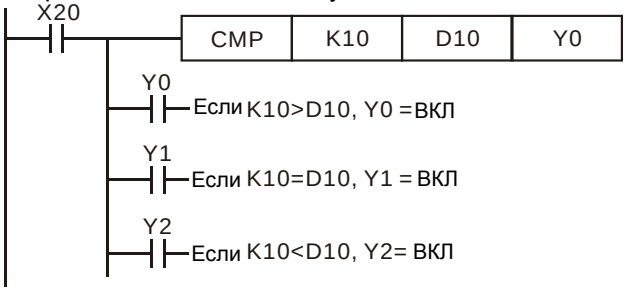
S_1: Сравниваемая величина 1 **S_2**: Сравниваемая величина 2 **D**: Результат сравнения

Описание:

1. Результат сравнения операндов **S_1** и **S_2** сохраняется в операнде **D** (двоичное число со знаком).
2. Если b15=1 в 16-битной команде или b31=1 в 32-битной команде, то результат сравнения будет иметь отрицательное значение
3. Операнд **D** занимает три последовательных регистра. **D**, **D** +1, **D** +2 где хранится результат сравнения **D** = ВКЛ. при **S_1** > **S_2**, **D** +1 = ВКЛ. при **S_1** = **S_2**, **D** +2 = ВКЛ. при **S_1** < **S_2**
4. Если операнды **S_1**, **S_2** используют индексный регистр F, то возможны только 16-битные команды.

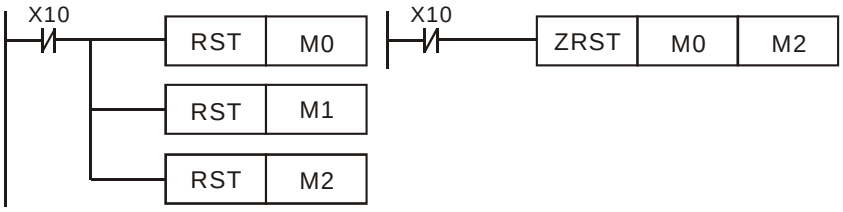
Пример программы:

1. Если **D** определён как Y0, то операнды Y0, Y1, Y2 будут показывать результат сравнения
2. При X20 = ВКЛ. команда CMP выполнится и один из операндов Y0, Y1, Y2 будет в состоянии ВКЛ. При X20 = ВЫКЛ. команда CMP не будет выполняться и состояние операндов Y0, Y1, Y2 не будет меняться.



3. Для сброса результата сравнения используются команды RST или ZRST .
   1) Y0: включен, если K10 > значения регистра D10
   2) Y1: включен, если K10 = значению регистра D10
   3) Y2: включен, если K10 < значения регистра D10
4. Y0, Y1, Y2 не изменяются, если входное условие X10 выключено.

5.



API	Команда			Операнд	Функция	Контроллеры			
11	D	ZCP	P	S_1 S_2 S D	Зонное сравнение числовых данных	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	ZCP, ZCPP: 9 шагов DZCP, DZCPP: 17 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
S					*	*	*	*	*	*	*	*	*	*	*	
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

S_1: Нижняя граница зоны сравнения **S2**: Верхняя граница зоны сравнения
S: Сравниваемое значение **D**: Результат сравнения

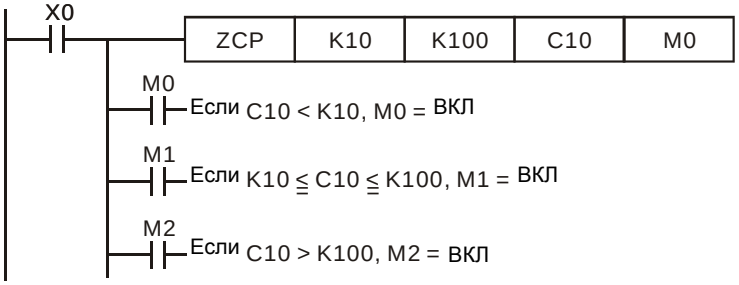
Описание:

1. Операнд S сравнивается операндом (нижний предел) S1 и с операндом S2 (верхний предел). В операнде D сохраняются результаты сравнения.
2. Если b15=1 в 16-битной команде или b31=1 в 32- битной команде, то результат сравнения будет иметь отрицательное значение.
3. Операнд **S_1** должен быть меньше **S_2**. При **S_1** > **S_2** команда принимает операнд **S_1** как первый t операнд для сравнения и производит сравнение аналогично команде CMP.
4. Если операнды **S_1**, **S_2** используют индексный регистр F, то возможны только 16-битные команды.
5. Операнд **D** занимает три последовательных регистра **D**, **D** +1, **D** +2, где хранится результат сравнения,

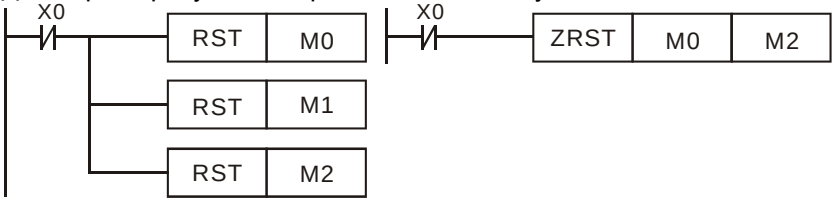
   D = ВКЛ. если **S_1** > **S**; **D** +1 = ВКЛ. если **S_1** ≦ **S** ≦ **S_2**; **D** +2 = ВКЛ. если **S_2** < **S**

Пример программы:

1. Если **D** определён как M0, то операнды M0, M1, M2 будут показывать результат сравнения
2. При X0 = ВКЛ. команда ZCP выполняется, и один из маркеров M0, M1, M2 будет включен. При X0 = ВЫКЛ. команда ZCP не выполняется, и состояние M0, M1, M2 не будет изменено.



3. Для сброса результата сравнения используются команды RST или ZRST.



API	Команда			Операнд	Функция	Контроллеры			
12	D	MOV	P	(S) (D)	Передача данных	ES2/EX2	SS2	SA2/ SE	SX2

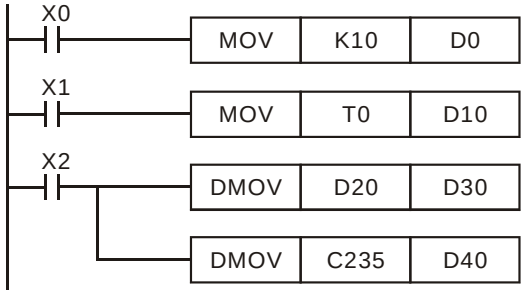
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	MOV, MOVP: 5 шагов DMOV, DMOVP: 9 шагов
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных **D**: приёмник данных

Описание:
1. При выполнении данной команды содержимое источника данных S передаётся непосредственно в приёмник данных **D**. В противном случае содержимое приёмника данных остаётся неизменным.

Пример программы:



При X0 = ВЫКЛ. содержимое регистра D0 остается неизменным.
При X0 = ВКЛ. число K10 передаётся в регистр D0.
При X1 = ВЫКЛ. содержимое регистра D10 остается неизменным.
При X1 = ВКЛ. содержимое таймера T0 передаётся в регистр D10.
Команда DMOV будет передавать 32-битное значение источника данных в приёмник данных.
При X2 = ВЫКЛ. содержимое регистров (D31, D30 и D41, D40) остается неизменным.
При X2 = ВКЛ. содержимое регистров (D21, D20) передаётся в регистры (D31, D30).
Содержимое счётчика C235 передаётся в регистры (D41, D40).

API	Команда			Операнд	Функция	Контроллеры			
13		SMOV	P	(S) (m_1) (m_2) (D) (n)	Передача данных со смещением	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	SMOV, SMOVP: 11 шагов
m_1					*	*										
m_2					*	*										
D								*	*	*	*	*	*	*	*	
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

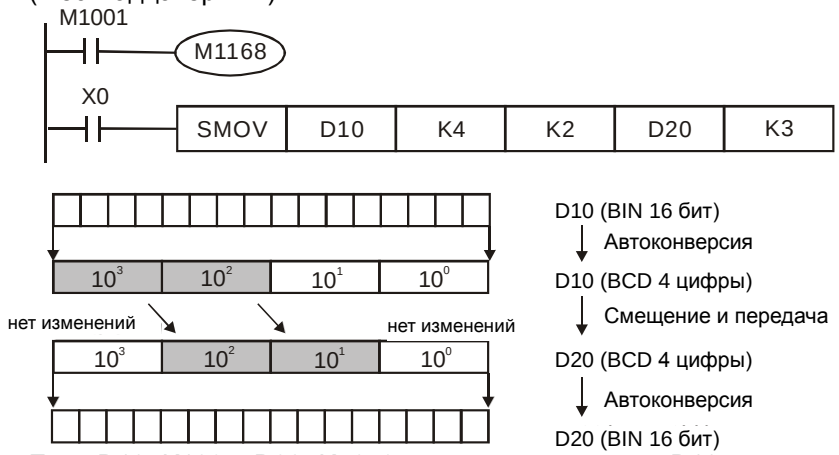
Операнды:
S: Источник данных **m1**: Первое место (в слове), которое должно быть смещено
m2: Число мест, которые должны быть смещены **D**: приёмник данных **n**: Первое место в приёмнике данных для смещённых данных.

Описание:
1. Эта команда позволяет разместить со смещением или скомбинировать данные.
   При выполнении команды SMOV цифры, содержащиеся в слове, начиная с цифры m1 (от старшей цифры к младшей, число цифр m2) источника данных S, будут переданы, начиная с цифры n (от старшей к младшей) в приёмнике D.
2. M1168 используется для настройки режима работы команды SMOV. Когда M1168=1, данные регистров D10, D20 не конвертируются в BCD-формат, а передаются как 4-х разрядное BIN-число. Когда M1168 = ВЫКЛ., команда выполняется в BCD формате

Замечания:
Диапазон $\mathbf{m_1}$: 1 – 4; $\mathbf{m_2}$: 1 – $\mathbf{m_1}$; **n**: $\mathbf{m_2}$ – 4

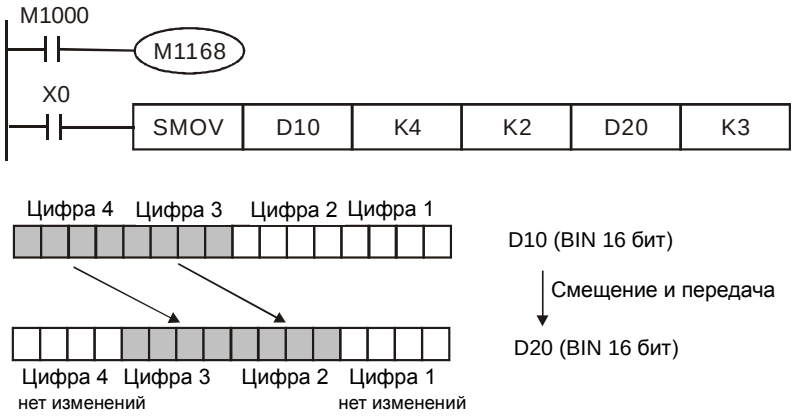
Пример программы 1:

1. Когда M1168=ВЫКЛ. (в режиме BCD (двоично-десятичный)) и X0=ВКЛ., 4-й (тысячи) и 3-й (сотни) цифры десятичного значения в D10 передвигаются на 3-й (сотни) и 2-й (десятки) цифры десятичного значения D20. После выполнения команды 10^3 и 10^0 у D20 останутся неизменными.
2. Если значение BCD выходит за пределы диапазона 0-9999, ПЛК определяет ошибку и не будет выполнять команды. M1067, M1068=ВКЛ. и D1067 сохраняет код ошибки OE18 (шестнадцатеричн.)



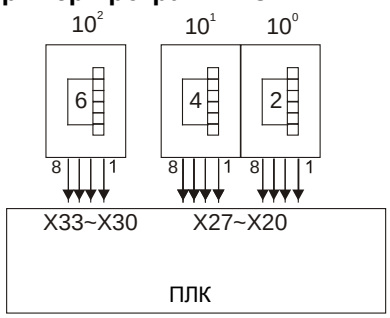
Если D10=K1234, D20=K5678 перед выполнением, D10 останется неизменным и D20=K5128 после выполнения.

Пример программы 2:

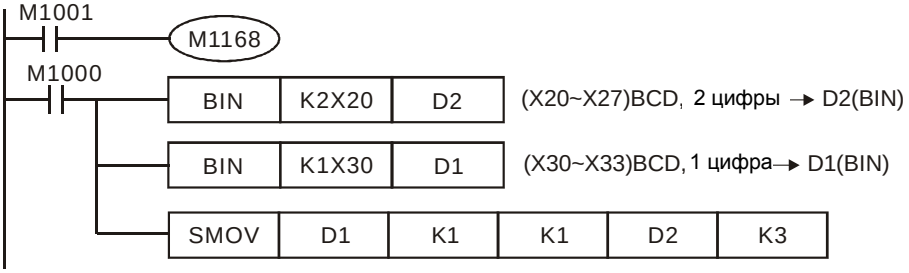


Когда M1168=ВКЛ. (в режиме BIN (бинарный)) и команда SMOV выполняется, D10 и D20 не будут конвертированы в формат BCD (двоично-десятичный), а будут перенесены в формате BIN (4 - разрядное).
Если D10=K1234, D20=K5678 перед выполнением, D10 останется неизменным и D20=K5128 после выполнения.

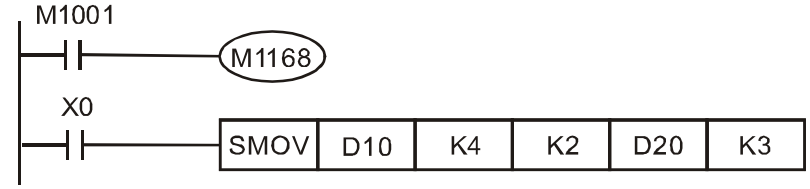
Пример программы 3:

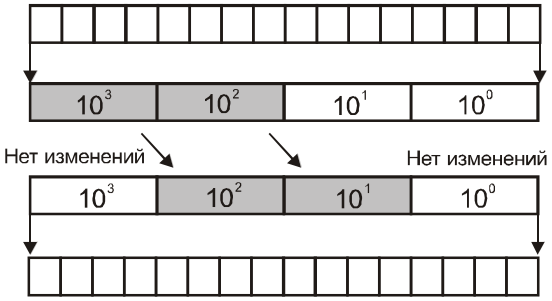


1. Эта команда может использоваться для задания комбинаций DIP-переключателей, подключенных к входам терминалов без постоянного номера.
2. Перемещение на 2 позиции правого DIP-переключателя (X27-X20) на 2-е значение D2 и на 1 позицию DIP-переключателя (X33-X30) на 1-е значение D1.
3. Используйте команду SMOV для перемещения 1-го значения D1 до 3-го значения D2 и объединения значений двух переключателей в единое целое.



Пример команды SMOV с M1168=0

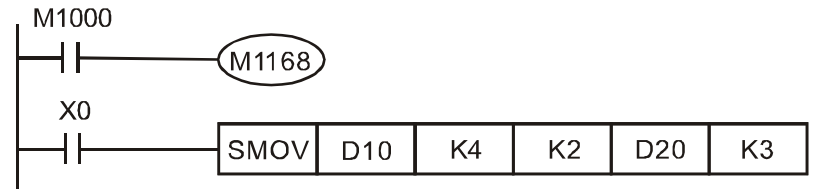


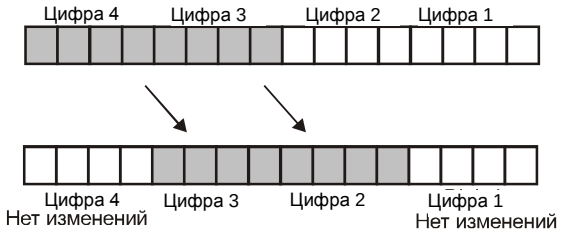


D10 (BIN 16 бит)
↓ Автоконверсия
D10 (BCD 4 цифры)
↓ Смещение и передача
D20 (BCD 4 цифры)
↓ Автоконверсия
D20 (BIN 16 бит)

Если входное условие X0 включено, то часть содержимого регистра D10=H1234 будет передано части содержимого регистра D20:
До выполнения:
D10=H1234,
D20=H5678
После выполнения:
D10=H1234,
D20=H5128

Пример команды SMOV с M1168=1

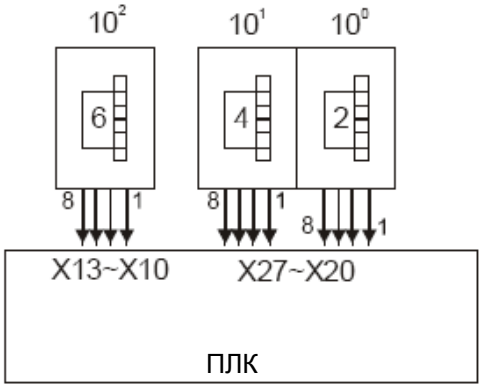




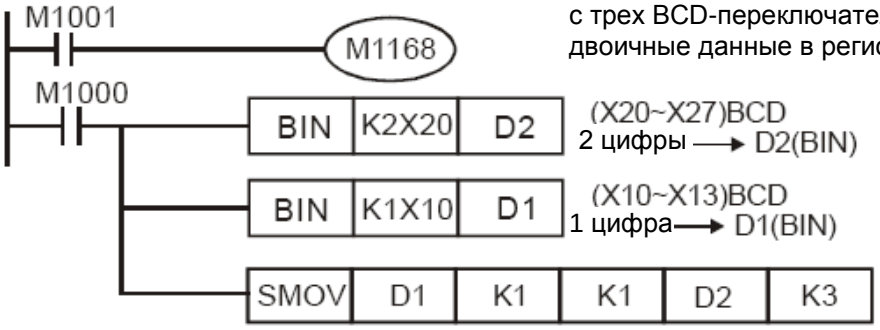
D10(BIN 16 бит)
↓ Смещение и передача
D20(BIN 16 бит)

Когда M1168=1, данные регистров D10, D20 не конвертируются в BCD-формат, а передаются как 4-х разрядное BIN-число.

Пример программирования для двоичных данных



1. Входные данные поставляются от трех BCD-переключателей, распределенных по соответствующим входам ПЛК.
2. Вначале конвертируются BCD-данные входов X20...X27 (K2X20 - две тетрады (8 бит) K2, начиная с бита X20) в двоичные данные и заносятся в регистр данных D2.
3. Аналогично преобразовываются в двоичные данные BCD-данные входов X10...X13 (K1X10).
4. BCD-значение регистра данных D1 (первая тетрада m1 = K1 и одна перемещаемая тетрада m2 = K1) записываются на третью позицию регистра цели D2. В заключение BCD-данные снова преобразовываются в двоичные данные.
5. В примере числовые входные данные снимаются с трех BCD-переключателей и запоминаются как двоичные данные в регистре данных D2.



API	Команда			Операнд	Функция	Контроллеры			
14	D	CML	P	(S) (D)	Передача данных с их инвертированием	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	CML, CMLP: 5 шагов DCML, DCMLP: 9 шагов
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Приёмник данных

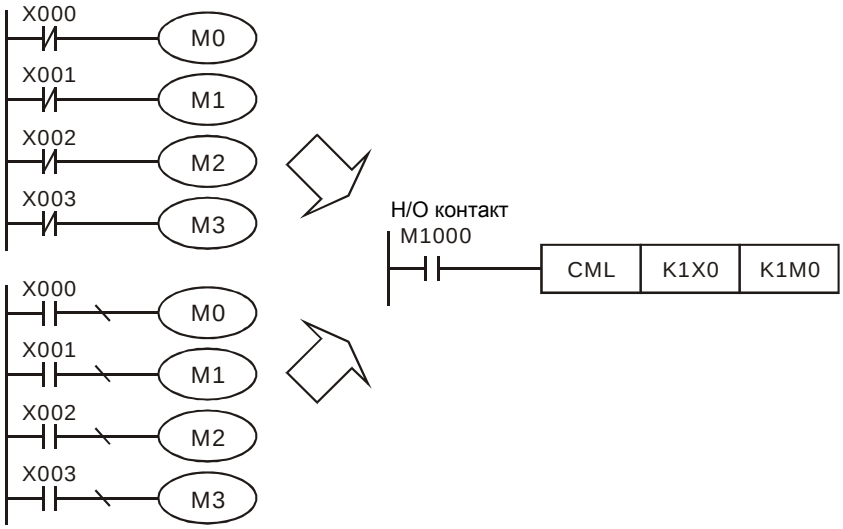
Описание:
1. Двоичное значение числа в (S) преобразовывается в свое 1-ое дополнение и записывается в приемник данных (D).
2. Если для операндов **S** и **D** применяется индексный регистр **F**, то возможно только 16 битное исполнение.

Пример программы 1:
Когда X20 = ВКЛ., b0 ~ b3 в регистр D1 будут инвертированы и переданы в Y0 ~ Y3



Пример программы 2:

Ниже показан пример передачи обратного состояния входов X внутренним реле M (программы слева идентичны инструкции справа):



API	Команда			Операнд	Функция	Контроллеры			
15		BMOV	P	(S) (D) (n)	Передача блока данных	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			BMOV, BMOVP: 7 шагов
D								*	*	*	*	*	*			
n					*	*					*	*	*			

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

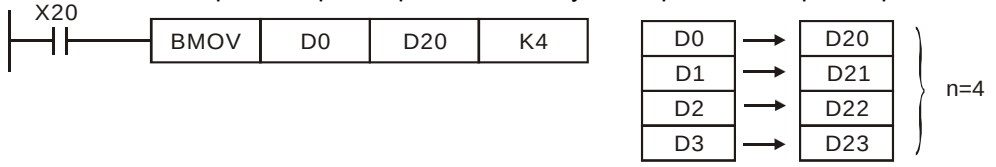
Операнды:
S: Начальное слово источника данных **D**: Начальное слово приёмника данных
n: Длина блока данных

Описание:

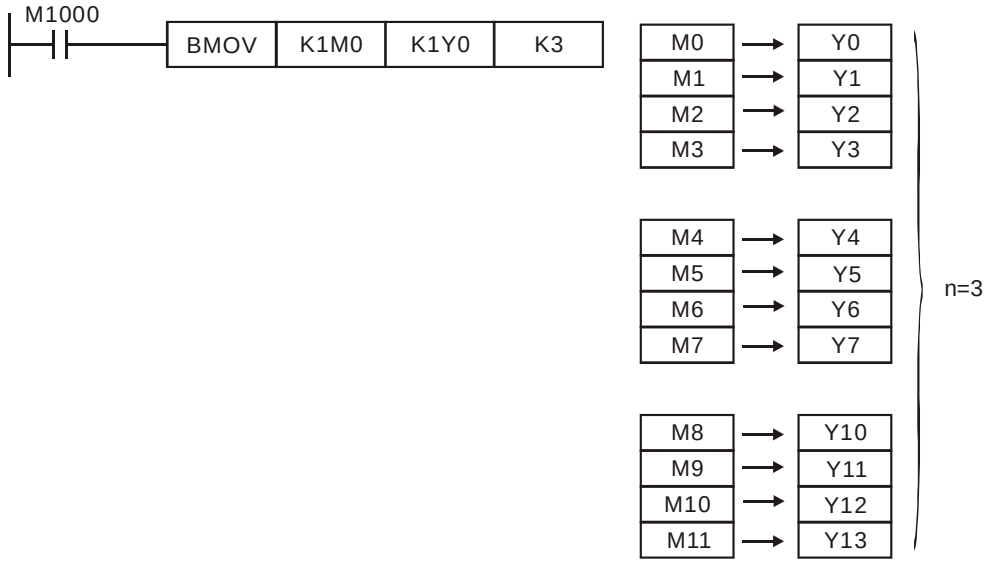
Переносится (передается) предварительно заданное количество словных операндов. Для передачи предварительно задаются стартовый адрес (S), адрес приёмника данных (D) и число переносимых слов (n).

1. Если величина пакета данных превышает величину областей цели или источника, то передаются только слова, которые могут поместиться в области
2. Диапазон **n**: 1 ~ 512.

Пример программы 1:
Когда X20 = ВКЛ. содержимое регистров D0 ~ D3 будет передано в 4 регистра D20 ~ D23



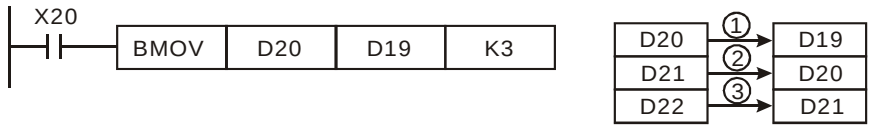
Пример программы 2:
Предположим, биты KNX, KnY, KnM и KnS предназначены для перемещения, значения S и D должны быть одинаковыми.



Пример программы 3:

Если адреса источников S и приемников данных D пересекаются, то порядок копирования будет следующий:

Если S > D: порядок копирования 1 → 2 → 3; Если S < D: порядок копирования 3 → 2 → 3



API	Команда			Операнд	Функция	Контроллеры			
16	D	FMOV	P	(S) (D) (n)	Передача данных в несколько адресов	ES2/EX2	SS2	SA2/SE	SX2

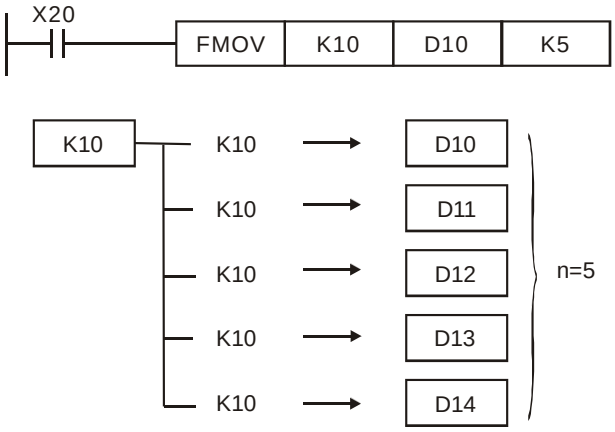
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	FMOV, FMOVP: 7 шагов DFMOV, DFMOVP: 13 шагов
D								*	*	*	*	*	*			
n					*	*										

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Приёмник данных **n**: число передаваемых слов

Описание:
1. Переносится (передается) предварительно заданное количество словных операндов. Для передачи предварительно задаются стартовый адрес (S), адрес первого регистра приёмника данных (D) и число переносимых слов (n)
2. Если величина блока данных превышает величину областей цели или источника, то передаются только слова, которые могут поместиться в области
3. Если для операндов **S** и **D** применяется индексный регистр **F,** то возможно только 16 битное исполнение команды.
4. Диапазон **n**: 1~ 512

Пример программы:
Когда X20 = ВКЛ., число K10 будет записано в 5 последовательных регистров, начиная с D10.



API	Команда			Операнд	Функция	Контроллеры			
17	D	XCH	P	D_1 D_2	Обмен данными	ES2/EX2	SS2	SA2/SE	SX2

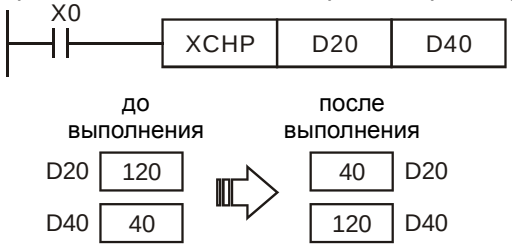
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D_1								*	*	*	*	*	*	*	*	XCH, XCHP: 5 шагов DXCH, DXCHP: 9 шагов
D_2								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D1: Данные для обмена 1 **D2**: Данные для обмена 2

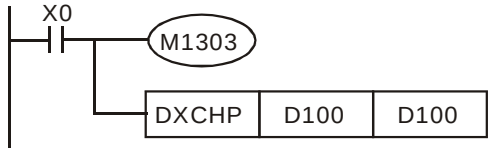
Описание:
1. Операнды, заданные как D1 и D2 будет обмениваться содержимым.
2. Предпочтительно применять импульсные команды (XCHP).
3. Если для операндов **D1** и **D2** применяется индексный регистр **F**, то возможно только 16 битное исполнение команды.

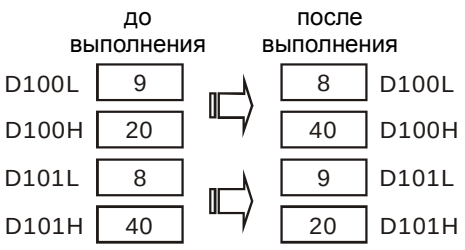
Пример программы:
Пример X0=ВЫКЛ.→ВКЛ., содержимое регистров D20 и D40 меняется местами.



Замечание:
1. При 16 битном исполнении команды, когда операнды D1 и D2 одинаковы и M1303 = ВКЛ., старшие и младшие 8 бит меняются местами.
2. При 32- битном исполнении команды, когда операнды D1 и D2 одинаковы и M1303 = ВКЛ., старшие и младшие 16 бит меняются местами

3. При X0 = ВКЛ. и M1303 = ВКЛ. 16-битные блоки в D100 и 16-битные блоки D101 поменяются местами.





API	Команда			Операнд	Функция	Контроллеры			
18	D	BCD	P	(S) (D)	BCD-конвертирование	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	BCD, BCDP: 5 шагов DBCD, DBCDP: 9 шагов
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных **D**: Результат преобразования

Описание:
1. Двоичные данные источника (S) конвертируются в BCD-данные и передаются в приемник данных (D).
2. Результат BCD-конвертирования должен находиться внутри допустимой области: 16-битовая инструкция: от 0 до +9 999 и 32-битовая инструкция: от 0 до +99 999 999

Источник ошибки:
1. Если результат BCD-конвертирования находится вне допустимой области, то появляется ошибка обработки программы (M1067, M1068 = 1) с кодом в D1068 = 0E18
2. Если для операндов **S** и **D** применяется индексный регистр **F**, то возможно только 16 битное исполнение команды.
3. Флаги: M1067 (Ошибка выполнения программы), M1068 (Фиксация ошибки выполнения), D1067 (код ошибки)

Пример программы:
1. Когда X0 = ВКЛ., двоичное значение D10 будет преобразовано в BCD формат, первая цифра результата будет сохранена в K1Y0 (Y0 ~ Y3, 4 разрядный регистр).



2. Когда D10=001E (шестнадцатеричное) = 0030 (десятичное), то будет выдан результат Y0~Y3 = 0000(двоичное).

API	Команда			Операнд	Функция	Контроллеры			
19	D	BIN	P	(S) (D)	BIN-конвертирование	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*	*	*	BIN, BINP: 5 шагов DBIN, DBINP: 9 шагов
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16- бит				32- бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных **D**: Результат преобразования

Описание:
1. Содержимое операнда S (BCD формат) преобразуются в двоичный формат D.
2. Допустимое значение операнда S: BCD формат (0 ... 9,999), DBCD (0 ... 99,999,999)

3. Если содержимое операнда **S** не является BCD, то появляется ошибка, Флаги ошибки M1067 и M1068 = ВКЛ. и содержимое D1067 = H0E18.
4. Флаги: M1067 (Ошибка выполнения программы), M1068 (Фиксация ошибки выполнения), D1067 (код ошибки)
5. Если для операндов **S** и **D** применяется индексный регистр **F**, то возможно только 16 битное исполнение команды.
6. Флаги: M1067 (Ошибка выполнения программы), M1068 (Фиксация ошибки выполнения), D1067 (код ошибки)

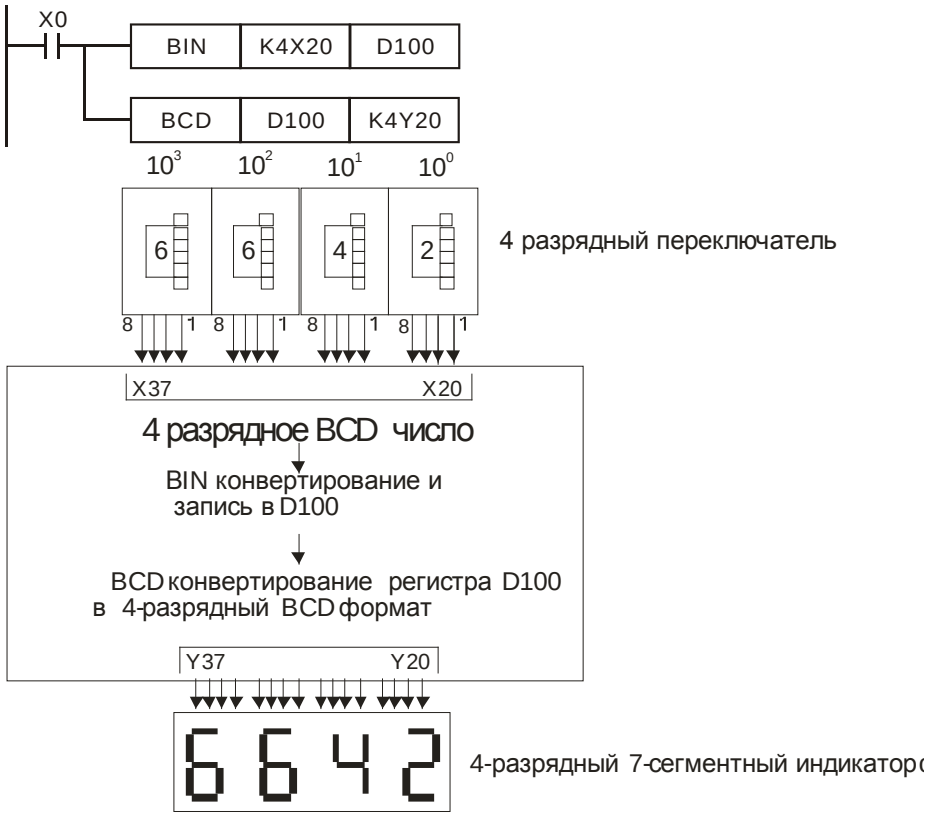
Пример программы:



Когда X0 = ВКЛ., BCD формат K1M0 будет преобразован в двоичный вид и результат будет сохранён в регистре D10.

Замечания:

1. Когда контроллеру необходимо считать состояние DIP переключателей в BCD формате, то команда BIN конвертирования сначала преобразует считанные данные в двоичный формат и далее сохраняет их в контроллере.
2. С другой стороны, когда контроллеру необходимо вывести на 7-сегментную индикацию данные, то используя команду BCD-конвертирования, требуется преобразовать данные в BCD формат и выдать их на индикацию.
3. При X0 = ВКЛ. BCD формат K4X20 превратится в двоичное число и сохранится в D100. Далее двоичное число D100 будет преобразовано в BCD формат и выдано на выходы K4Y20.



API	Команда			Операнд	Функция	Контроллеры			
20	D	ADD	P	S_1 S_2 D	Сложение	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	ADD, ADDP: 7 шагов DADD, DADDP: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Слагаемое **S_2**: Слагаемое **D**: Сумма

Описание:

a) Эта команда суммирует данные в двоичном формате из источников **S1** и **S2** и сохраняет результат в адресе суммы **D**.

b) В старшем бите (MSB) запоминается знак числа суммирования (0: знак положительного числа, 1: знак отрицательного числа). Все расчеты алгебраически обрабатываются, например: 3 + (-9) = -6.

c) Если S1, S2 и D используются с индексом F, то возможно только 16-битное выполнение команды.

d) Флаги: M1020 (флаг нуля), M1021 (флаг заимствования), M1022 (флаг переноса).

Пример программы 1:
При выполнении16-битной команды:
Если X0 = ВКЛ., содержимое D0 суммируется с D10 и результат сохраняется в D20.



Пример программы 2:
При выполнении 32-битной команды:
Если X0 = ВКЛ., содержимое (D31, D30) суммируется с содержимым (D41, D40) и результат сохраняется в (D51, D50). D30, D40 и D50 являются операндами слова младших 16 бит, а D31, D41 и D51 - операндами слова старших 16 бит.



(D31, D30) + (D41, D40) = (D51, D50)

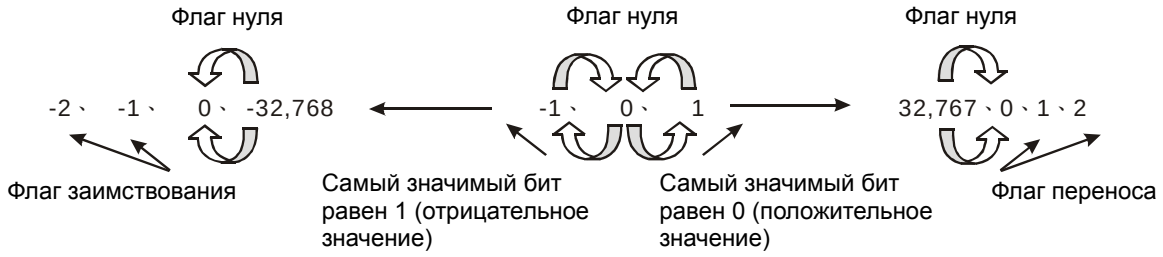
Применение флагов:
16-битная команда:

1. Если результат равен "0", то включается флаг нуля M1020.
2. Если результат меньше "-32 768", то включается флаг заимствования M1021.
3. Если результат больше "32 767", то включается флаг переноса M1022.

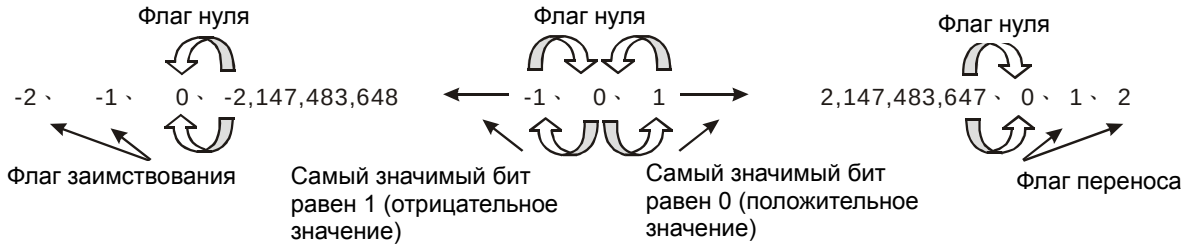
32-битная команда:

1. Если результат равен "0", то включается флаг нуля M1020.
2. Если результат меньше "-2 147 483 648", то включается флаг заимствования M1021.
3. Если результат больше "2 147 483 647", то включается флаг переноса M1022.

16-битная команда:



32-битная команда:



API	Команда			Операнд	Функция	Контроллеры			
21	D	SUB	P	S_1 S_2 D	Вычитание	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

Шаги программы: SUB, SUBP: 7 шагов; DSUB, DSUBP: 13 шагов

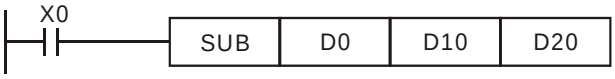
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходное значение **S_2**: Вычитаемое **D**: Результат

Описание:
1. Эта команда вычитает данные S2 в двоичном формате из источника S1 и сохраняет результат в адресе D.
2. В старшем бите (MSB) запоминается знак операции:1 - отрицательный, 0 - положительный. Все расчеты алгебраически обрабатываются.
3. Если S1, S2 и D используются с индексом F, то возможно только 16-битное выполнение команды.
4. Флаги: M1020 (флаг нуля), M1021 (флаг заимствования), M1022 (флаг переноса), аналогично флагам операции сложения (команда ADD).

Пример программы 1:
При выполнении 16-битной команды:

Если X0 = ВКЛ., из содержимого D0 вычитается содержимое D10 и результат сохраняется в D20.



Пример программы 2:
При выполнении 32-битной команды:

Если X0 = ВКЛ., из содержимого (D31, D30) вычитается содержимое (D41, D40) и результат сохраняется в (D51, D50). D30, D40 и D50 являются операндами слова младших 16 бит, а D31, D41 и D51 - операндами слова старших 16 бит.



(D31, D30) – (D41, D40) = (D51, D50)

API	Команда			Операнды	Функция	Контроллеры			
22	D	MUL	P	S_1 S_2 D	Умножение	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1					*	*	*	*	*	*	*	*	*	*	
S_2					*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	

Шаги программы: MUL, MULP: 7 шагов; DMUL, DMULP: 13 шагов

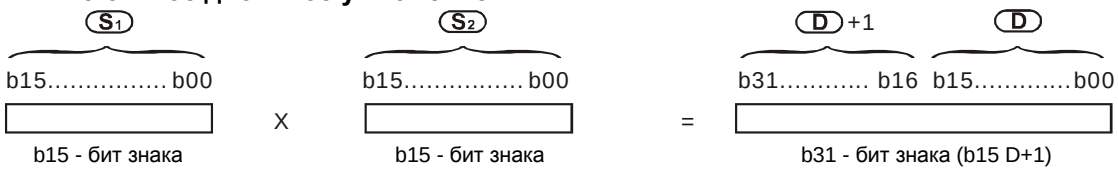
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Множимое **S_2**: Множитель **D**: Результат

Описание:
1. Эта команда перемножает данные S1 и S2 в двоичном формате и сохраняет результат в D. Обратите внимание на положительный/отрицательный знак при выполнении 16- и 32-битных команд.
2. MSB = 0, положительный знак; MSB = 1, отрицательный знак.

3. Если операнды S1, S2 используются с индексом F, то возможно выполнение только 16-битных команд.
4. Если операнд D используется с индексом E, то возможно выполнение только 16-битных команд.

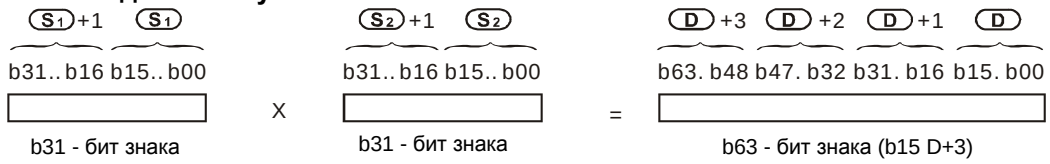
16-битное двоичное умножение



b15=0, S1 – положительное значение
b15=1, S1 – отрицательное значение
b15=0, S2 – положительное значение
b15=1, S2 – отрицательное значение
b31=0, D(D+1) – положительное значение
b31=1, D(D+1) – отрицательное значение

Если результат D определяется в битном формате, K1-K4 назначаются для хранения 16-битного результата. Пользователь может использовать два последовательных 16-битных регистра для хранения 32-битного результата.

32-битное двоичное умножение



b31=0, S1(S1+1) – положительное значение
b31=1, S1(S1+1) – отрицательное значение
b31=0, S2(S2+1) – положительное значение
b31=1, S2(S2+1) – отрицательное значение
b63=0, D…(D+3) – положительное значение
b63=1, D…(D+3) – отрицательное значение

Если результат D определяется в словном формате, K1-K8 назначаются для хранения 32-битного результата. Пользователь может использовать два последовательных 32-битных регистра для хранения 64-битного результата.

Пример программы:

16-битный D0 умножаем на 16-битный D10, получаем 32-битный результат. Старшие 16 бит записываются в D21, младшие в 20. ВКЛ./ВЫКЛ. MSB показывает соответственно положительный/отрицательный знак результата.



(D0) × (D10) = (D21, D20)
16-бит × 16-бит = 32-бит

API	Команда			Операнды	Функция	Контроллеры			
23	D	DIV	P	S1 S2 D	Деление	ES2/EX2	SS2	SA2/ SE	SX2

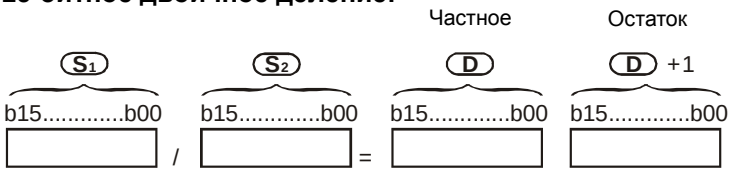
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DIV, DIVP: 7 шагов DDIV, DDIVP: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S_1: Исходное значение **S_2**: Делитель **D**: Результат

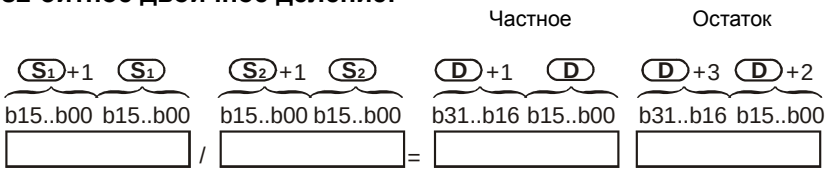
Описание:
1. Эта команда делит данные S1 на данные S2 в двоичном формате и сохраняет результат в D. Обратите внимание на положительный/отрицательный знак S1, S2 и D при выполнении 16- и 32-битных команд.
2. Команда не будет выполнена, если делитель равен 0. Флаги M1067 и M1068 будут включены, а в D1067 будет записан код ошибки 0E19 (в шестнадцатеричном коде).
3. Если операнды S1, S2 используются с индексом F, то возможно выполнение только 16-битных команд.
4. Если операнд D используется с индексом E, то возможно выполнение только 16-битных команд.

16-битное двоичное деление:



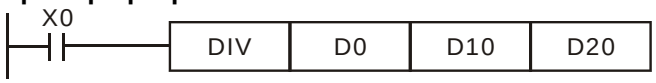
Если результат D определяется в битном формате, K1-K4 назначаются для хранения 16-битного результата. Пользователь может использовать два последовательных 16-битных регистра для хранения 32-битного результата.

32-битное двоичное деление:



Если результат D определяется в битном формате, K1-K4 назначаются для хранения 32-битного результата. Пользователь может использовать два последовательных 16-битных регистра для хранения 32-битного результата.

Пример программы:



Когда X0 = ВКЛ., D0 будет разделен на D10 и результат будет сохранен в D20, остаток деления будет сохранен в D21. Статус ВКЛ./ВЫКЛ. в MSB указывает соответственно на положительный/отрицательный знак результата.

API	Команда			Операнды	Функция	Контроллеры			
24	D	INC	P	D	Приращение на 1	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	INC, INCP: 3 шагов DINC, DINCP: 5 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D: Результат

Описание:
1. Если команда не используется в импульсном режиме, содержимое D будет увеличиваться на "1" в каждом цикле программы. В 16-битной команде при достижении значения 32767 и добавлении "1" будет записываться значение -32768, а в 32-битной команде при достижении значения 2147483647 и добавлении "1" будет записано значение - 2147483648.
2. Эта команда обычно используется в импульсном режиме выполнения (INCP, DINCP).
3. Если операнд D используется с индексом E, то возможно выполнение только 16-битных команд.
4. Результат операции не влияет на значение флагов M1020-M1022.



Пример программы:
Когда X0 срабатывает, содержимое D0 увеличивается на "1".

API	Команда			Операнды	Функция	Контроллеры			
25	D	DEC	P	D	Уменьшение на 1	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	DEC, DECP: 3 шагов DDEC, DDECP: 5 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D: Результат

Описание:
1. Если команда не используется в импульсном режиме, содержимое D будет уменьшаться на "1" в каждом цикле выполнения программы. При выполнении 16-битной команды при достижении результата -32768 и уменьшении на "1" будет записано +32767. При выполнении 32-битной команды при достижении результата - 2147483648 и уменьшении на "1" будет записано +2147483647.
2. Эта команда обычно используется в импульсном режиме выполнения (DECP, DDECP).
3. Если операнд D используется с индексом F, возможно выполнение только 16-битных команд. Результат операции не влияет на значение флагов M1020-M1022.

Пример программы:

Когда X0 срабатывает, содержимое D0 уменьшается на "1".



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
26	WAND	P	(S1) (S2) (D)	Логическое И				

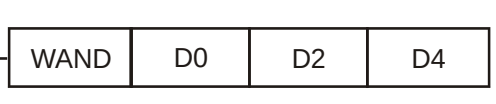
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WAND, WANDP: 7 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

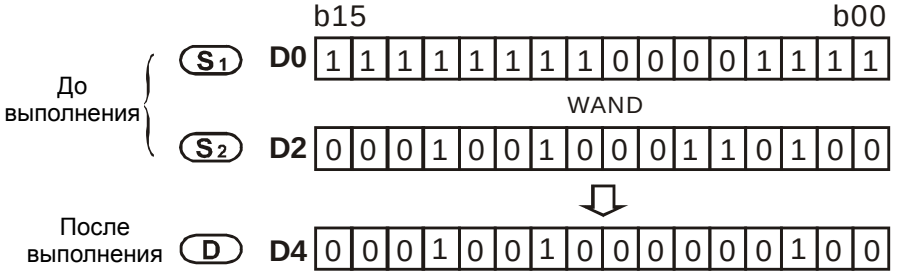
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходные данные 1 **S_2**: Исходные данные 2 **D**: Результат операции

Описание:
1. Эта команда производит логическую операцию И с операндами S1 и S2 в 16-битном режиме, после чего сохраняет результат в D.
2. Для выполнения 32-битной операции используйте команду DAND.

Пример программы:
Когда X0 включен, 16-битные источники D0 и D2 анализируются и результат логической операции сохраняется в D4.





API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
26	DAND	P	(S1) (S2) (D)	Логическое И (32-бит)				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DAND, DANDP: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

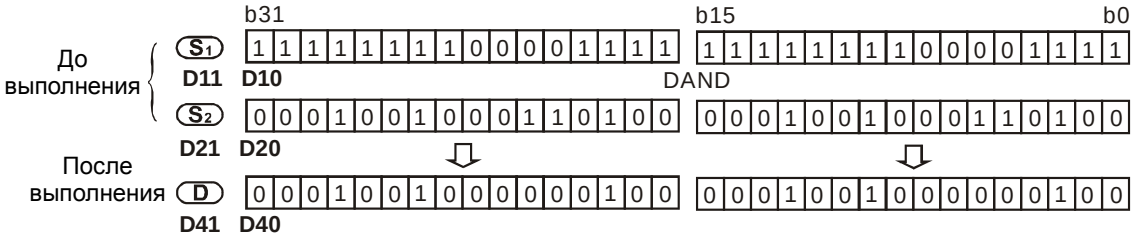
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходные данные 1 **S_2**: Исходные данные 2 **D**: Результат операции

Описание:
1. Эта команда производит логическую операцию И с операндами S1 и S2 в 32-битном режиме, после чего сохраняет результат в D.
2. Если операнды S1, S2 используются с индексом F, то возможно выполнение только 16-битных команд.

Пример программы:
Когда X1 включен, 32-битные источники (D11, D10) и (D21,D20) анализируются и результат логической операции сохраняется в (D41, D40).





API	Команда			Операнды	Функция	Контроллеры			
27		WOR	P	S_1 S_2 D	Логическое ИЛИ	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	WOR, WORP: 7 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

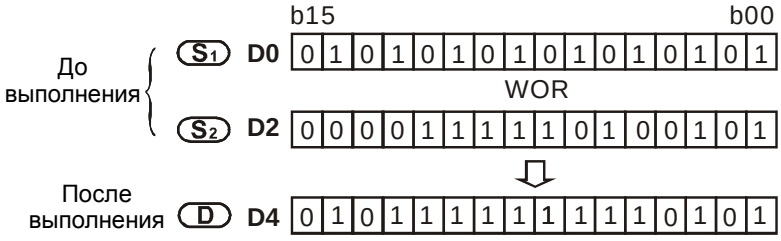
PULSE				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S_1: Исходные данные 1 **S_2**: Исходные данные 2 **D**: Результат операции

Описание:
1. Эта команда производит логическую операцию ИЛИ с операндами S1 и S2 в 16-битовом режиме, после чего сохраняет результат в D.
2. Для выполнения 32-битной операции используйте команду DOR.

Пример программы:
Когда X0 включен, 16-битные источники D0 и D2 анализируются и результат логической операции сохраняется в D4.





API	Команда		Операнды	Функция	Контроллеры			
27	DOR	P	(S1) (S2) (D)	Логическое ИЛИ (32-бит)	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DOR, DORP: 13 шагов
S_1					*	*	*	*	*	*	*	*	*	*		
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

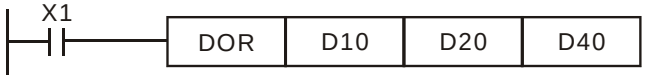
S_1: Исходные данные 1 **S_2**: Исходные данные 2 **D**: Результат операции

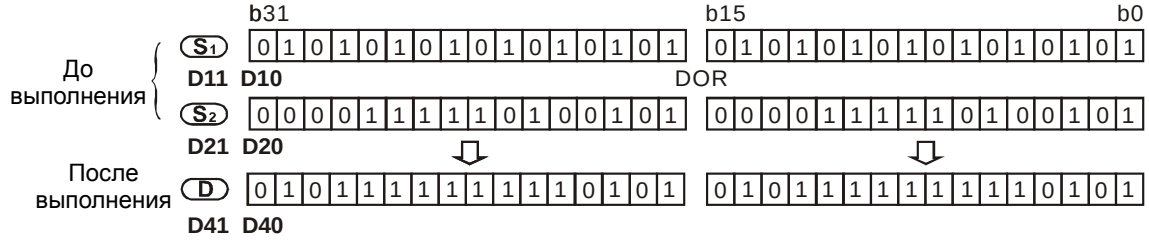
Описание:

1. Эта команда производит логическую операцию ИЛИ с операндами S1 и S2 в 32-битовом режиме, после чего сохраняет результат в D.
2. Если операнды S1, S2, D используются с индексом F, то возможно выполнение только 16-битных команд.

Пример программы:

Когда X1 включен, 32-битные источники (D11, D10) и (D21,D20) анализируются и результат логической операции сохраняется в (D41, D40)





API	Команда		Операнды	Функция	Контроллеры			
28	WXOR	P	(S1) (S2) (D)	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WXOR, WXORP: 7 шагов
S_1					*	*	*	*	*	*	*	*	*	*	*	
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

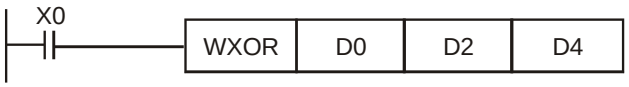
S_1: Исходные данные 1 **S_2**: Исходные данные 2 **D**: Результат операции

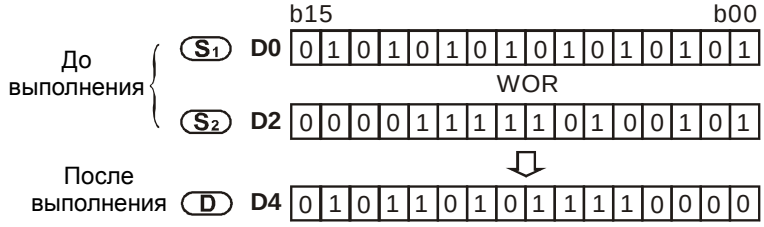
Описание:

Производится логическая операция ИСКЛЮЧАЮЩЕЕ ИЛИ с операндами S1 и S2 в 16-битовом режиме (команда DXOR для 32-битной операции), результат сохраняется в D.

Пример программы:

Когда X0 включен, 16-битные источники D0 и D2 анализируются и результат логической операции сохраняется в D4.





API	Команда			Операнды	Функция	Контроллеры			
28		DXOR	P	(S1) (S2) (D)	Логическое ИСКЛЮЧАЮЩЕЕ ИЛИ (32-бит)	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DXOR, DXORP: 13 шагов
S_1					*	*	*	*	*	*	*	*	*	*		
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

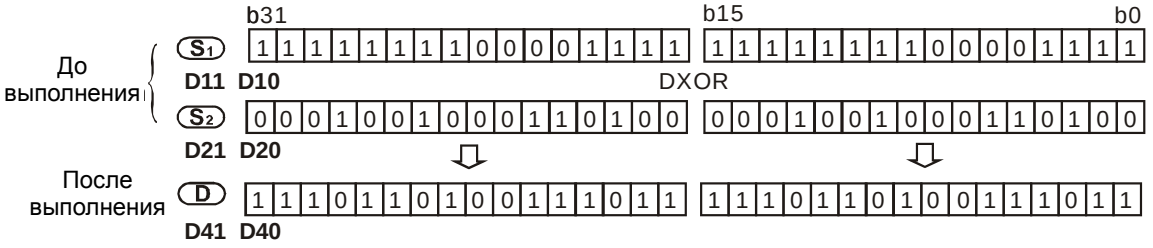
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходные данные 1 **S_2**: Исходные данные 2 **D**: Результат операции

Описание:
1. Эта команда производит логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ с операндами S1 и S2 в 32-битовом режиме, после чего сохраняет результат в D.
2. Если операнды S1, S2, D используются с индексом F, возможно выполнение только 16-битных команд.

Пример программы:
Когда X1 включен, 32-битные источники (D11, D10) и (D21,D20) анализируются и результат логической операции сохраняется в (D41, D40)





API	Команда			Операнды	Функция	Контроллеры			
29	D	NEG	P	(D)	Логическое отрицание	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	NEG, NEGP: 3 шагов DNEG, DNEGP: 5 шагов
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

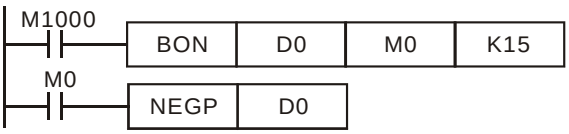
Операнды:
D: Результат

Описание:
1. Эта команда осуществляет операцию логического отрицания (инверсия всех битов в двоичном формате и сложение с 1).
2. Эта команда обычно используется в импульсном режиме выполнения (DEGP, DDEGP).
3. Если операнд D используется с индексом F, то возможно выполнение только 16-битных команд.

Пример программы 1:
Когда X0 включен, произойдет инвертирование всех битов в D10 (0 в 1 и 1 в 0), а затем будет прибавлена 1 в младший бит регистра (LSB). Результат операции будет сохранен в D10.

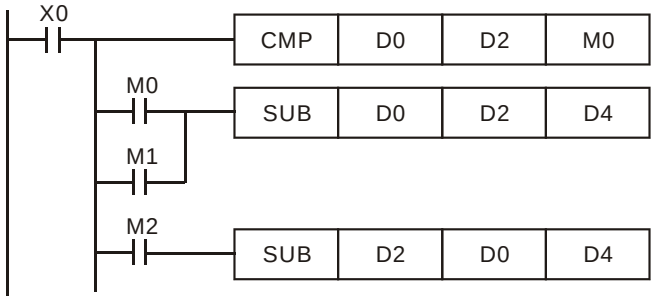


Пример программы 2:



Для получения абсолютной величины отрицательного значения: Если 15 бит в регистре D0 равен 1 (значение D0 - отрицательное), реле M0 включится. При включенном M0 команда NEG позволяет получить абсолютное значение D0.

Пример программы 3:

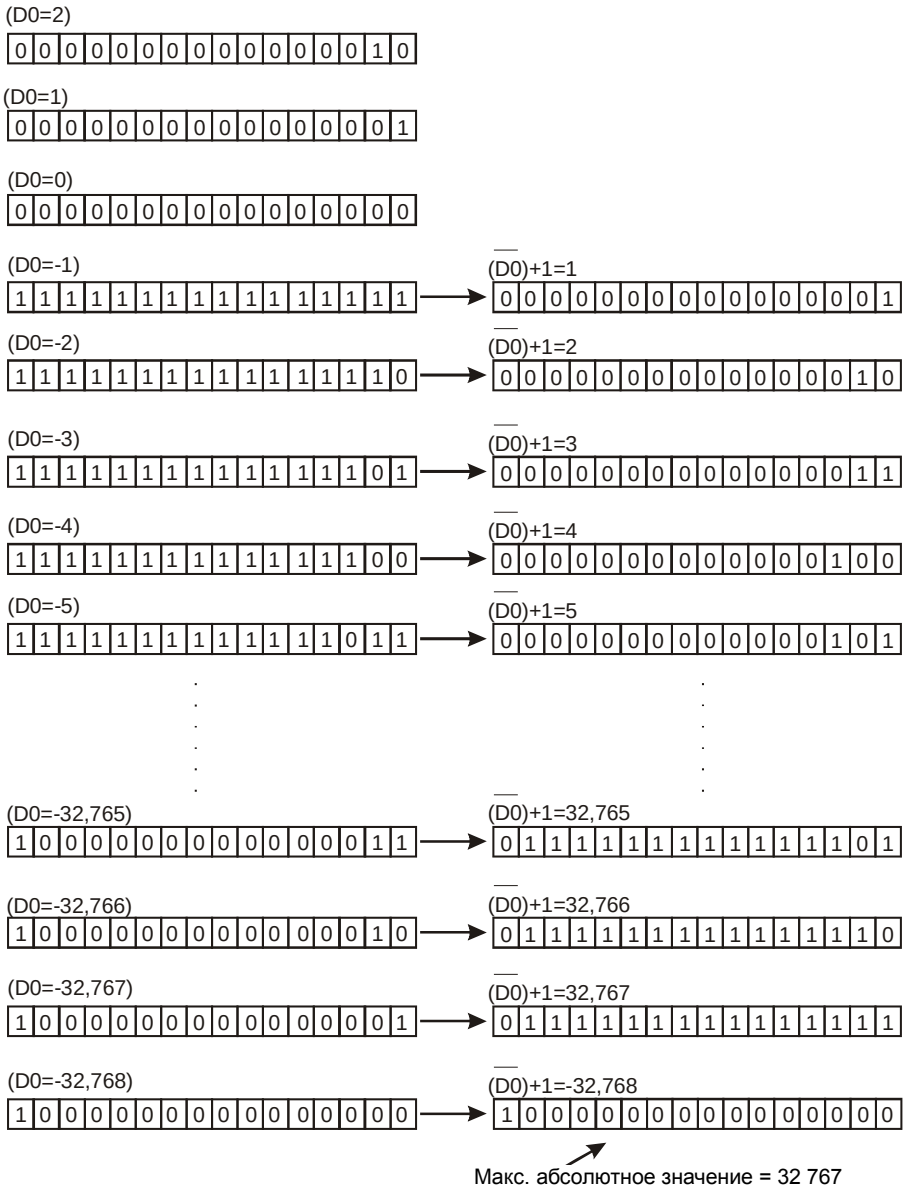


Получение абсолютного значения результата вычитания. Когда X0 включен:

a) Если D0 > D2, M0 = ВКЛ.
b) Если D0 = D2, M1 = ВКЛ.
c) Если D0 < D2, M2 = ВКЛ.
d) D4 остается положительным.

Подробное описание отрицательного значения и его абсолютного значения

1. MSB = 0 указывает на положительное значение, MSB = 1 указывает на отрицательное значение.
2. Команда NEG может быть применена для преобразования отрицательного значения в его абсолютное значение.



API	Команда			Операнды	Функция	Контроллеры			
30	D	ROR	P	(D) (n)	Кольцевой сдвиг вправо	ES2/EX2	SS2	SA2/SE	SX2

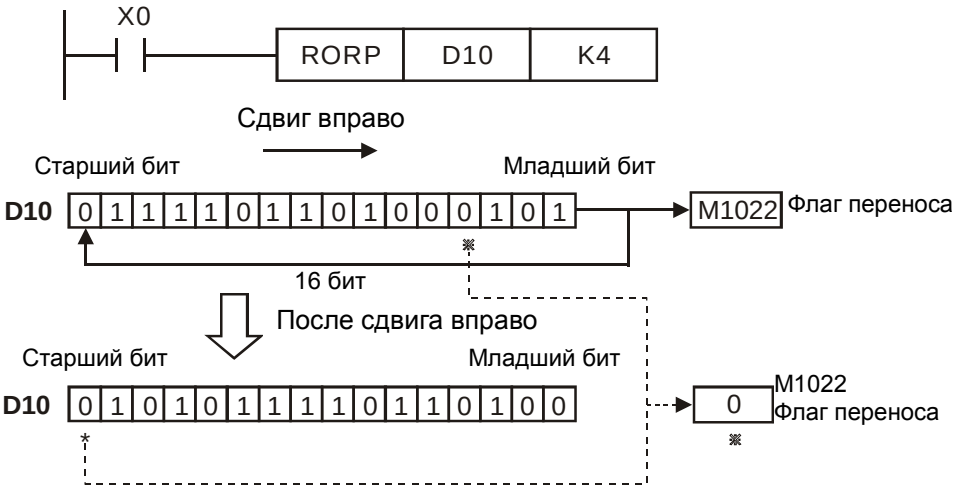
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ROR, RORP: 5 шагов DROR, DRORP: 9 шагов
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D: Операнд для кольцевого сдвига **n**: Количество сдвигаемых бит

Описание:
1. Эта команда сдвигает битовое состояние D вправо на n мест (бит).
2. Состояние последнего сдвигаемого бита (отмечен ※) копируется во флаг переноса M1022.
3. Эта команда обычно используется в импульсном режиме выполнения (RORP, DRORP).
4. Если операнд D используется с индексом F, то возможно выполнение только 16-битных команд.
5. Если операнд D - это KnY, KnM, KnS, то возможно только Kn = K4 (для 16 бит) или K8 (для 32 бит).
6. Необходимое условие для операнда n: 1≤ **n** ≤16 (16-бит), 1≤ **n** ≤32 (32-бит)

Пример программы:
Когда X0 включен, 16 бит (4 бита в группе) в D10 повернется вправо, как показано на рисунке ниже. Значение последнего бита, отмеченного ※, будет сохранен во флаге переноса M1022.



API	Команда			Операнды	Функция	Контроллеры			
31	D	ROL	P	(D) (n)	Кольцевой сдвиг влево	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ROL, ROLP: 5 шагов DROL, DROLP: 9 шагов
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

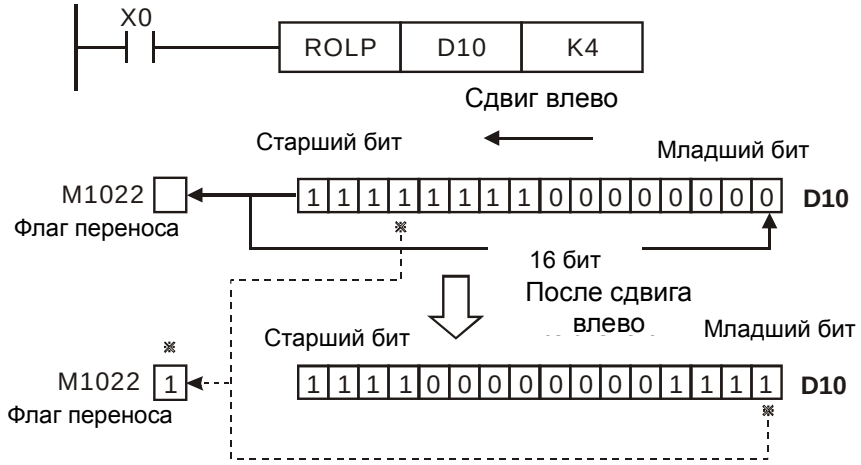
D: Операнд для кольцевого сдвига **n**: Количество сдвигаемых бит

Описание:
1. Эта команда сдвигает битовое состояние D влево на n мест (бит).
2. Состояние последнего сдвигаемого бита (отмечен ※) копируется во флаг переноса M1022.
3. Эта команда обычно используется в импульсном режиме выполнения (ROLP, DROLP).

4. Если операнд D используется с индексом F, то возможно выполнение только 16-битных команд.
5. Если операнд D это KnY, KnM, KnS, то возможно только Kn = K4 (для 16 бит) или K8 (для 32 бит).
6. Необходимое условие для операнда n: 1≤ **n** ≤16 (16-бит), 1≤ **n** ≤32 (32-бит)

Пример программы:

Когда X0 выключен, 16 бит (4 бита в группе) в D10 повернется влево, как показано на рисунке ниже. Значение последнего бита, отмеченного ※, будет сохранено во флаге переноса M1022.



API	Команда			Операнды	Функция	Контроллеры			
32	D	RCR	P	(D) (n)	Кольцевой сдвиг вправо с переносом	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	RCR, RCRP: 5 шагов DRCR, DRCRP: 9 шагов
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

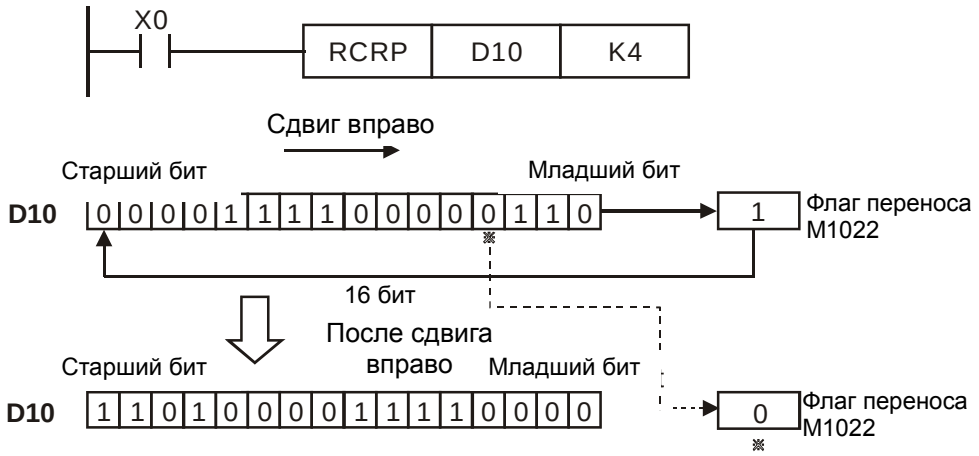
D: Операнд для кольцевого сдвига **n**: Количество сдвигаемых бит

Описание:

1. Эта команда сдвигает битовое состояние D вправо на n мест (бит) с использованием флага переноса M1022
2. Состояние последнего сдвигаемого бита (отмечен ※) копируется во флаг переноса M1022.
3. Эта команда обычно используется в импульсном режиме выполнения (RCRP, DRCRP).
4. Если операнд D используется с индексом F, то возможно выполнение только 16-битных команд.
5. Если операнд D - это KnY, KnM, KnS, то возможно только Kn = K4 (для 16 бит) или K8 (для 32 бит).
6. Необходимое условие для операнда n: 1≤ **n** ≤16 (16-bit), 1≤ **n** ≤32 (32-bit)

Пример программы:

Когда X0 выключен, 16 бит (4 бита в группе) в D10 вместе с флагом переноса M1022 (всего 17 бит) сдвинуться вправо, как показано на рисунке ниже. Значение последнего бита, отмеченного ※, будет сохранено во флаге переноса M1022.



API	Команда			Операнды	Функция	Контроллеры			
33	D	RCL	P	(D) (n)	Кольцевой сдвиг влево с переносом	ES2/EX2	SS2	SA2/ SE	SX2

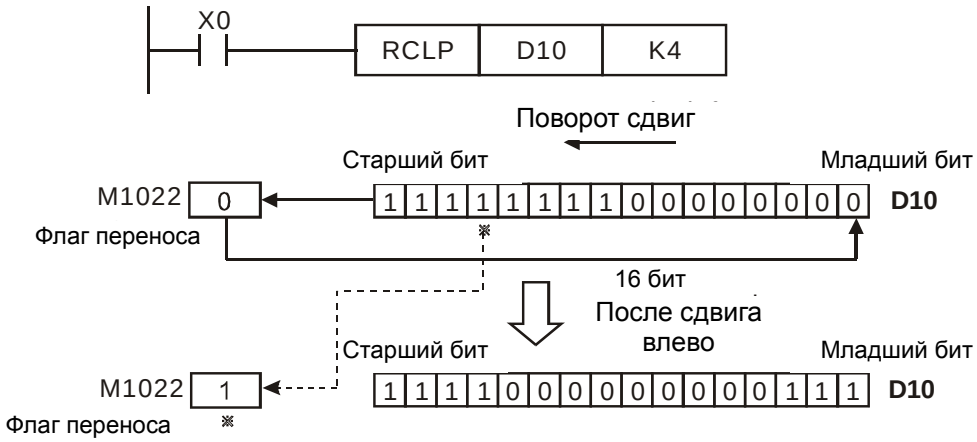
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	RCL, RCLP: 5 шагов DRCL, DRCLP: 9 шагов
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
D: Операнд для кольцевого сдвига **n**: Количество сдвигаемых бит

Описание:
1. Эта команда сдвигает битовое состояние D влево на n мест (бит) с использованием флага переноса M1022.
2. Состояние последнего сдвигаемого бита (отмечен ※) копируется во флаг переноса M1022.
3. Эта команда обычно используется в импульсном режиме выполнения (RCLP, DRCLP).
4. Если операнд D используется с индексом F, то возможно выполнение только 16-битных команд.
5. Если операнд D - это KnY, KnM, KnS, то возможно только Kn = K4 (для 16 бит) или K8 (для 32 бит).
6. Необходимое условие для операнда n: 1≤ **n** ≤16 (16-bit), 1≤ **n** ≤32 (32-bit)

Пример программы:
Когда X0 выключен, 16 бит (4 бита в группе) в D10 вместе с флагом переноса M1022 (всего 17 бит) сдвинется влево, как показано на рисунке ниже. Значение последнего бита, отмеченного, будет сохранено во флаге переноса M1022.



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
34	SFTR	P	(S) (D) (n1) (n2)	Сдвиг значений битовых операндов вправо				

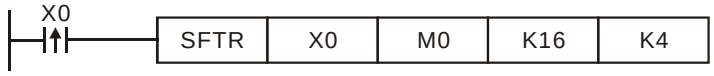
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTR, SFTRP: 9 шагов
D		*	*	*												
n_1					*	*										
n_2					*	*										

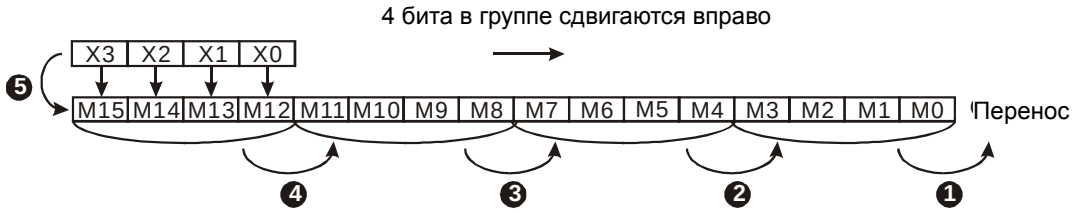
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Начальное значение источника **D**: Начальное значение цели **n_1**: Длина данных для сдвига **n_2**: Число смещаемых бит

Описание:
1. Эта команда сдвигает вправо биты длиной n1 на n2 позиций от источника S до цели, начиная со значения D.
2. Эта команда обычно используется в импульсном режиме выполнения (SFTRP).
3. Необходимое условие для операндов: **n1** и **n2**: 1≤ **n2** ≤ **n1** ≤1024

Пример программы:
1. Когда X0 включается, команда осуществляет сдвиг X0-X4 в 16-битном формате, M0-M15 и M15-M0 также сдвигаются вправо группами по 4 бит.
2. На рисунке ниже показан сдвиг вправо битов в одном цикле программы.
   - ❶ M3~M0 → Перенос
   - ❷ M7~M4 → M3~M0
   - ❸ M11~M8 → M7~M4
   - ❹ M15~M12 → M11~M8
   - ❺ X3~X0 → M15~M12 выполнено





API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
35	SFTL	P	(S) (D) (n1) (n2)	Сдвиг значений битовых операндов влево				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												SFTL, SFTLP: 9 шагов
D		*	*	*												
n_1					*	*										
n_2					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

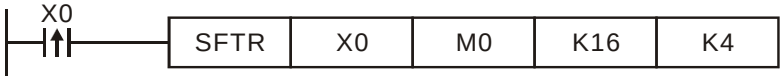
Операнды:
S: Начальное значение источника **D**: Начальное значение цели **n_1**: Длина данных для сдвига **n_2**: Число смещаемых бит

Описание:

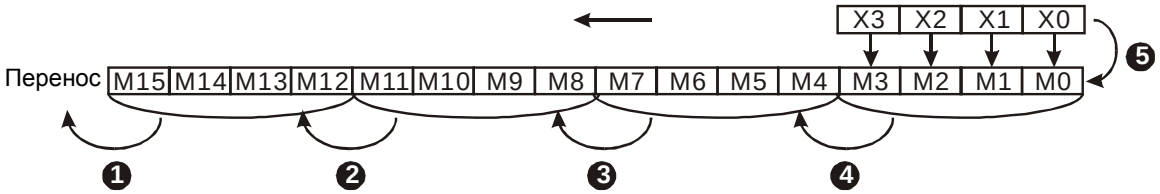
1. Эта команда сдвигает влево биты длиной n1 на n2 позиций от источника S до цели, начиная со значения D.
2. Эта команда обычно используется в импульсном режиме выполнения (SFTLP).
3. Необходимое условие для операндов: **n1**, **n2**: 1≤ **n2** ≤ **n1** ≤1024

Пример программы:

1. Когда X0 включается, команда осуществляет сдвиг X0-X4 в 16-битном формате, M0-M15 и M15-M0 также сдвигаются влево группами по 4 бит.
2. На рисунке ниже показан сдвиг влево битов в одном цикле программы.
   - ❶ M15~M12 → Перенос
   - ❷ M11~M8 → M15~M12
   - ❸ M7~M4 → M11~M8
   - ❹ M3~M0 → M7~M4
   - ❺ X3~X0 → M3~M0 выполнено



4 бита в группе сдвигаются влево



API	Команда			Операнды	Функция	Контроллеры			
36		WSFR	P	(S) (D) (n1) (n2)	Пословный сдвиг значения регистра вправо	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			WSFR, WSFRP: 9 шагов
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

S: Начальное значение источника **D**: Начальное значение цели **n_1**: Длина данных для сдвига **n_2**: Число смещаемых бит

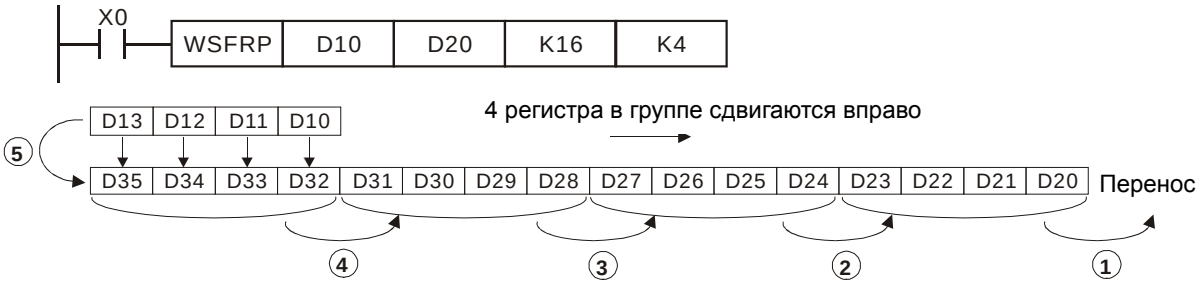
Описание:

1. Эта команда сдвигает пословно вправо биты длиной n1 на n2 позиций от источника S до цели, начиная со значения D.
2. Эта команда обычно используется в импульсном режиме выполнения (WSFRP).
3. Присвоенные значения S и D должны быть одинаковыми, например KnX, KnY, KnM, KnS в одной категории и T, C, D в другой категории.
4. Если S и D присвоены значения типа Kn, количество знаков Kn в S и D должно быть одинаковым.
5. Необходимое условие для операндов: **n1** и **n2:** 1≤ **n2** ≤ **n1** ≤512

Пример программы 1:

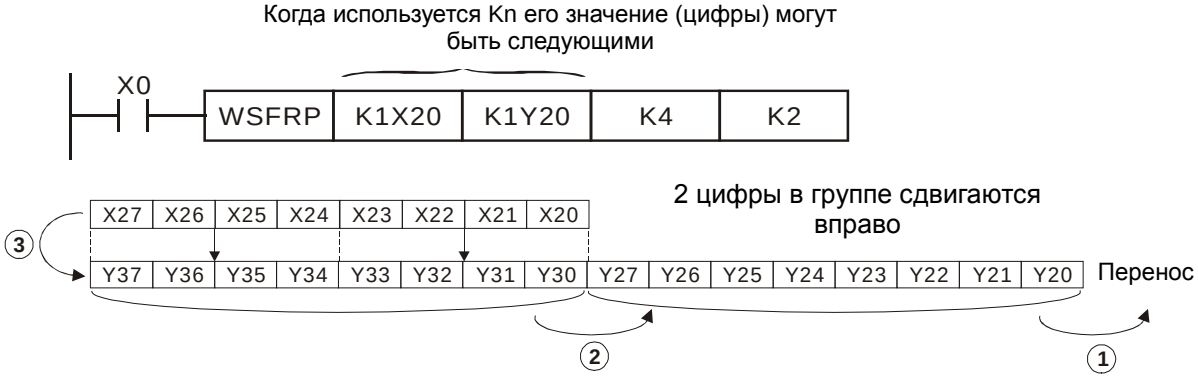
1. Когда срабатывает X0, данная команда пословно сдвигает D10-D13 в данных регистра D20-D35 и D20-D35 также сдвигаются вправо группами по 4 регистра.
2. На рисунке ниже показан сдвиг вправо регистров в одном цикле программы.
   - ❶ D23~D20 → Перенос
   - ❷ D27~D24 → D23~D20
   - ❸ D31~D28 → D27~D24

❹ D35~D32 → D31~D28
❺ D13 ~D10 → D35~D32 выполнено



Пример программы 2:

1. Когда срабатывает X0, данная команда пословно сдвигаетX20-X27 в данных регистра Y20-Y37 и Y20-Y37 также сдвигаются вправо группами по 4 регистра.
2. На рисунке ниже показан сдвиг вправо регистров в одном цикле программы.
   ❶ Y27~Y20 → Перенос
   ❷ Y37~Y30 → Y27~Y20
   ❸ X27~X20 → Y37~Y30 выполнено



API	Команда		Операнды	Функция	Контроллеры			
37	WSFL	P	S D n_1 n_2	Пословный сдвиг значения регистра влево	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	WSFL, WSFLP: 9 шагов
S							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n_1					*	*										
n_2					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

S: Начальное значение источника **D**: Начальное значение цели **n_1**: Длина данных для сдвига **n_2**: Число смещаемых бит

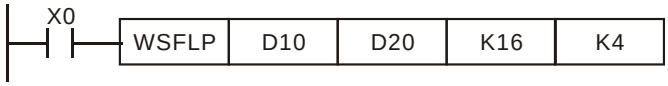
Описание:

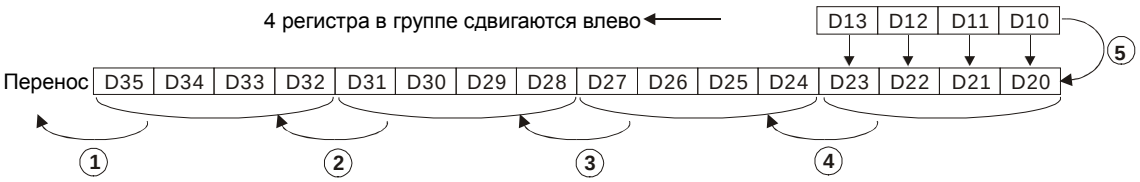
1. Эта команда сдвигает пословно влево биты длиной n1 на n2 позиций от источника S до цели, начиная со значения D.
2. Эта команда обычно используется в импульсном режиме выполнения (WSFLP).
3. Присвоенные значения S и D должны быть одинаковыми, например KnX, KnY, KnM, KnS в одной категории и T, C, D в другой категории.
4. Если S и D присвоены значения типа Kn, количество знаков Kn в S и D должно быть одинаковым.
5. Необходимое условие для операндов: **n1** и **n2**: 1≤ **n2** ≤ **n1** ≤512

Пример программы:

1. Когда срабатывает X0, данная команда пословно сдвигает D10-D13 в данных регистра D20-D35 и D20-D35 также сдвигаются влево группами по 4 регистра.
2. На рисунке ниже показан сдвиг влево битов в одном цикле программы.
   ❶ D35~D32 → Перенос

❷ D31~D28 → D35~D32
❸ D27~D24 → D31~D28
❹ D23~D20 → D27~D24
❺ D13~D10 → D23~D20 выполнено





API	Команда			Операнды	Функция	Контроллеры			
38		SFWR	P	S D n	Запись данных в стек	ES2/EX2	SS2	SA2/ SE	SX2

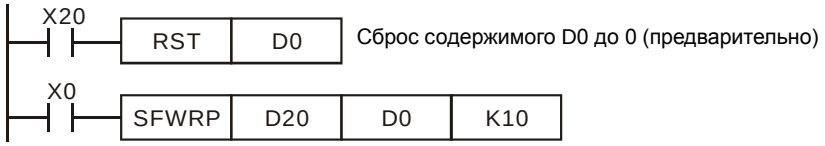
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SFWR, SFWRP: 7 шагов
D								*	*	*	*	*	*			
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных **D**: Главный адрес стека данных **n**: Длина стека

Описание:
1. Данная команда записывает данные стека из n слов в стековую память типа FIFO (First-in/First-OUT). Первым адресом стека является D. Когда команда выполнена, содержимое указателя увеличивается на 1 и содержимое S будет переписано в указателе. Если содержимое стека превышает n-1 (D используется также в качестве указателя стека), выполнение команды прерывается и включается флаг переноса M1022.
2. Эта команда обычно используется в импульсном режиме выполнения (SFWRP).
3. Необходимое условие для операндов: **n**: 2≤ **n** ≤512

Пример программы:
1. Сначала происходит сброс содержимого D0. Когда X0 включается, содержимое указателя D0 принимает значение 1, а D20 записывается в D1. Если содержимое D20 изменяется и X0 включается снова, указатель D0 становится равным 2, а содержимое D20 записывается в D2.
2. На рисунке ниже показан сдвиг и запись в процессе выполнения команды.
   ❶ Содержимое D0 становится равным 1.
   ❷ Содержимое D20 записывается в D1.





Замечание: Эта команда может применяться вместе с API 39 SFRD для чтения/записи FIFO памяти стека данных.

API	Команда		Операнды	Функция	Контроллеры			
39	SFRD	P	(S) (D) (n)	Чтение данных из стека	ES2/EX2	SS2	SA2/SE	SX2

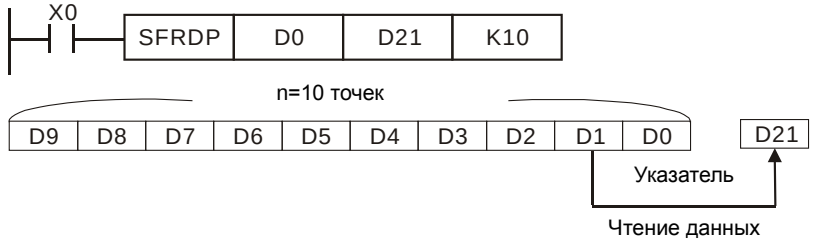
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			SFRD, SFRDP: 7 шагов
D								*	*	*	*	*	*	*	*	
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Главный адрес стека данных **D**: Результат **n**: Длина стека

Описание:
1. Данная команда читает данные из стека в n слов из стековой памяти FIFO. Первым адресом является S. S используется также в качестве указателя. Когда команда выполняется, первые данные из S (S+1) будут считаны в D, содержимое указателя S уменьшается на 1.При значении указателя, равном 0, выполнение команды прерывается и включается флаг переноса M1022.
2. Эта команда обычно используется в импульсном режиме выполнения (SFRDP).
3. Необходимое условие для операндов: **n**: 2≤ **n** ≤512

Пример программы:
1. Когда X0 включается, данные D9-D2 сдвигаются вправо и указатель D0 уменьшается на 1, содержимое D1 считывается и передвигается в D21.
2. На рисунке ниже показан процесс сдвига и чтения при выполнении команды.
   ❶ Содержимое D1 считывается и перемещается в D21.
   ❷ D9-D2 смещается вправо.
   ❸ Содержимое D0 уменьшается на 1.



API	Команда		Операнды	Функция	Контроллеры			
40	ZRST	P	(D_1) (D_2)	Групповой сброс операндов	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D_1		*	*	*							*	*	*			ZRST, ZRSTP: 5 шагов
D_2		*	*	*							*	*	*			

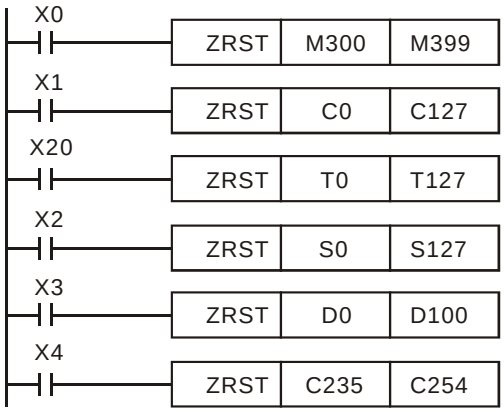
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D_1: Первый операнд диапазона сброса D_2: Последний операнд диапазона сброса

Описание:
1. При выполнении команды диапазон D1-D2 будет сброшен.
2. Операнды D1 и D2 должны быть однотипны, необходимое условие: $D_1 \leqq D_2$
3. Если D1 > D2 отключается только операнд D2.
4. Эта команда обычно используется в импульсном режиме выполнения (ZRSTP).

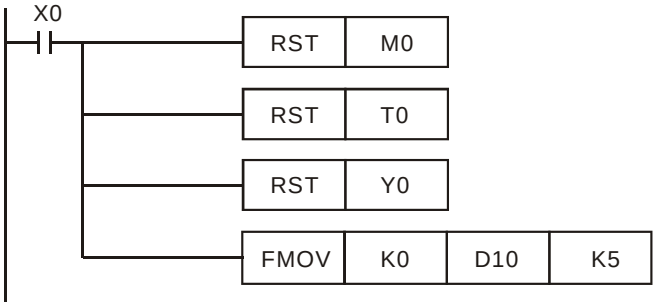
Пример программы:
1. Когда X0 = ON, M300 - M399 будут сброшены.
2. Когда X1 = ON, C0 - C127 будут сброшены, т.е. текущее значение равно 0 и связанный с ним контакт / выход будет также сброшен.
3. Когда X20 = ON, T0 - T127 будут сброшены, т.е. текущее значение равно 0 и связанный с ним контакт / выход будет также сброшен.

4. Когда X2 = ON, шаги S0-S127 будут сброшены.
5. Когда X3 = ON, данные D0 - D100 будут сброшены.
6. Когда X4 = ON, C235 - C254 будут сброшены, т.е. текущее значение равно 0 и связанный с ним контакт / выход будет также сброшен.



Замечания:

1. Биты Y, M, S и слова T,C, D могут быть сброшены при помощи RST-команды.
2. Групповой сброс битов KnY, KnM, KnS или слов T, C, D может быть осуществлен командой FMOV (API 16).



API	Команда		Операнды	Функция	Контроллеры			
41	DECO	P	(S) (D) (n)	Дешифратор 8 - 256 бит	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*	*	*					*	*	*	*	*	DECO, DECOP: 7 шагов
D		*	*	*							*	*	*	*	*	
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

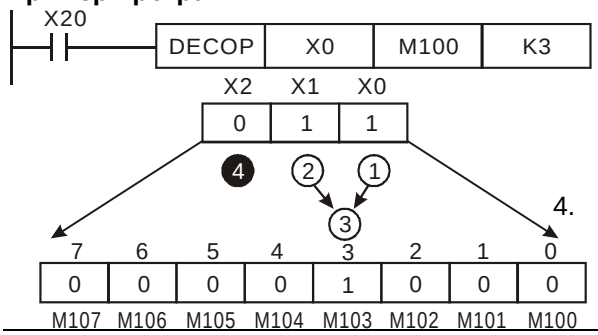
Операнды:

S: Источник данных для дешифровки **D**: Адрес хранения результата **n**: Количество последовательных бит S.

Описание:

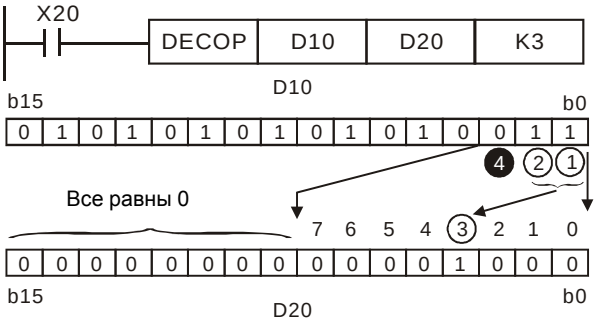
1. Команда дешифрует n младших битов S и сохраняет результат в 2n операндах адреса цели D.
2. Эта команда обычно используется в импульсном режиме выполнения (DECOP).
3. Когда операнд D битовый, **n** = 1~8, а когда операнд D словный, **n** = 1~4.

Пример программы 1:



1. Когда D является битным, n=1-8, а ошибка возникает при n=0 или n>8.
2. Если n = 8, дешифрованные данные - 2^8 = 256 бит данных.
3. Когда X20 включается, данные X0-X2 будут дешифрованы в M100-M107. Если значение исходного операнда равно 3, включается M103 (3-й бит M100). После завершения, X20 отключается. Дешифрованные данные будут сохранены.

Пример программы 2:



1. Когда D является словным, n=1-4, и ошибка возникает при n=0 или n>4.
2. Если n = 4, дешифрованные данные - 2^4 = 16 бит данных.
3. Когда X20 включается, данные в D10 (b2-b0) будут дешифрованы и сохранены в D20 (b7-b0). Неисполь- зуемые биты в D20 (b15-b8) будут = 0.
4. Младшие 3 бита D10 дешиф- руются и сохраняются в младшие 8 бит D20. Старшие 8 бит D20 равны 0.
5. После завершения X20 отключается. Дешифрованные данные будут сохранены.

API	Команда		Операнды	Функция	Контроллеры			
42	ENCO	P	S D n	Шифратор 256 - 8 бит	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DECO, DECOP: 7 шагов
S	*	*	*	*							*	*	*	*	*	
D											*	*	*	*	*	
n					*	*										

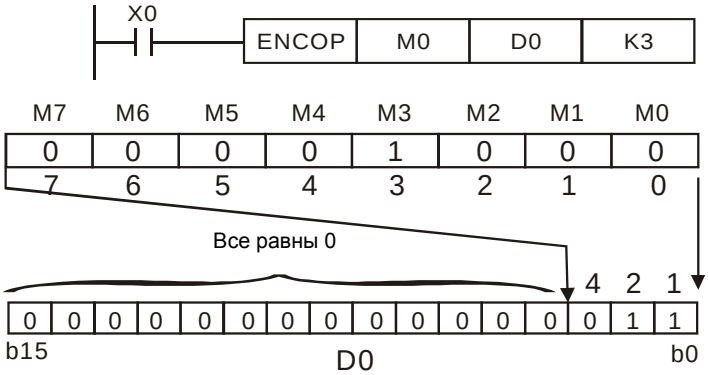
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных для шифровки **D**: Адрес хранения результата **n**: Количество последовательных бит S.

Описание:

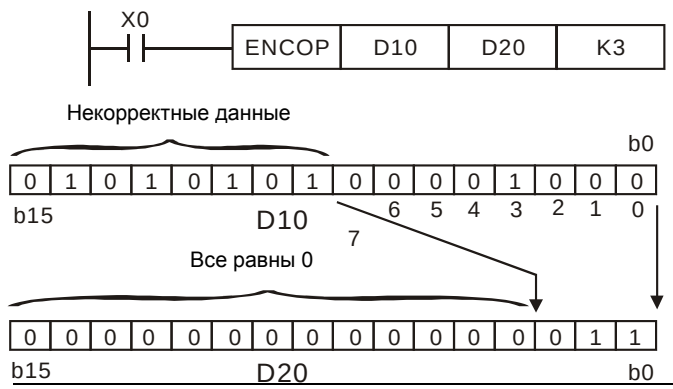
1. Команда кодирует 2^n младших битов S и сохраняет результат в адресе цели D.
2. Старшие биты в S имеют приоритет для кодирования.
3. Эта команда обычно используется в импульсном режиме выполнения (ENCOP).
4. Когда операнд S является битным, n=1-8, а когда словным, n=1-4.
5. Если ни один бит в S не является активным (статус 1), включаются флаги M1067 и M1068 и записывается код ошибки 0E1A (шестнадцатеричный).

Пример программы 1:



1. Когда S является битным, n=1-8, и ошибка возникает при n=0 или n>8.
2. Если n = 8, дешифрованные данные - 2^8 = 256 бит данных.
3. Когда X0 включается, данные в (M0-M7) будут кодированы и сохранены в 3 младших бита D0 (b2-b0). Неиспользуемые биты в D0 (b15-b3) будут равны 0.
4. После завершения X0 выклю- чается, и данные в D будут сохранены.

Пример программы 2:



1. Когда D является словным, n=1-4, а ошибка возникает при n=0 или n>4.
2. Если n = 4, дешифрованные данные - 2^4 = 16 бит данных.
3. Когда X0 включается, 23 бита (b0-b7) в D10 будут кодированы и сохранены в 3 младших бита D20 (b2-b0). Неиспользуемые биты в D0 (b15-b3) будут равны 0.
4. После завершения X0 выключается и данные в D будут неизменны.

API	Команда			Операнды	Функция	Контроллеры			
43	D	SUM	P	(S) (D)	Сумма активных битов	ES2/EX2	SS2	SA2/SE	SX2

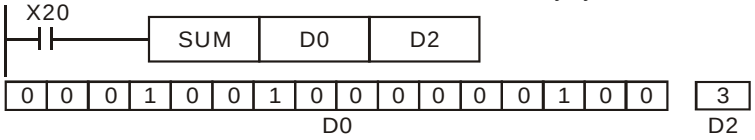
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SUM, DSUMP: 5 шагов DSUM, DSUMP: 9 шагов
D											*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат

Описание:
1. Эта команда подсчитывает общее число активных битов S и сохраняет результат в D.
2. D занимает 2 регистра при использовании 32-битной команды.
3. Если операнд D используется с индексом F, доступна только 16-битная команда.
4. Если нет активных битов, включается флаг нуля M1020.

Пример программы:
Когда X20 включен, все активные биты D0 будут подсчитаны и результат сохранен в D2.



API	Команда			Операнды	Функция	Контроллеры			
44	D	BON	P	(S) (D) (n)	Проверка состояния битов	ES2/EX2	SS2	SA2/SE	SX2

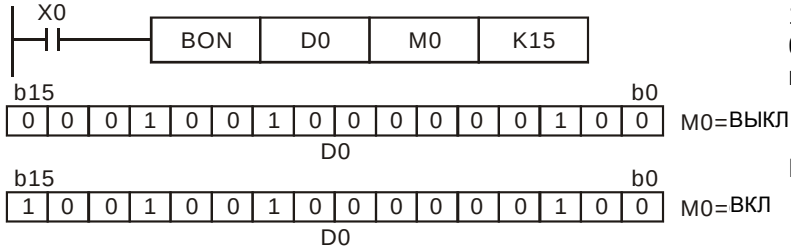
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	BON, BONP: 7 шагов DBON, DBONP: 13 шагов
D		*	*	*												
n					*	*					*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Адрес хранения результата **n**: Количество проверенных битов

Описание:
1. Команда проверяет состояние отдельных битов (в количестве n) в S и сохраняет результат в D.
2. Если операнд D используется с индексом F, возможно выполнение только 16-битных команд.
3. Необходимое условие для операнда n: **n** = 0~15 (16-бит), **n** = 0~31 (32-бит)

Пример программы:



1. Когда X0 включен и 15 бит D0 равны “1”, M0 включается. Если 15 бит равны “0”, M0 отключается.
2. Когда X0 выключен, M0 сохраняет прежнее состояние.

API	Команда			Операнды	Функция	Контроллеры			
45	D	MEAN	P	(S) (D) (n)	Среднее арифметическое	ES2/EX2	SS2	SA2/ SE	SX2

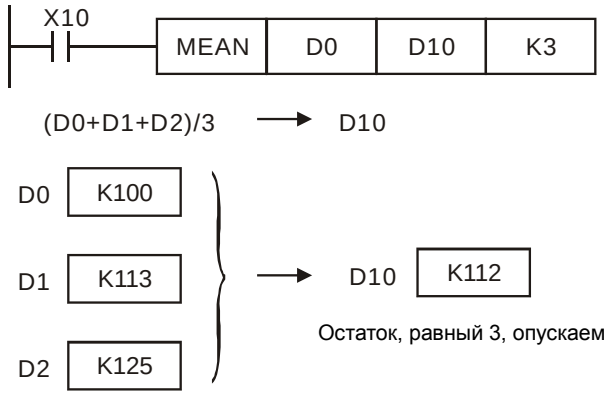
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MEAN, MEANP: 7 шагов DMEAN, DMEANP: 13 шагов
D								*	*	*	*	*	*	*	*	
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных **D**: Адрес хранения результата **n**: Количество последовательных слов данных с S.

Описание:
1. Команда рассчитывает среднее арифметическое значение n последовательных слов данных и сохраняет результат в D.
2. Остаток после получения целого числа будут проигнорированы.
3. Если количество слов n в S превышает допустимый диапазон, будут обработаны только адреса, входящие в допустимый диапазон.
4. Если n находится вне допустимого диапазона (1-64), ПЛК выдаст ошибку.
5. Если операнд D используется с индексом F, то возможно выполнение только 16-битных команд.
6. Необходимое условие для операнда n: **n** = 1~64

Пример программы:



Когда X0 включен, содержимое 3 (n=3) регистров, начиная с D0, будут суммированы и затем разделены на 3 для получения среднего значения. Результат будет сохранен в D10, остаток будет отброшен.

API	Команда	Операнды	Функция	Контроллеры			
46	ANS	(S) (m) (D)	Тревожная сигнализация с задержкой на включение	ES2/EX2	SS2	SA2/ SE	SX2

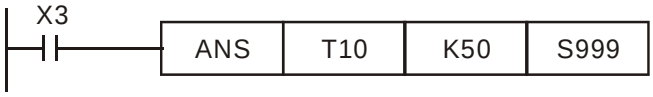
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					ANS: 7 шагов
m					*											
D				*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Таймер тревожной сигнализации **m**: Установка времени **D**: Тревога (сигнальный бит)

Описание:
1. Эта команда используется для управления подачей тревожной сигнализацией в назначенное время.
2. Диапазон операнда S: T0~T183; m: K1~K32,767 (умноженный на 100 мс); D: S912~S1023
3. Флаги: M1048 (Включен: Сигнализация активируется), M1049 (Включен: Включено управление сигнализацией)
4. См. команду ANR для получения дополнительной информации.

Пример программы:



Если X3 включен в течение более чем 5 сек., реле сигнализации S999 будет включено и останется включенным после возврата X3 в исходное состояние (T10 сбросится, текущее значение "0").

API	Команда		Функция	Контроллеры			
47	ANR	P	Сброс сигнализации	ES2/EX2	SS2	SA2/ SE	SX2

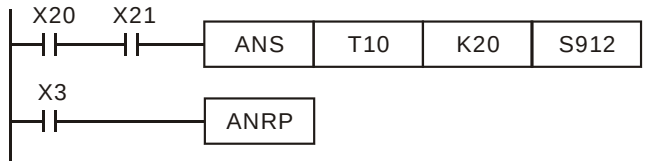
Операнд	Описание	Шаги программы
НЕТ	Команда управляет контактом тревожной сигнализации	ANR, ANRP: 1 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Описание:

1. Данная команда используется для сброса сигнала тревоги. Когда включены несколько адресов сигнализации, будет сброшена сигнализация с самого младшего адреса.
2. Эта команда обычно используется в импульсном режиме выполнения (ANRP).

Пример программы:



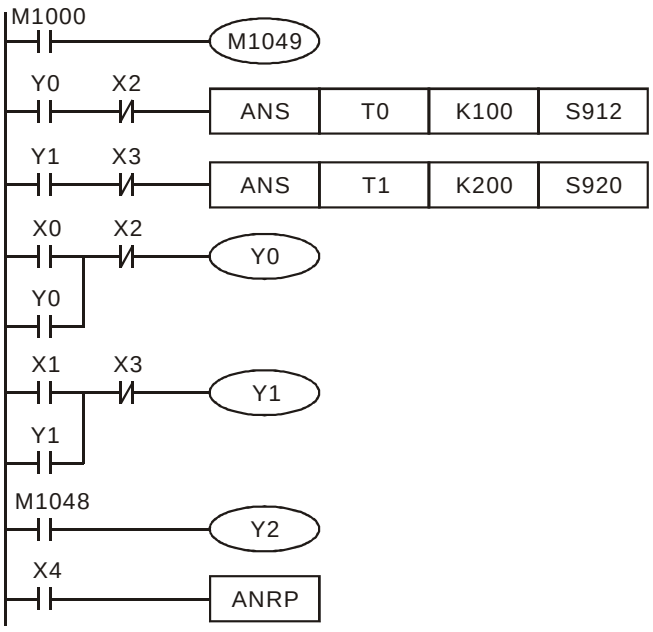
1. Если X20 и X21 включены одновременно более 2 сек., тревожный сигнал с S912 включается. Если X20 или X21 сбрасываются, сигнал с S912 будет оставаться включенным, но T10 будет сброшено и текущее состояние очистится.
2. Если X20 и X21 включены одновременно менее 2 сек., текущее состояние T10 будет очищено.
3. Когда X3 включается, активные тревожные сигналы S912 будут сброшены.
4. Когда X3 снова включается, тревожный сигнал следующего младшего адреса будет сброшен.

Заметки:

Флаги:

1. M1048 (индикация флага тревоги): Когда M1049 включен, и любой из операндов тревоги S912-S1023 включен, флаг M1048 включается.
2. M1049 (Активация флага тревоги): Когда M1049 включен, D1049 будет автоматически удерживать младшие адреса из активных тревог.

Пример применения команды тревожной сигнализации в производстве:



X0 = Пуск вперед
X1 = Пуск назад
X2 = Концевой выключатель движения вперед
X3 = Концевой выключатель движения назад
X4 = Кнопка сброса тревожной сигнализации
Y0 = Движение вперед
Y1 = Движение назад
Y2 = Индикация аварии
S912 = Сигнализация превышения времени движения вперед
S920 = Сигнализация превышения времени движения назад

1. M1046 и D1049 существуют только при включенном M1049.
2. Когда Y0 включен более 10 сек. и положение концевого выключателя X2 при движении вперед не достигнуто, S912 включается

3. Когда Y1 включен более 10 сек. и положение концевого выключателя X3 при движении назад не достигнуто, S920 включатся.
4. Когда пуск назад X1 и Y1 включены, Y1 выключится только при достижении при движении заднего концевого выключателя X3.
5. Y2 включен при любом включенном тревожном сигнале.
6. Когда X4 включен, один тревожный сигнал будет сброшен. При наличии нескольких активных сигналов сброс начнется с сигнала самого младшего адреса, потом следующего за ним младшего и так по порядку.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
48	D	SQR	P	(S) (D)	Вычисление квадратного корня				

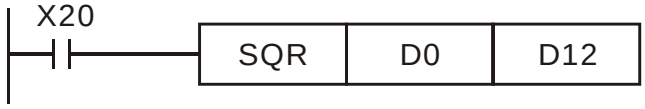
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			SQR, SQRP: 5 шагов DSQR, DSQRP: 9 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Адрес хранения результата

Описание:
1. Эта команда вычисляет квадратный корень из S и сохраняет результат в D.
2. **S** может иметь только положительное значение. Попытка вычисления квадратного корня из отрицательного значения приведет к ошибке, и команда не будет выполнена. Флаг ошибки M1067, M1068 включены и в D1067 записывается код ошибки H0E1B.
3. Результат операции D должен быть только целым числом, десятинные знаки будут отброшены. Когда такие знаки после запятой отсутствуют, включается флаг M1021.
4. Когда результат **D** = 0, флаг нуля M1020 включается.

Пример программы:



Когда X20 включен, квадратный корень D0 сохраняется в D12.

$\sqrt{D0} \rightarrow D12$

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
49	D	FLT	P	(S) (D)	Преобразование целого числа в число с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			FLT, FLTP: 5 шагов DFLT, DFLTP: 9 шагов
D													*			

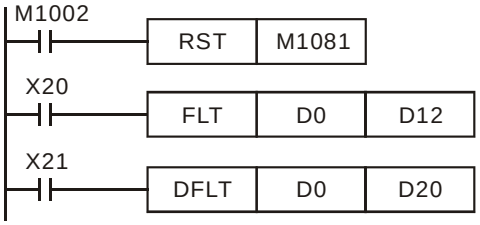
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Адрес хранения результата преобразования

Описание:
1. Когда M1018 выключен, данные S преобразуются из целого двоичного числа в двоичное число с плавающей запятой. При этом 16-битная команда занимает 1 регистр S и 2 регистра D.
2. Если абсолютное значение результата преобразования $>$ максимально допустимого значения, флаг переноса M1022 включается.
3. Если абсолютное значение результата преобразования $<$ минимально допустимого значения, флаг переноса M1021 включается.
4. Если абсолютное значение результата преобразования равно 0, флаг нуля M1020 включается.

5. Когда M1018 включен, данные S преобразуются из двоичного числа с плавающей запятой в целое двоичного число (десятичные знаки отбрасываются). При этом, 16-битная команда занимает 2 регистра S и 1 регистр D, аналогично команде INT.
6. Если результат превышает доступный диапазон целого двоичного числа D (для 16-бит: -32,768 ~ 32,767; для 32-бит: -2,147,483,648 ~ 2,147,483,647), D получит максимальное или минимальное значение и включится флаг переноса M1022.
7. Если десятичные игнорируются, включится флаг M1021.
8. Если результат равен 0, включится флаг нуля M1020.
9. После преобразования, D сохраняет результат в 16-битном формате.

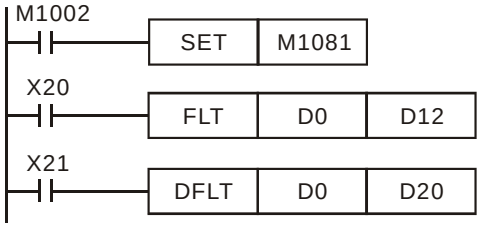
Пример программы 1:



1. Когда M1081 выключен, целое двоичное число преобразуется в двоичное число с плавающей запятой.
2. Когда X20 включен, D0 преобразуется в D13, D12 (с плавающей запятой).
3. Когда X21 включен, D1, D0 преобразуются в D21, D20 (с плавающей запятой).
4. Считаем D0 как K10. Когда X10 включен, преобразованное 32-битное значение составит H41200000 и сохранится в 32-битном регистре D12 (D13).
5. Если 32-битный регистр D0 (D1)=K100,000, X21 включается. 32-битное число с плавающей запятой после преобразования составит H47C35000 и сохранится в 32-битном регистре D20 (D21).

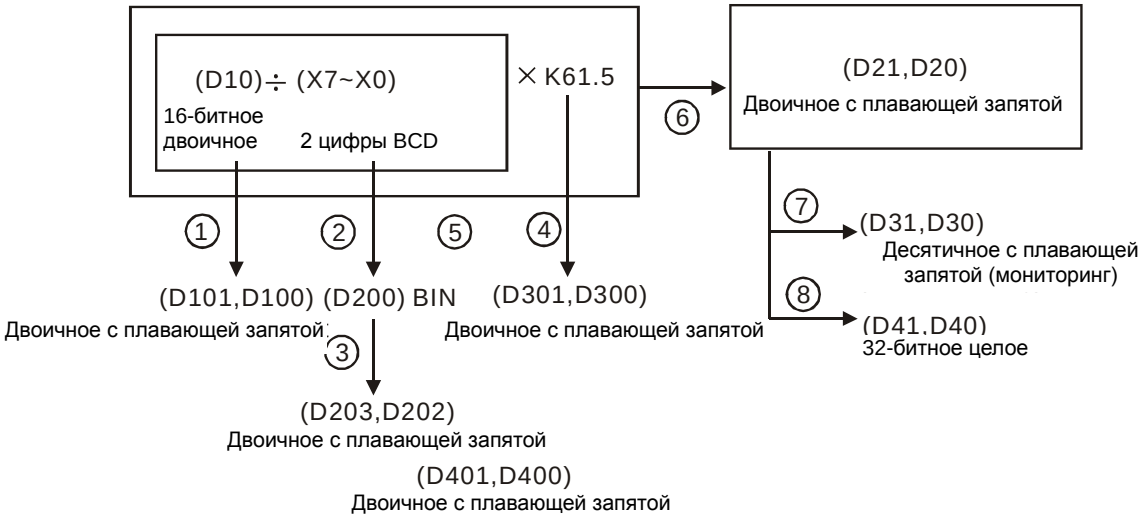
Пример программы 2:

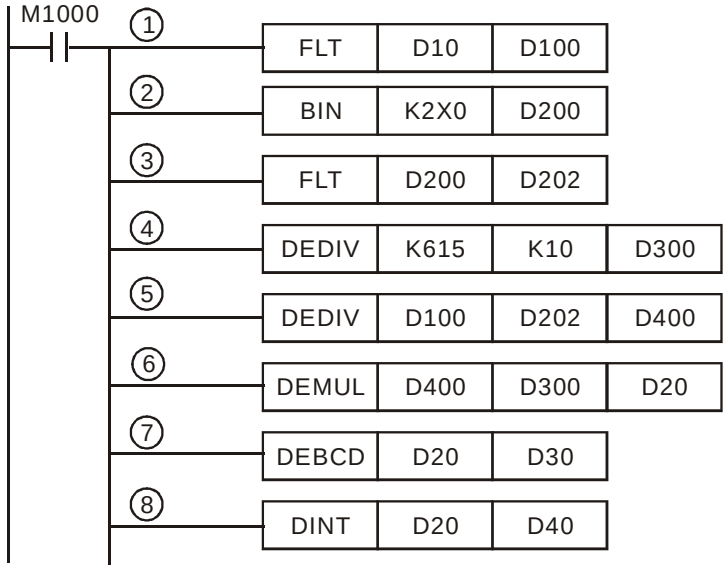
1. Когда M1081 включен, исходные данные преобразуются в целое двоичное число с плавающей запятой (десятичные отбрасываются).
2. Когда X20 включен, D1 и D0 (с плавающей запятой) преобразуются в D12 (целое двоичное). Если D0 (D1) = H47C35000, результат составит 100,000 что превышает допустимый диапазон двоичных чисел в 16-битном регистре D12. В этом случае результат составит D12 = K32767 и включится флаг M1022.
3. Когда X21 включен, D1 и D0 (с плавающей запятой) преобразуются в D21, D20 (целое двоичное). Если D0 (D1) = H47C35000, результат составит 100,000 и будет сохранен в 32-битный регистр D20 (D21).



Пример программы 3:

Применение команды FLT для выполнения следующих операций:





1. Преобразовать D10 (целое двоичное) в D101, D100 (с плавающей запятой).
2. Преобразовать диапазон X7~X0 (двоично-десятичное) в D200 (двоичное).
3. Преобразовать D200 (целое двоичное) в D203, D202 (с плавающей запятой).
4. Сохранение результата K615 ÷ K10 в D301, D300 (с плавающей запятой).
5. Деление чисел с плавающей запятой:
   Сохранение результата (D101, D100) ÷ (D203, D202) в D401, D400 (с плавающей запятой).
6. Умножение чисел с плавающей запятой:
   Сохранение результата (D401, D400) × (D301, D300) в D21, D20 (с плавающей запятой).
7. Преобразование числа с плавающей запятой (D21, D20) в десятичное с плавающей запятой (D31, D30).
8. Преобразование числа с плавающей запятой (D21, D20) в целое двоичное (D41, D40).

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
50	REF	P	(D) (n)	Обновление состояния входов/выходов				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D	*	*														REF, REFP: 5 шагов
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

D: Начальный адрес обновляемых входов/выходов **n**: Количество обновляемых входов/выходов

Описание:

1. ПЛК обновляет состояние входов/выходов между циклами программы (после выполнения команды END). Если необходимо обновление в других случаях, команда REF осуществляет эту операцию немедленно.
2. Операнд **D** может быть только кратным 10, т.е. X0, Y0. Команда не работает с входами/выходами дискретных модулей вх/вых.
3. Только входы/выходы MPU могут быть заданы для операнда **D**.
   - Когда **D** определяется X0 и **n**<=8, будут обновлены только X0-X7. Если **n**>8, будут обновлены все входы/выходы.
   - Когда **D** определяется Y0 и **n** = 8, будут обновлены только Y0~X7. Если **n** > 8, будут обновлены все входы/выходы.
   - Когда **D** определяется X10 или Y10, входы/выходы MPU будут обновляться все вне зависимости от значения n за исключением X0-X7 или Y0-Y7.
4. Только для EX2/SX2 MPU: Если M1180 включен и команда REF выполняется, ПЛК будет считывать A/D значения и обновит D1110-D1113. Если M1181 включен и команда REF выполняется, ПЛК выведет D/A значения в D1116 и D111 немедленно. Когда A/, D/A значения обновлены, M1180 и M1181 автоматически сбрасываются.
5. Диапазон для **n (ES2/EX2):** 4 ~ все входы/выходы MPU. **n** всегда кратно 4.
6. Диапазон для **n (SS2/SA2/SE/SX2):** 8 все входы/выходы MPU.

Пример программы 1:



Когда X0 включен, ПЛК обновляет состояние входов X0 ~ X7 без задержки.

Пример программы 2:



Когда X0 включен, 4 выходных сигнала на Y0 ~ Y3 будут оправлены на выходные клеммы непосредственно перед выполнением программой команды END.

Пример программы 3:

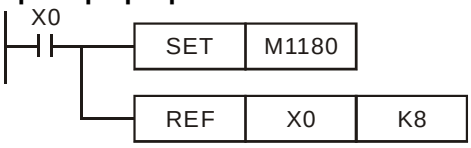


Когда X0 включен, входы/выходы, начиная с X10 или Y4, будут обновлены.

ИЛИ



Пример программы 4:



Только для DVP-EX2/SX2: Когда X0 включен и M1180 включен, A/D сигнал в D1110~D1113 будут обновлены немедленно независимо от настроек n и D.

API	Команда		Операнды	Функция	Контроллеры			
51	REFF	P	n	Изменение времени входного фильтра	ES2/EX2	SS2	SA2/SE	SX2

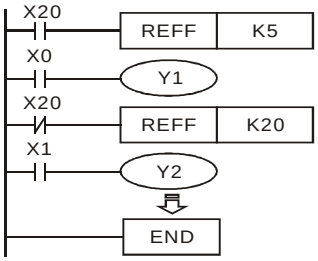
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
n					*	*										REFF, REFFP: 3 шага

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
n: Время фильтра (мс)

Описание:

1. ПЛК выполняет цифровую фильтрацию входного сигнала во избежание помех. Время фильтрации (n) входных фильтров X0-X7 регулируется командой REFF. Команда устанавливает значения n в D1020 напрямую.
2. Когда ПЛК включается или команда END выполнена, значение времени фильтрации определяется значением D1020.
3. Во время выполнения программы значение D1020 может быть изменено с помощью команды MOV.
4. При выполнении команды REFF во время выполнения программы, изменение времени фильтрации будет переноситься в D1020 и обновляться до следующего цикла программы.
5. Диапазон **n**: = K2 ~ K20.



Пример программы:

1. При подаче питания на ПЛК, время фильтрации входов X0~X7 определяется значением в D1020.
2. Когда X20 включен, команда REFF K5 выполняется, время изменяется до 5 мс и остается таким на следующий цикл программы.
3. Когда X20 выключен, команда REFF не будет выполнена, время будет установлено в 20 мс и таким же останется в следующем цикле программы.

Заметки: Когда используются входные прерывания, инструкции высокоскоростного счета или команда SPD, то для входов используемых этими командами входной фильтр не действует.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
52	MTR	S D_1 D_2 n	Матричный ввод				

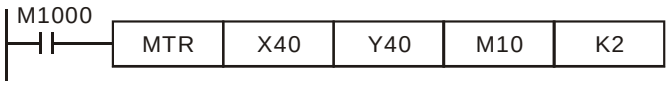
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															MTR: 9 шагов
D_1		*														
D_2		*	*	*												
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

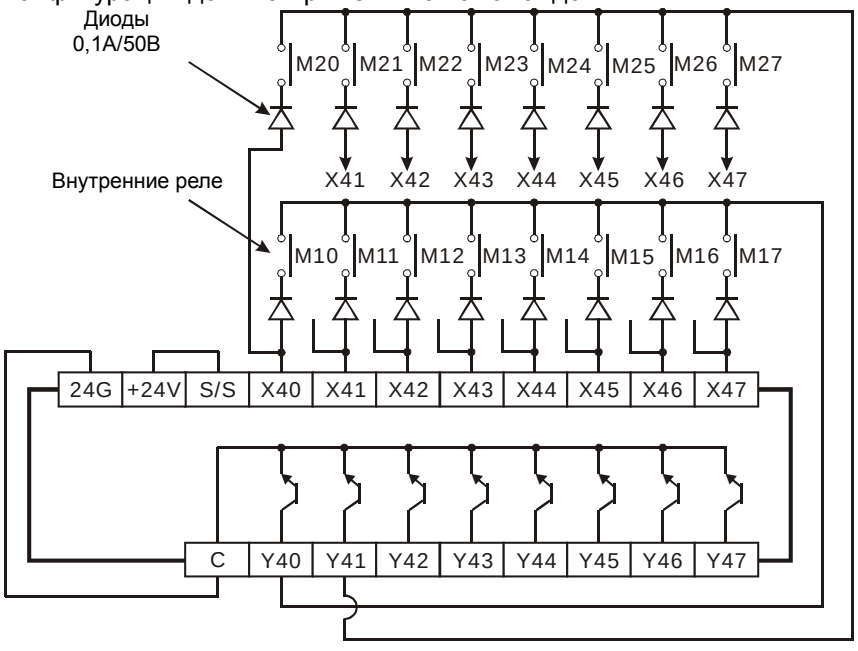
Операнды:
S: Адреса входов **D_1**: Адреса выходов **D_2**: Адреса матричной обработки входных сигналов **n**: Количество рядов в матрице

Описание:
1. Чтение в ПЛК матрицы размером 8 x n
2. S являются источником матрицы входов и занимает 8 позиций
3. D_1 является триггером (транзисторный выход Y) для чтения входных сигналов и занимает n последовательных точек.
4. D_2 является начальным адресом матрицы, где сохраняется результат чтения входов.
5. Переключатель 8 x n-матрицы считывается по методу умножения по 8-ми входам и
6. n выходам. Результат операции отображается в матричной таблице, начиная с D2. Набор из 8 входных сигналов сгруппирован в строку, число строк, соответственно, равно n. Каждая выбранная для чтения строка срабатывает на выходах, начиная с D1. Результат сохраняется в матричной таблице, начиная с адреса D2.
7. Максимально возможны 64 входа - 8 массивов (n=8), т.е. 8x8=64
8. Время обработки каждого массива около 25 мс, т.е. время обработки матрицы не более 200 мс.
9. Рекомендуется использовать специальные вспомогательные реле M1000 (H/O контакт).
10. Специальное реле M1029 включится, как только матрица будет обработана. M1029 выключится при выключении входных сигналов.
11. Флаг: M1029, флаг завершения выполнения.

Пример программы:

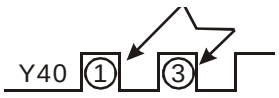


Когда ПЛК включится, выполняется команда MTR. Состояние входовX40-X47 будет считано 2 раза, т.е. по двум выходам (рядам) Y40, Y41, т.е. будут сгенерированы 16 сигналов, которые будут записаны во внутренние реле M10-M17 и M20-M27.
На рисунке ниже показана схема 2-х строчной матрицы, входной контур построен на X40-X47 и Y40-Y41. 16 коммутаторов соответствуют внутренним реле M10-M17 и M20-M27.
Конфигурация должна применяться с командой MTR.

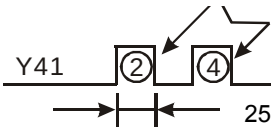


Когда выход Y40 включен, будут читаться только входы первой строки. Результат хранится во вспомогательном реле M10-M17. После выключения Y40 включается Y41. В этот раз будут считаны только входы второй строки. Результаты сохраняются в M20-M27.

Чтение состояния контактов 1-й строки



Чтение состояния контактов 2-й строки



Время обработки каждой строки прибл. 25 мс

Заметки:

1. Операнд **S** должен быть кратным 10, т.е. 00, 10, 20, соответственно X0, X10… и т.д. и занимает 8 адресов.
2. Операнд **D_1** должен быть кратным 10, т.е. 00, 10, 20, соответственно Y0, Y10… и т.д. и занимает n адресов.
3. Операнд **D_2** должен быть кратным 10, т.е. 00, 10, 20, соответственно M0, M10, S0, S10… и т.д.
4. Диапазон **n** = 2~8

API	Команда			Операнды	Функция	Контроллеры			
53	D	HSCS		(S1) (S2) (D)	Включение выхода от высокоскоростного счетчика	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DHSCS: 13 шагов
S_1					*	*	*	*	*	*	*	*	*	*		
S_2												*				
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

S_1: Сравниваемое значение **S_2**: Номер скоростного счетчика **D**: Результат

Описание:

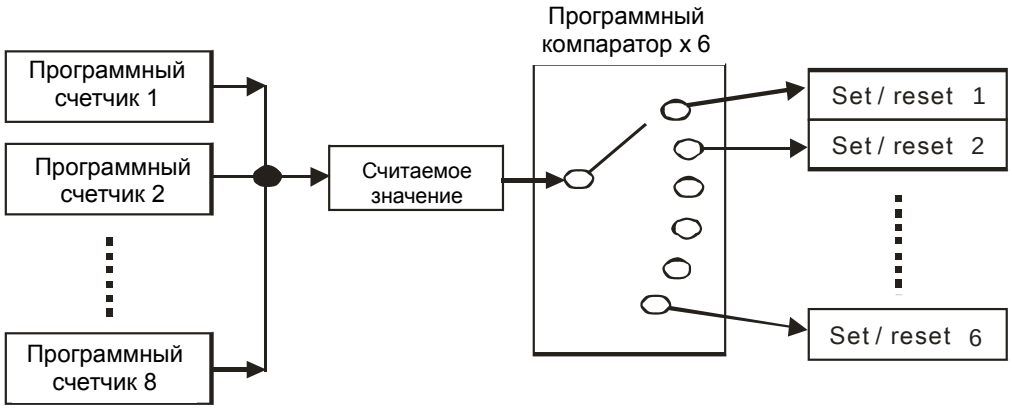
1. Функции, связанные с высокоскоростными счетчиками, используют прерывания, поэтому состояние устройства, заданного в D, обновится сразу. Эта команда сравнивает текущее значение счетчика S2 со значением в S1. Когда текущее значение счетчика S2 достигнет значения S1, то устройство в операнде D будет включено и останеться включенным после расхождения S1 и Ы2.
2. Если D задано как Y0-Y3 и во время работы значение счетчика стало равно S1, то результат сравнения сразу будет выводиться на внешние выходы Y0-Y3, в то время как другие выходы Y будут по-прежнему обновляться до конца программы. Кроме того, значения M и S не влияя на время цикла программы, будут немедленно обновлены, как и устройства Y, указанные в данной команде.
3. Диапазон операнда D: I0□0, □=1~8
4. Высокоскоростные счетчики могут быть программными или аппаратными, а также есть два типа компаратора - аппаратный и программный. Для подробной информации по высокоскоростным счетчикам см. раздел 2.12 настоящего руководства.
5. Пояснения по программным высокоскоростным компараторам для команд DHSCS/DHSCR:
   - Есть 6 программных компараторов для высокоскоростного сравнения с операцией включения/сброса (Set/Reset)
   - Эти 6 программных компараторов используются с соответствующими прерываниями высокоскоростного счетчика. Номера используемых прерываний должны быть правильно указаны перед связанными подпрограммами прерывания.

- Суммартное количество компараторов Set/Reset для обоих команд DHSCS и DHSCR не может быть более 6, в противном случае определяется синтаксическая ошибка.
- Таблица параметров программных счетчиков и компараторов:

Счетчик	C232	C233	C234	C235	C236	C237
DHSCS Высокоскоростное прерывание	I010	I050	I070	I010	I020	I030
Высокоскоростное сравнение Set/Reset (включение/сброс)	C232~C242 6 программных компараторов					

Счетчик	C238	C239	C240	C241	C242
DHSCS Высокоскоростное прерывание	I040	I050	I060	I070	I080
Высокоскоростное сравнение Set/Reset (включение/сброс)	C232~C242 6 программных компараторов				

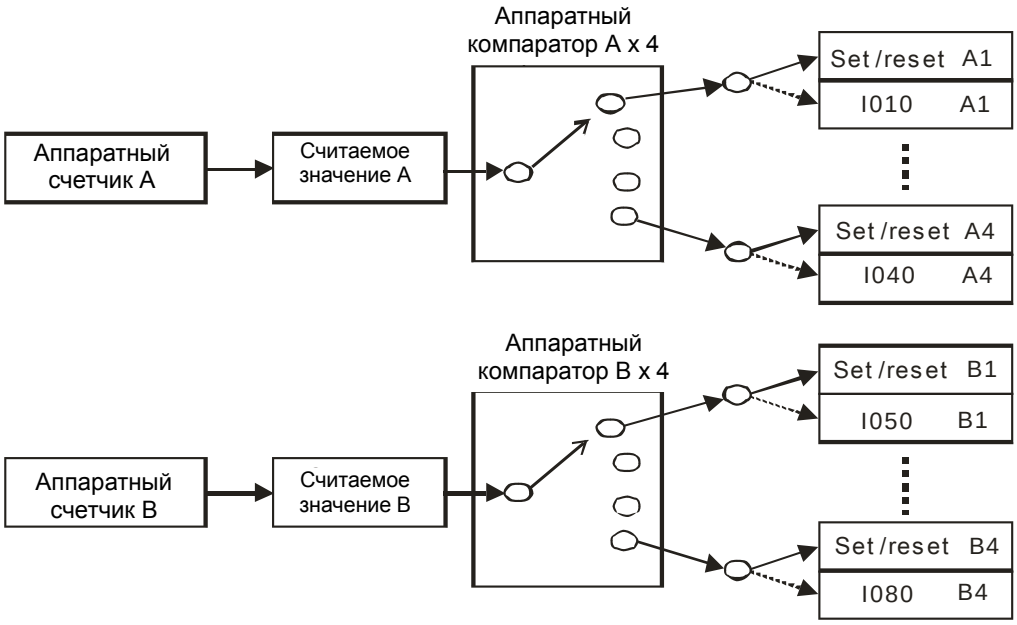
- DVP-SS2 не поддерживает программный высокоскоростной счетчик C232.
- Блок-схема программных счетчиков и компараторов:



6. Пояснения по аппаратным компараторам команд DHSCS/DHSCR:
- Есть 2 группы аппаратных компараторов, соответствующим 2 группам аппаратных счетчиков (группы A и B). В каждой группе используется 4 компаратора с индивидуальной функцией включения/сброса (Set/Reset).
- Суммартное количество компараторов Set/Reset для обоих команд DHSCS и DHSCR не может быть более 4, в противном случае определяется синтаксическая ошибка.
- Каждый высокоскоростной счетчик связан с аппаратным компаратором, следовательно, номер прерывания не может повторяться. Кроме того, I010-I040 могут быть применены только для группы A, а I050-I080 только для группы B.
- Если команда DCNT определяет C243 как высокоскоростной счетчик (группа A) и при этом команда DHSC определяет C245 как высокоскоростной счетчик (группа A), ПЛК автоматически использует C243 как счетчик и синтаксической ошибки не возникает.
- Таблица параметров аппаратных счетчиков и компараторов:

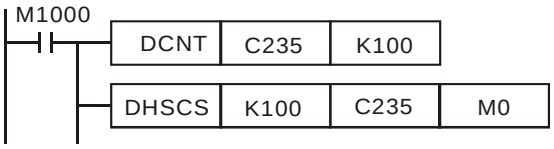
Аппаратный счетчик	Группа A				Группа B			
	A1	A2	A3	A4	B1	B2	B3	B4
Номер счетчика	C243, C245~C248, C251,C252				C244, C249, C250, C253, C254			
Высокоскоростное прерывание	I010	I020	I030	I040	I050	I060	I070	I080
Высокоскоростное сравнение Set/Reset (включение/сброс)	4 аппаратных компаратора в группе				4 аппаратных компаратора в группе			

- Блок-схема аппаратных счетчиков и компараторов:

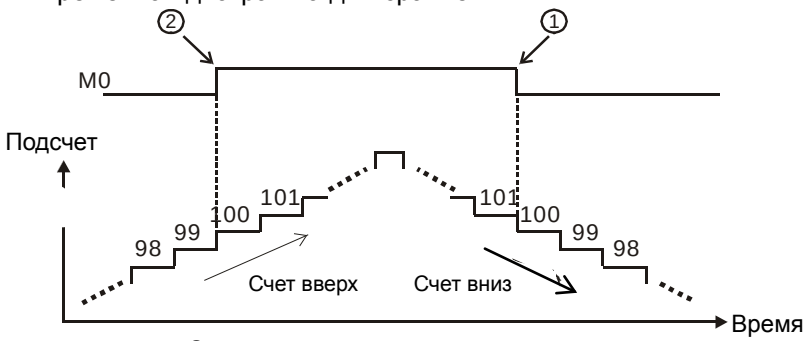


7. Различия между программными и аппаратными компараторами:
   - 6 компараторов доступны для программных счетчиков и 8 для 2 групп аппаратных счетчиков (по 4 в каждой группе)
   - Синхронизация выхода программного компаратора → значение счетчика равно сравнительному значению режимов "подсчета" и "отсчета"
   - Синхронизация выхода аппаратных компараторов → значение счетчика равно сравнительному значению +1 в режиме положительного счета и сравнительному значению -1 в режиме отрицательного.

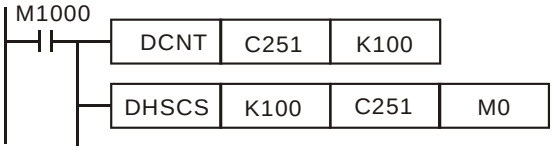
Пример программы 1:
Включение/сброс M0 при применении программного компаратора



- Если значение C235 в пределах 99 -100, команда DHSCS включает M0 (M1235 выключен, C235 считает вверх).
- Если значение C235 в пределах 101 - 100, команда DHSCR сбрасывает M0. (M1235 включен, C235 считает вниз)
- Временная диаграмма для сравнения:

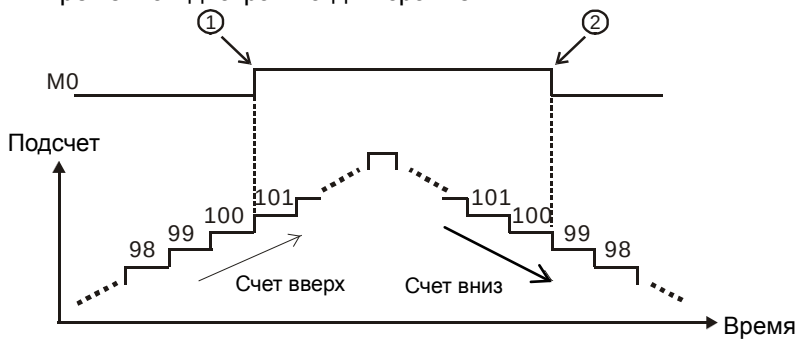


Пример программы 2:
Включение/сброс M0 при применении аппаратного компаратора

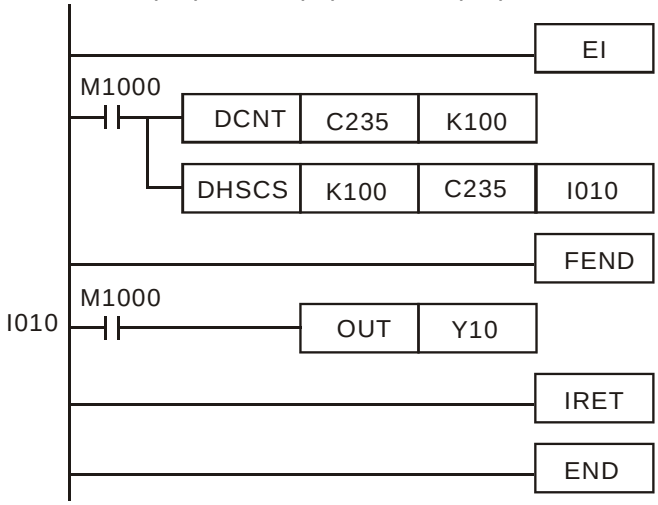


- Когда C251 считает вверх и значение C251 в пределах 100 -101, команда DHSCS включает M0.
- Когда C251 считает вниз и значение C251 в пределах 100 - 99, команда DHSCR сбрасывает M0.

- Временная диаграмма для сравнения:



Пример программы 3:
Выполнение подпрограммы прерывания программным компаратором.



- Когда значение C235 в пределах 99 -100, Подпрограмма прерывания вызывает I010 и включает Y0.

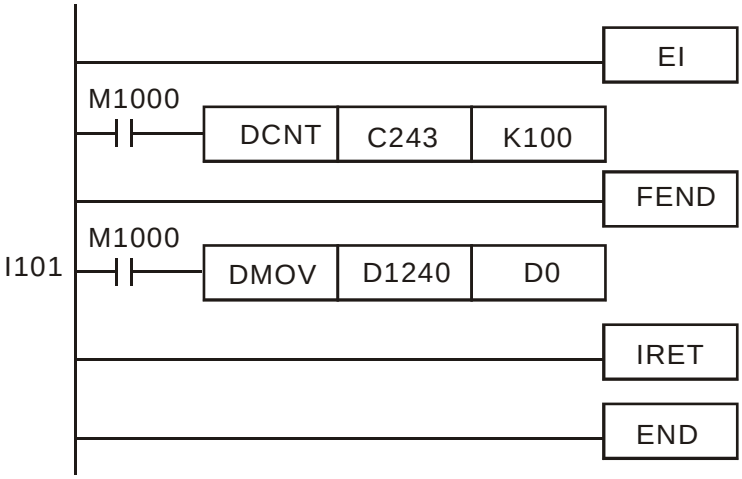
Заметки:

- Если операнд D задан как S, M или Y0-Y3, то результат сравнения будет сразу выводиться на внешние выходы Y0-Y3 (Y0-Y5 для SS2/SX2). Однако если D задан как Y4-Y337, внешние выходы будут обновлены до конца программы (задержка на один цикл сканирования).

8. Значения функций хранения высокоскоростного прерывания:
   - Когда X1, X3, X4 и X5 применяются для функции сброса и связанные с ними внешние прерывания отключены, пользователь может задать функцию сброса по переднему и заднему фронту с помощью специальных реле M, указанными в таблице программных высокоскоростных счетчиков, однако, если применяются внешние прерывания, то они имеют больший приоритет. Кроме того, ПЛК будет передавать текущие данные в счетчиках в связанные регистры (см. ниже), а затем обнулять счетчики.
   - Когда X0 (вход счетчика) и X1 (внешнее устройство прерывания I100/I101) работают с C243, значение счетчика будет перемещено в D1240 и D1241 в момент включения прерывания, а затем счетчик будет обнулен.
   - Когда X2 (вход счетчика) и X3 (внешнее устройство прерывания I300/I301) работают с C244, значение счетчика будет перемещено в D1242 и D1243 в момент прерывания, а затем счетчик будет обнулен.
   - Когда X0 (вход счетчика) и X4 (внешнее устройство прерывания I100/I101) работают с C246, значение счетчика будет перемещено в D1240 и D1241 в момент прерывания, а затем счетчик будет обнулен.
   - Когда X2 (вход счетчика) и X5 (внешнее устройство прерывания I300/I301) работают с C244, C250, C254 значение счетчика будет перемещено в D1242 и D1243 в момент прерывания, а затем счетчик будет обнулен.

D	D1241, D1240				D1243, D1242		
Счетчик	C243	C246	C248	C252	C244	C250	C254
Прерыватель	X1(I100/I101)	X4(I400/I401)			X3(I300/I301)	X5(I500/I501)	

Пример программы 4:



Если прерывание I101 включается от входа X1 во время работы счетчика C243, подпрограмма прерывания выполняется немедленно и значение счетчика C243 будет перемещено в D0. После этого C243 сбросится.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
54	D	HSCR	S_1 S_2 D	Выключение от высокоскоростного счетчика				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSCR: 13 шагов
S_2												*				
D		*	*	*								*				

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Сравнительное значение **S_2**: Номер скоростного счетчика **D**: Результат

Описание:

1. Команда сравнивает текущее значение S2 со значением S1. Когда эти значения становятся равны, D выключается. После этого даже в случае расхождения значений S1 и S2 D будет оставаться выключенным.
2. Если D задано как Y0-Y3, результат сравнения будет сразу выводиться на внешние выходы Y0-Y3 (назначенные значения Y0-Y3 будут сброшены). Тем не менее, другие выходы Y будут обновляться до конца программы (задержка на один цикл сканирования). Кроме того, M и S, не влияя на время цикла программы, также будут обновлены.
3. Операнд D может быть задан с высокоскоростными счетчиками C232-C254 (SS2 не поддерживает C232) так же как и S2.
4. Высокоскоростные счетчики могут быть программными или аппаратными, а также есть два типа компаратора - аппаратный и программный. Для подробной информации по высокоскоростным счетчикам см. раздел 2.12 настоящего руководства.
5. Для ознакомления с программными и аппаратными счетчиками обратитесь к API53.
6. Для ознакомления с примерами программ см. примеры программ 1 и 2 в API53.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
55	D	HSZ	S_1 S_2 S D	Зонное сравнение для высокоскоростного счетчика				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		DHSZ: 17 шагов
S_2					*	*	*	*	*	*	*	*	*	*		
S												*				
D		*	*	*												

PULSE				16-bit				32-bit			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

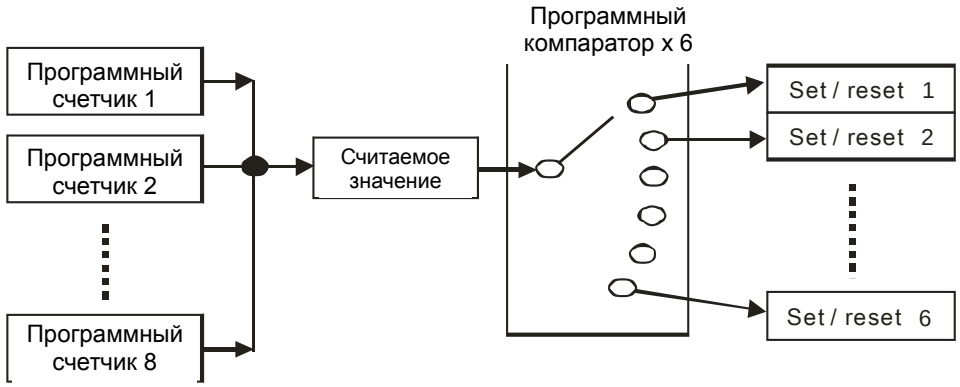
Операнды:
S_1: Нижняя граница зоны сравнения **S_2**: Верхняя граница зоны сравнения **S**: Номер скоростного счетчика **D**: Результат (3 последовательных адреса)

Описание:

1. Сравнение текущего значения высокоскоростного счетчика с областью, ограниченной значениями, указанными в (**S_1**/**S_2**). Результат сравнения представляется в операндах (D), (D+1), (D+2). (D) = 1, если S < S1; (D+1) = 1, если S1 ≤ S ≤ S2; (D+2) = 1, если S > S2.
2. **S_1** должно быть меньше или равно **S_2** (**$S_1 \leqq S_2$**).
3. Если D определяется как Y0-Y3, результат сравнения будет сразу выводиться на внешние выходы Y0-Y3, в то время как другие выходы Y будут по-прежнему обновляться до конца программы. Кроме того, M и S, не влияя на время цикла программы, также будут обновлены.
4. Высокоскоростные счетчики включают в себя программные или аппаратные решения. Есть два типа сравнения - для обоих типов. Для подробной информации по высокоскоростным счетчикам см. раздел 2.9 настоящего руководства.
5. Пояснения по программным компараторам и счетчикам для команды DHSZ:
   - **Таблица программных счетчиков и компараторов:**

Счетчик	C232	C233	C234	C235	C236	C237	C238	C239	C240	C241	C242
Высокоскоростное сравнение Set/Reset (включение/сброс)	6 программных компараторов										

   - Блок-схема программного счетчика и компаратора:



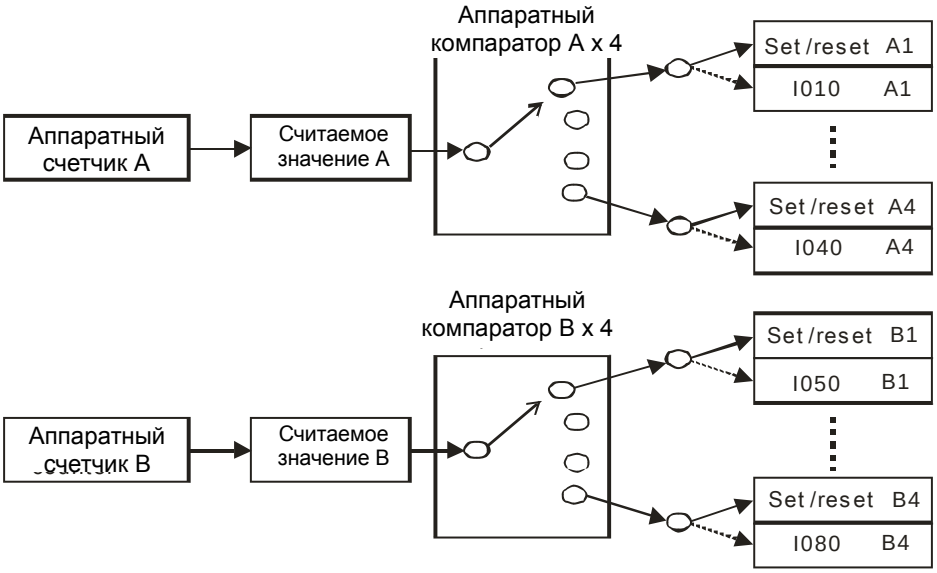
   - Только 6 программных зонных компаратора доступны для данной операции, поэтому они не включают сравнение DHSCS и DHSCR.
   - SS2 не поддерживает счетчик C232.

6. Пояснения по аппаратным компараторам и счетчикам для команды HSZ:

Таблица аппаратных счетчиков и компараторов:

Аппаратный счетчик	Группа A				Группа B			
	A1	A2	A3	A4	B1	B2	B3	B4
Номер счетчика	C243, C245~C248, C251,C252				C244, C249, C250, C253, C254			
Высокоскоростное сравнение Set/Reset (включение/сброс)	4 аппаратных компаратора в группе A				4 аппаратных компаратора в группе B			

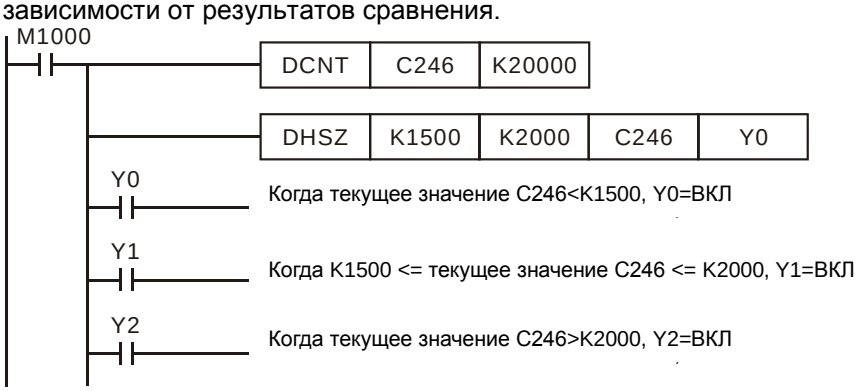
Блок-схема аппаратных счетчиков и компараторов:



- Две группы могут быть использованы только один раз для каждой пары компараторов. Например, если команда DHSZ использует A3 и A4, то командам DHSSCS, DHSCR доступны только компараторы A1 и A2.
- Когда DHSCS использует I030 или I040, компараторы A3 и A4 больше недоступны для команды DHSZ. А когда DHSCS использует I070 или I080, компараторы B3 и B4 не доступны для команды DHSZ. Если компараторы используются повторно, будет обнаружена синтаксическая ошибка.

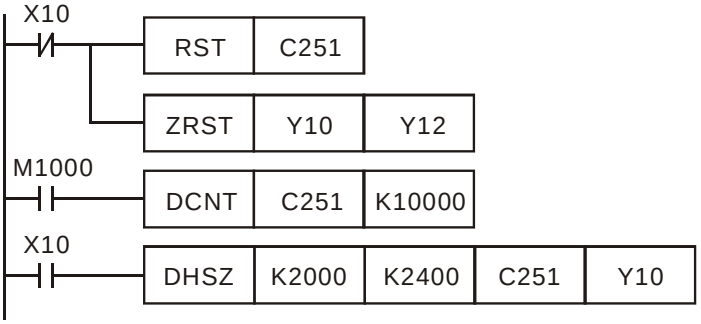
Пример программы 1: Применение высокоскоростного счетчика

1. Когда D определяется как Y0, то Y0-Y2 будут заняты автоматически.
2. Когда выполняется DHSZ, происходит сравнение текущего значения C246 с верхней/нижней границами (1500/2000) зоны сравнения и Y0-Y2 включаются в зависимости от результатов сравнения.



Пример программы 2: Применение команды DHSZ для управления тремя скоростями

1. C251 является двухфазным высокоскоростным счетчиком. Когда X10 включен, команда сравнивает текущее значение с K2000, если текущее значение меньше или равно K2000, Y10 включается.
2. Когда X10 выключен, Y10~Y12 сбрасываются.



(S1): окончание быстрого хода (пуск медленного хода)
(S2): конец медленного хода (введение торможения)
(S): определение высокоскоростного счетчика (D):
Y10-> быстрый ход
Y11 -> медленный ход
Y12 -> торможение

Временная диаграмма



API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
56	SPD	(S1) (S2) (D)	Определение скорости				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SPD: 7 шагов
S_1	*															
S_2					*	*	*	*	*	*	*	*	*	*	*	
D											*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Внешний импульсный вход **S_2**: Время подсчета импульсов (мс) **D**: Результат (5 последовательных адресов)

Описание:

1. Команда подсчитывает количество импульсов S1 за время S2, и результат записывается в D.
2. ES2/EX2 до V0.92. Внешние импульсные входы **S_1**:

Доступные входы	X0, X2	X1 (X0/X1)	X6, X7
Входной режим	1-фазный	2-фазный	1-фазный
Макс. частота	100 кГц	5 кГц	10 кГц

3. ES2/EX2 V1.00 или выше. Внешние импульсные входы **S_1**:

Доступные входы	X0, X2	X1 (X0/X1), X3 (X2/X3) X5 (X4/X5), X7 (X6/X7)	X4, X6
Входной режим	1-фазный	2-фазный	1-фазный
Макс. частота	100 кГц	5 кГц	10 кГц

4. SS2/SA2/SE/SX2. Внешние импульсные входы **S_1**:

Доступные входы	X0, X2	X1 (X0/X1), X3 (X2/X3) X5 (X4/X5), X7 (X6/X7)	X4, X6
Входной режим	1-фазный	2-фазный	1-фазный
Макс. частота	SA2/SE/SX2: 100 кГц SS2: 20 кГц	5 кГц. X1(X0/X1) в SA2/SE: 50 кГц	10 кГц

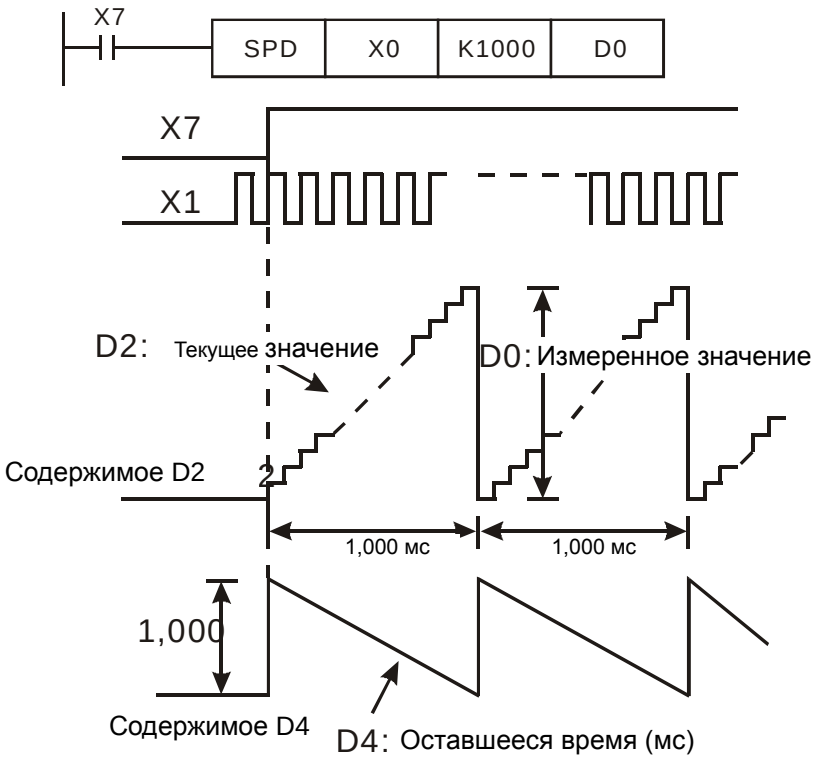
5. **D** занимает 5 последовательных регистров, **D** + 1 и **D** сумма импульсов после отсчета времени; **D** +3 и **D** + 2 текущее значение импульсов внутри интервала времени; **D** + 4 остающееся отсчитываемое время (макс. 32767 мс).
6. Если X0, X1, X2, X6 или X7 используются командой SPD, связанные с ними высокоскоростные датчики и внешние прерыватели I000/I001, I100/I101, I200/I201, I600/I601 или I700/I701 не могут использоваться.
7. Для ES2/EX2 до V0.92: когда X0, X2, X6 и X7 используются, они будут определены как 1-фазный вход. Когда X1 используется, X0(A) и X1(B) будут определены как 2-фазный вход.
8. Для SS2/SA2/SE/SX2 и ES2/EX2 V1.00 или выше: когда X0, X2, X4 и X6 используются, они будут определены как 1-фазный вход. Когда X1, X3, x5, X7 используются, X0, X2, X4, X6 будут определены как 2-фазный вход.
9. Эта команда используется для получения скорости вращения и сохраняется в D как пропорция скорости вращения. Скорость вращения N можно рассчитать по следующей формуле:

$$N=\frac{60(D0)}{nt}\times 10^3(rpm)$$

N: Скорость вращения
n: Число импульсов на оборот
t: Время дискретизации **S_2** (мс)

Пример программы:

1. Когда X7 включен, D2 считает количество включений X0 в течение 1000мс, после чего выключается. Результат сохраняется в D0, D1.
2. Когда 1000 мс закончатся, D2 сбрасывается. При новом включении X7, D2 начинает отсчет снова.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
57	D	PLSY		S_1 S_2 D	Импульсный выход				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PLSY: 7 шагов DPLSY: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Частота импульсного выхода (Y0 ~ Y3 доступные) **S_2**: Количество выходных импульсов **D**: Адрес выхода

Описание:

1. Когда выполняется команда PLSY, заданное количество импульсов S2 будет подаваться на импульсный выход D с указанной в S1 частотой.
2. Таблица частот **S_1**

Диапазон выходных частот MPU			
Диапазон	Выход	Y0, Y2	Y1, Y3
	16-битная команда	SS2: 0~10,000Гц ES2/EX2/SA2/SE/SX2: 0~32,767 Гц	0~10,000Гц
	32-битная команда	SS2: 0~10,000Гц ES2/EX2/SA2/SE/SX2: 0~100,000 Гц	0~10,000Гц
Если частота равна 0 или меньше указанной, импульсный выход будет отключен. Если частота больше указанной, ПЛК будет выводить максимальную указанную частоту.			

3. **S_2** определяет количество выходных импульсов.
   16-битная команда: -32,768~32,767. 32-битная команда: -2,147,483,648~2,147,483,647.
4. Когда **S_2** определяется как K0, создается неограниченный ряд импульсов, независимо от предельного числа импульсов.
5. Когда D1220/D1221 = K1 или K2, положительный/отрицательный знак **S_2** обозначает направление импульсного выхода.

6. Четыре режима импульсных выходов:

Режим / Выход	D1220						D1221					
	K0		K1	K2	K3		K0		K1	K2	K3#	
Y0	Pulse		Pulse	A	CW							
Y1		Pulse	Dir	B		Pulse						
Y2							Pulse		Pulse	A	CCW	
Y3								Pulse	Dir	B		Pulse

Pulse: Импульсный A: A – фазный импульс CW: По часовой стрелке
Dir: Направленный B: B –фазный импульс CCW: Против часовой стрелки

Замечание: Когда D1220 определяется как K3, D1221 не существует.

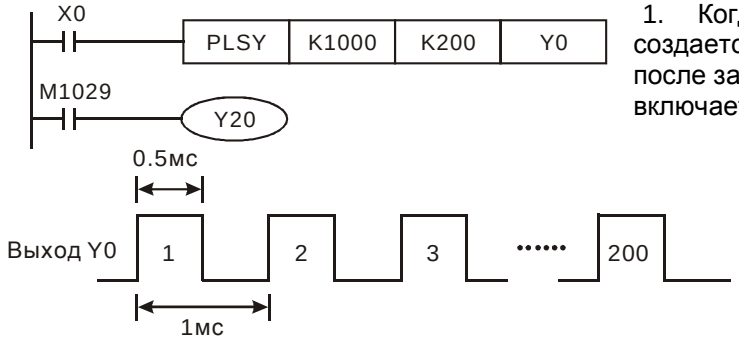
7. Флаги импульсных выходов:

Выход	Y0	Y1	Y2	Y3
Флаг завершения	M1029	M1030	M1102	M1103
Быстрая пауза	M1078	M1079	M1104	M1105
Выход 0.01~100 Гц	M1190	M1191	M1192	M1193

a. M1029 включается после завершения выполнения команды для Y0/Y1 (D1220=K1, pulse/Dir).
M1102 включается после завершения выполнения команды для Y2/Y3 (D1221=K1, pulse/Dir).
M1029 включается после завершения выполнения команды для Y0/Y2 (D1220 = K3, CW/CCW).
b. Флаги завершения M1029, M1030, M1102 и M1103 следует сбросить вручную после завершения выполнения команды.
c. Когда команды PLSY / DPLSY выключены, флаги завершения сброшены.
d. Когда M1190~M1192 включены, выходной диапазон для PLSY Y0~Y3 составляет 0.01~100 Гц.

8. При выполнении команды PLSY изменение параметра S2 игнорируется. Для изменения количества импульсов на выходе необходимо остановить выполнение команды, а затем изменить число импульсов.
9. Частота (данные в S1) может быть изменена во время выполнения данной команды.
10. Отношение времени включения и выключения импульсного выхода составляет 1:1.
11. Если операнды S1, S2 используются с индексом F, возможно выполнение только 16-битных команд.
12. Программа позволяет одновременно выполнять только 4 команды (PLSY, PWM, PLSR). Если выход Y1 используется одновременно несколькими командами, ПЛК будет обрабатывать их по порядку.

Пример программы:



1. Когда X0 включен, на выходе Y0 создается 200 импульсов частотой 1 кГц, после завершения обработки выхода включается M1029 для установки Y20.
2. Когда X0 выключен, импульсный выход Y0 выключается. При новом включении X0 импульсный выход включается заново.

Заметки:

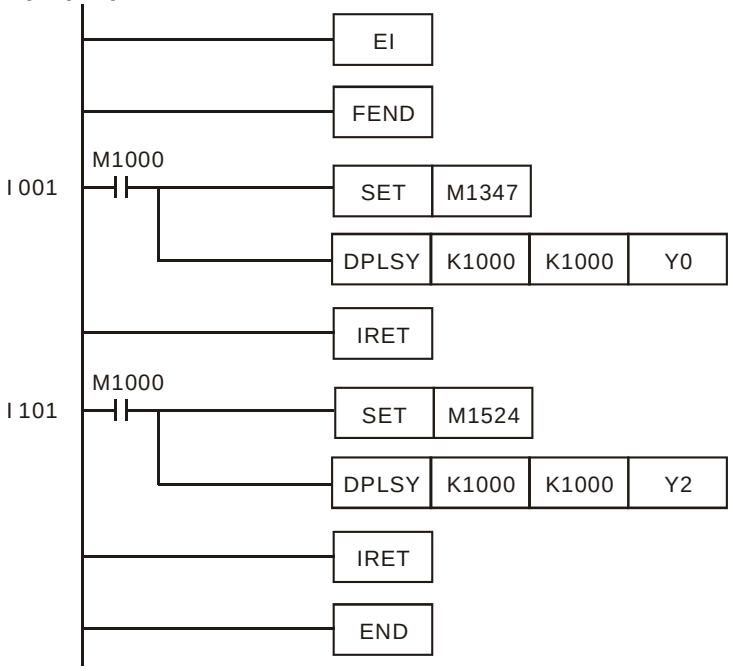
1. Связанные флаги (специальные реле):
   - M1029: M1029 включен, когда команда PLSY или PLSR для импульсного выхода Y0 полностью выполнена.
   - M1030: M1030 включен, когда команда PLSY или PLSR для импульсного выхода Y1 полностью выполнена.
   - M1102: M1102 включен, когда команда PLSY или PLSR для импульсного выхода Y2 полностью выполнена.
   - M1103: M1103 включен, когда команда PLSY или PLSR для импульсного выхода Y3 полностью выполнена.
   - M1078: Остановка выполнения команды на Y0 (без задержки)
   - M1079: Остановка выполнения команды на Y1 (без задержки)
   - M1104: Остановка выполнения команды на Y2 (без задержки)
   - M1105: Остановка выполнения команды на Y3 (без задержки)

M1190 Установка высокоскоростного выхода Y0 на 0.01~100Hz
M1191 Установка высокоскоростного выхода Y1 на 0.01~100Hz
M1192 Установка высокоскоростного выхода Y2 на 0.01~100Hz
M1193 Установка высокоскоростного выхода Y3 на 0.01~100Hz
M1347: Автоматический сброс флага прерывания для Y0 по завершении выполнения команды
M1348: Автоматический сброс флага прерывания для Y1 по завершении выполнения команды
M1524: Автоматический сброс флага прерывания для Y2 по завершении выполнения команды
M1525: Автоматический сброс флага прерывания для Y3 по завершении выполнения команды
M1538: Индикация остановки для Y0
M1539: Индикация остановки для Y1
M1540: Индикация остановки для Y2
M1541: Индикация остановки для Y3

2. Связанные (специальные) регистры D:
D1030: Выданное количество импульсов на выходе Y0 (младшее слово).
D1031: Выданное количество импульсов на выходе Y0 (старшее слово).
D1032: Выданное количество импульсов на выходе Y1 (младшее слово).
D1033: Выданное количество импульсов на выходе Y1 (старшее слово).
D1336: Выданное количество импульсов на выходе Y2 (младшее слово).
D1337: Выданное количество импульсов на выходе Y2 (старшее слово).
D1338: Выданное количество импульсов на выходе Y3 (младшее слово).
D1339: Выданное количество импульсов на выходе Y3 (старшее слово).
D1220: Фаза 1-й группы импульсных выходов (Y0,Y1), см. описание команды.
D1221: Фаза 2-й группы импульсных выходов (Y2,Y3), см. описание команды.

3. Дополнительные пояснения для M1347,M1348, M1524, M1525:
Обычно, когда количество импульсов отработано полностью, команда PLSY должна быть остановлена, после чего начнется новый цикл ее выполнения. Когда M1347, 1348 или 1525 включены, связанные с ними выходы Y0-Y3 будут автоматически сброшены по завершении работы команды. ПЛК сканирует 4 флага после выполнения команды END, следовательно, команде PLSY в режиме непрерывных импульсов необходимо время задержки в 1 цикл для новой обработки импульсного выхода. Эта функция в основном используется в подпрограммах или процессе прерывания, когда требуется высокая скорость работы импульсного выхода:

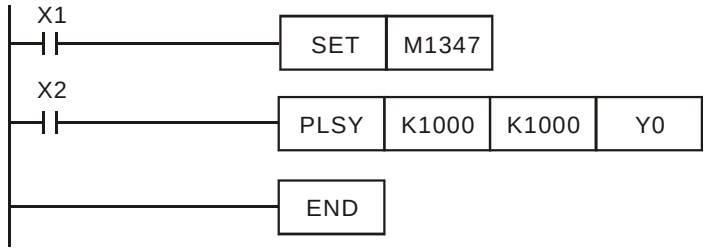
Пример программы 1:



Описание: Всякий раз при срабатывании I001 Y0 выдает 1000 импульсов, а при срабатывании I101 Y2 выдает 1000 импульсов. После завершения действия команды необходим интервал не менее одного цикла для запуска следующей операции обработки

импульсного выхода.

Пример программы 2:



Описание:
Когда включены оба X1 и X2, импульсный выход Y0 будет работать в режиме непрерывных импульсов. Тем не менее, задержка в 1 цикл будет осуществляться каждые 1000 импульсов.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
58	PWM	(S1) (S2) (D)	Импульсный выход с ШИМ				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	PWM: 7 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S_1: Ширина импульса (t) (мс) **S_2**: Продолжительность периода (T) (мс) **D**: Импульсные выходы (Y0, Y1, Y2, Y3)

Описание:
1. Условие: **S1 ≤ S2**
2. Кроме DVP-SE:

Справочная таблица периода и ширины импульса					
Диапазон импульсного выхода (период/ширина)	Выход	Y0	Y2	Y1	Y3
	Ширина	0~1000		0~32767	
	t / T	0~1000 мс / 0~1000 мс		0~32,767 мс / 0~32767 мс	
Флаги		M1112	M1113	M1070	M1071
Флаги высокоскоростного выхода		M1116 включен (ед.изм. 1 мкс)		M1117 включен (ед.изм. 10 мкс)	

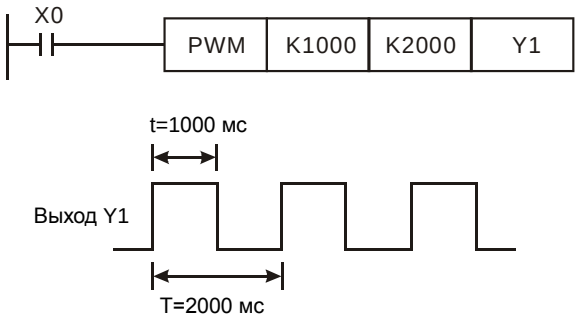
3. Только для DVP-SE:

Справочная таблица периода и ширины импульса					
Диапазон импульсного выхода (период/ширина)	Выход	Y0	Y2	Y1	Y3
	Ширина	0~1000		0~32767	
	t / T	0~100.0мс, 0~10.00мс		0~32,767мс, 0~3,276.7мс	
Флаги		M1112	M1113	M1070	M1071

4. Импульсные выходы для операнда D: Y0, Y1, Y2, Y3,
5. Когда несколько команд обработки импульсного выхода (PLSY, PWM, PLSR) используют Y1 или Y3 в качестве выходного устройства в одном и том же цикле, ПЛК выполнит первую в очереди команду.
6. Когда **S_1**<0, **S_2**<0 или **S_1** > **S_2**, возникает ошибка (M1067 и M1068 не включаются), и импульс на выход подаваться не будет. Когда **S_1** = **S_2**, на выход подается постоянный сигнал.
7. **S_1**, **S_2** могут быть изменены во время выполнения команды.
8. Когда M1112 включен, для выхода Y0 дискретность задания 10 мкс, когда M1112 выключен, дискретность 100 мкс.
9. Когда M1070 включен, для выхода Y1 дискретность задания 100 мкс, когда M1070 выключен, дискретность 1 мс.
10. Когда M1113 включен, для выхода Y2 дискретность задания 10 мкс, когда M1113 выключен, дискретность 100 мкс (неприменимо к DVP-SE).
11. Когда M1113 включен, для выхода Y2 дискретность задания 100 мкс, когда M1113 выключен, дискретность 1 мс (только для DVP-SE).
12. Когда M1071 включен, для выхода Y3 дискретность задания 100мкс, когда M1071 выключен, дискретность 1мс.
13. При включенном M1116 M1112 и M1113 не работают. Время импульса между Y0 и Y2 равно 1 мкс. Поддерживаются DVP-ES2 версии 3.00 / SS2 версии 2.80 / SA2 версии 2.60 / SE версии 2.60 / SX2 версии 2.40.

14. При включенном M1117 флаги M1070 и M1071 не работают. Время импульса между Y1 и Y3 равно 10 мкс. Поддерживаются DVP-ES2 версии 3.00 / SS2 версии 2.80 / SA2 версии 2.60 / SE версии 2.60 / SX2 версии 2.40.
15. Если M1116 для DVP-SS2 включен, минимальная ширина выходного импульса должна быть больше 20. Иначе, в связи с аппаратными ограничениями для выходов Y0 и Y2, ширина выходного импульса будет некорректной.

Пример программы:
Когда X0 = ON, на Y1 выводится импульс как показано на рисунке. Когда X0 = ВЫКЛ., выход Y1 выключен.



Заметки:
1. Флаги:
   - M1070: Дискретность задания Y1 для команды PWM (ВКЛ:100 мкс, ВЫКЛ: 1 мс)
   - M1071: Дискретность задания Y3 для команды PWM (ВКЛ:100 мкс, ВЫКЛ: 1 мс)
   - M1112 Дискретность задания Y0 для команды PWM (ВКЛ:10 мкс, ВЫКЛ: 100 мкс)
   - M1113 Дискретность задания Y2 для команды PWM (ВКЛ:10 мкс, ВЫКЛ: 100 мкс)
   - M1116 Если M1116 включен, время выходного импульса между Y0 и Y2 равно 1 мкс. M1112 и M1113 не работают.
   - M1117 Если M1117 включен, время выходного импульса между Y1 и Y3 равно 10 мкс. M1070 и M1071 не работают.
2. Специальные регистры D:
   - D1030 Выданное количество импульсов на Y0 (младшее слово)
   - D1031 Выданное количество импульсов на Y0 (старшее слово)
   - D1032: Младшее слово значения Y1
   - D1033 Старшее слово значения выхода Y1
   - D1336 Выданное количество импульсов на Y2 pulse output (младшее слово)
   - D1337 Выданное количество импульсов на Y2 pulse output (старшее слово)
   - D1338: Младшее слово значения выходаY3
   - D1339: Старшее слово значения Y3

API	Команда			Операнды	Функция
59	D	PLSR		S_1 S_2 S_3 D	Импульсный выход с ускорением/замедлением

Контроллеры			
ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
S_3					*	*	*	*	*	*	*	*	*	*	*
D		*													

Шаги программы:
PLSR: 9 шагов
DPLSR: 17 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Максимальная частота (Гц) **S_2**: Количество импульсов **S_3**: Время ускорения/замедления (мс) **D**: Импульсный выход (доступны Y0, Y1, Y2 и Y3)

Описание:
1. Команда PLSR создает на выходе заданное число импульсов с заданной частотой. Скорость изменения частоты растет (разгон) до достижения заданного значения, после чего уменьшается (замедление) также до достижения заданного значения за время S3.

2. Частотный диапазон **S1**:

Диапазон S_1:			
Выходная частота:	Выход	Y0, Y2	Y1, Y3
	16- бит	SS2: 6~10,000 Гц ES2/EX2/SA2/SE/SX2: 6~32,767 Гц	6~10,000 Гц
	32- бит	SS2: 6~10,000 Гц ES2/EX2/SA2/SE/SX2: 0~100,000 Гц	6~10,000 Гц
Если заданная частота меньше 6 Гц ПЛК будет выдавать 6 Гц. Если частота больше указанной, ПЛК будет выводить максимальную указанную частоту.			

3. Если выходами являются Y0 и Y2, то начальная/конечная частота Y0 устанавливается D1340 и начальная/конечная частота Y2 устанавливается D1352.
4. Если выходами являются Y1 и Y3, то начальная/конечная частота равна 0 Гц.
5. Когда D1220/D1221 = K1 или K2, положительный/отрицательный знак S_2 определяет направление выходного импульса.
6. Команда PLSR поддерживает 2 режима:

Режим / Выход	D1220			D1221		
	K0		K1	K0		K1
Y0	Pulse		Pulse			
Y1		Pulse	Dir			
Y2				Pulse		Pulse
Y3					Pulse	Dir

7. Когда выходы - Y0 и Y2 работают в режиме Pulse, т.е. D1220 = K0, D1221 = K0, диапазон S_2 составляет 1~32,767 (16- битовая команда) и 1~2,147,483,647 (32- битовая команда).
8. Когда выходы - Y0 и Y2 работают в режиме Pulse/Dir, т.е. D1220 = K1, D1221 = K1, диапазон S_2 составляет 1~32,767 или -1~-32,768 (16- битовая команда) и 1~2,147,483,647 или -1~-2,147,483,648 (32- битовая команда)
9. Когда выходы - Y1 и Y3, диапазон S_2 составляет 1~32,767 (16- битовая команда) и 1~2,147,483,647 (32- битовая команда).
10. S_3: Время ускорения/замедления (мс, мин. 20 мс).
    Когда выходы - Y1 и Y3, заданное время разгона/замедления должно быть одинаковым.
    Когда выходы - Y0 и Y2, и если:
    - M1348 выключен (Y0) и M1535 выключен (Y2), заданное время разгона/замедления должно быть одинаковым..
    - M1348 включен и M1535 включен, тогда S_3 определяет только время ускорения. Время замедления определяют значения D1348 (Y0) и D1349 (Y2).
11. Импульсные выходы для операнда D: Y0, Y1, Y2, Y3
12. Когда M1257 выключен, характеристика процесса ускорения/замедления Y0 и Y2 является линейной. Когда M1257 включен, характеристика процесса ускорения/замедления является S-образной. Данный процесс для Y1 и Y3 является фиксированно линейным.
13. Изменение S_1, S_2 или S_3 во время выполнения данной команды игнорируются. Выполнение команды PLSR должно быть остановлено, если требуется изменение значений S_1, S_2 или S_3.
14. Флаги состояния импульсного выхода:

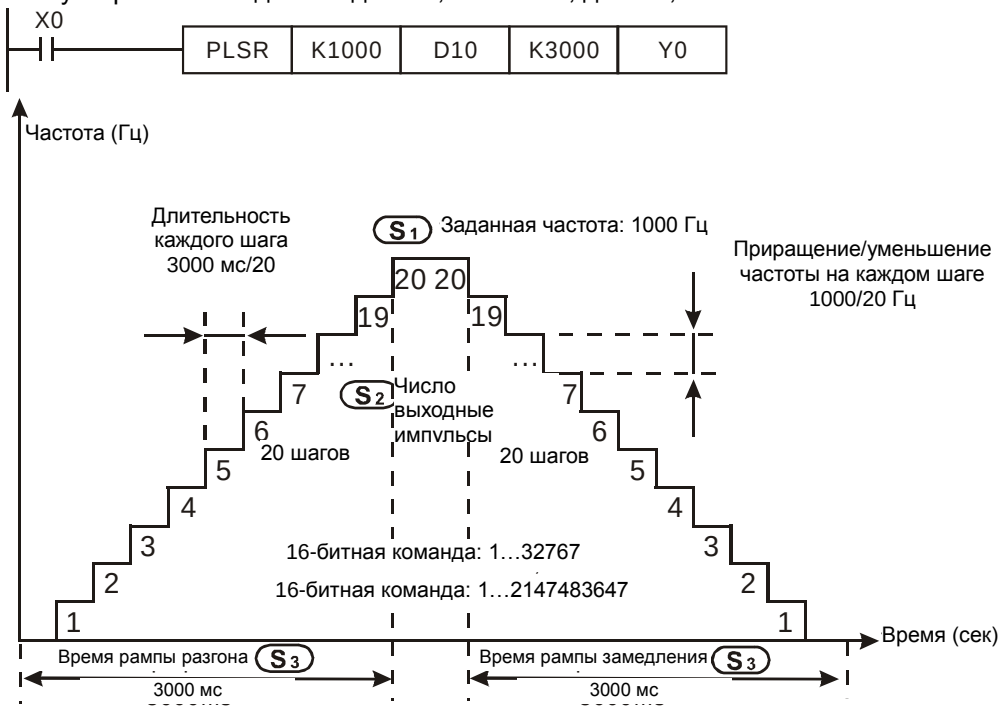
Выход	Y0	Y1	Y2	Y3
Флаг завершения	M1029	M1030	M1102	M1103
Быстрая пауза	M1078	M1079	M1104	M1105

    a. Когда заданное количество импульсов на выходе Y0/Y1 в режиме Pulse/Dir (D1220 = K1) отработано, включается флаг M1029.
    b. Когда заданное количество импульсов на выходе Y2/Y3 в режиме Pulse/Dir (D1221 = K1) отработано, включается флаг M1102.
    c. Когда команды PLSR/DPLSR запускаются снова, флаги завершения автоматически сбрасываются.
15. Во время процесса разгона количество импульсов (частота x время) на каждом шаге изменения скорости может иметь нецелое значение, но ПЛК допускает только целые числа. В этом случае, отброс десятичных значений мог бы привести к ошибке - число импульсов в каждом шаге изменения скорости будет неодинаковым. Для обеспечения корректной работы импульсного выхода ПЛК будет автоматически компенсировать эти расхождения.
16. В цикле могут быть одновременно не более 4 команд. Когда несколько команд (PLSY, PWM, PLSR) в ходе одного цикла используют один и тот же выход Y1, ПЛК выполнит эти команды в соответствии с их порядком.
17. Если установленное значение операндов выходит за рамки допустимого диапазона, они будут автоматически корректироваться.

Пример программы:

1. Когда X0 включен, команда PLSR генерирует на выходе Y0 сигнал заданной частотой 1000 Гц, количество выходных импульсов D10 и время ускорения/замедления 3000 мс. Ускорение начинает возрастать до 1000 шагами по 20 Гц за 3000 мс.
2. Когда X0 выключен, выход немедленно отключается и процесс будет запушен со значения D1030, D1031 при возобновлении выполнения команды.

Шаг ускорения/замедления для Y0, Y2: 20 мс; для Y1, Y3: 10 мс



Пояснения по применяемым флагам и регистрам:

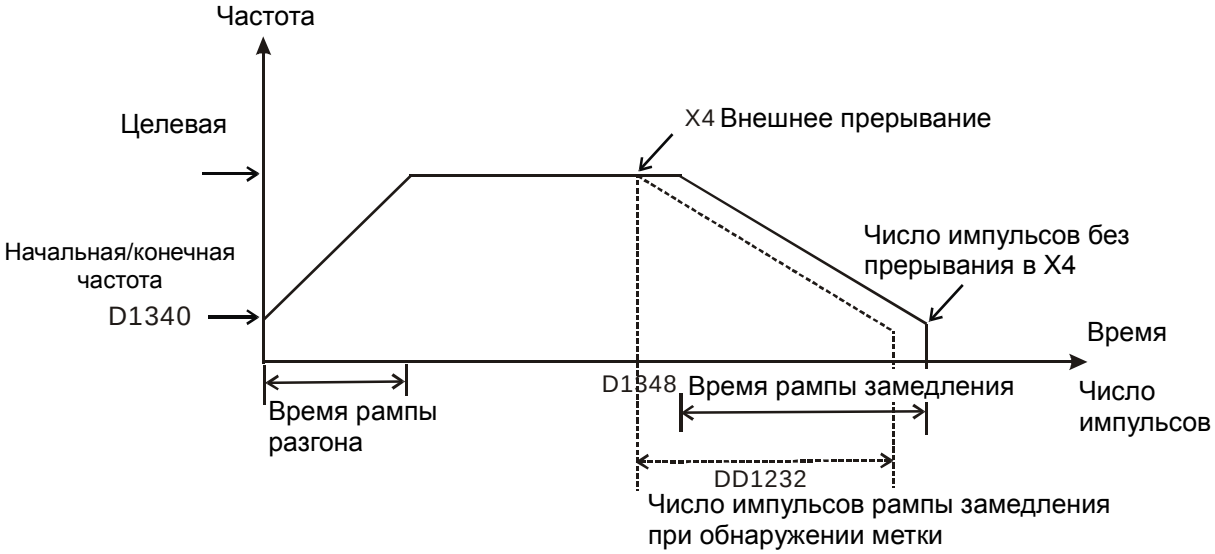
Применяемые флаги и регистры: Для M1029, M1030, M1102, M1103, M1078, M1079, M1104, M1105, M1538, M1539, M1540, M1541, M1347, M1348, M1524, M1525, см. команду PLSY.

M1108: Пауза выхода Y0 (замедление). ВКЛ = пауза, ВЫКЛ = возобновление
M1109: Пауза выхода Y1 (замедление). ВКЛ = пауза, ВЫКЛ = возобновление
M1110: Пауза выхода Y2 (замедление). ВКЛ = пауза, ВЫКЛ = возобновление
M1111: Пауза выхода Y3 (замедление). ВКЛ = пауза, ВЫКЛ = возобновление
M1156: Включение маски и функции выравнивания метки на I400/I401, относящимся к Y0
M1257: Установка ускорения/замедления Y0, Y2 по S-образной характеристике ON = включение S-образной характеристики.
M1158: Включение маски и функции выравнивания метки на I600/I601, относящимся к Y2
M1534: Включение настройки времени замедления на Y0. Используется с D1348
M1535: Включение настройки времени замедления на Y2. Используется с D1349

Специальные регистры: Для D1030~D1033, D1336~D1339, D1220, D1221, см. команду PLSY

D1026: M1156 включен, D1026 показывает число импульсов для маски Y0 (младшее слово).
D1027: M1156 вкл., D1026 показывает число импульсов для маски Y0 (старшее слово).
D1135: M1158 включен, D1135 показывает число импульсов для маски Y2 (младшее слово).
D1136: M1158 вкл., D1135 показывает число импульсов для маски Y2 (старшее слово).
D1232: Количество импульсов для замедления при получении Y0 сигнала метки (младшее слово).
D1233: Количество импульсов для замедления при получении Y0 сигнала метки (старшее слово).
D1234: Количество импульсов для замедления при получении Y2 сигнала метки (младшее слово).
D1235: Количество импульсов для замедления при получении Y2 сигнала метки (старшее слово).
D1348: M1534 включен, D1348 показывает время замедления для импульсного выхода CH0(Y0, Y1).
D1349: M1535 включен, D1349 показывает время замедления для импульсного выхода CH1(Y2, Y3).
D1340 Начальная/конечная частота импульсного выхода CH0 (Y0, Y1)
D1352 Начальная/конечная частота импульсного выхода CH1 (Y2, Y3)

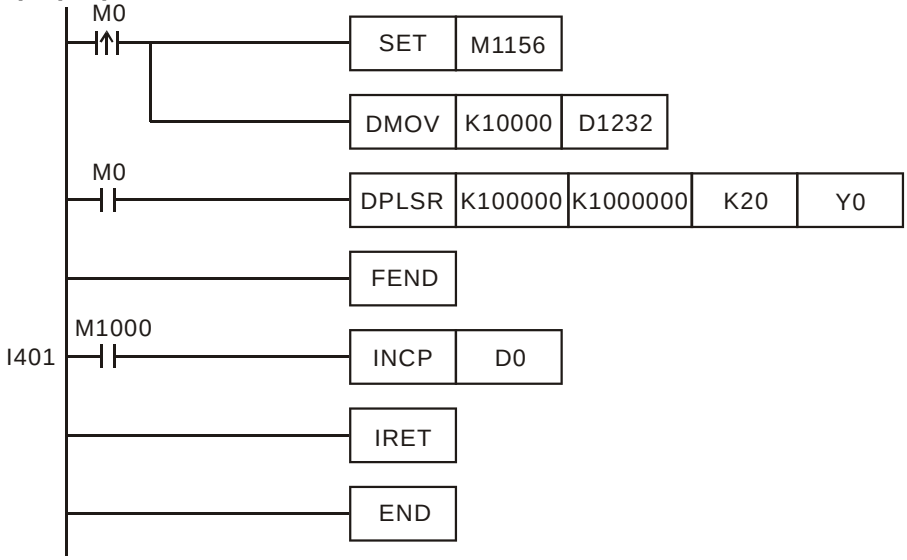
1. Работа функции метки Y0:



- Когда M1156/M1158 включены, включается пауза замедления (функция метки) на Y0/Y2 если X4/X6 получает сигнал на прерывание.
- Когда функция метки включена, время замедления не зависит от времени ускорения. Пользователь может установить время ускорения в S3 и время замедления в D1348/D1349. (Диапазон: 20мс~32767мс)
- При работающей функции метки и заданными параметрами замедления ПЛК выполнит замедление с данными параметрами после обнаружения метки. Однако, если значение DD1232/DD1234 меньше, чем указанное время замедления (D1348/D1349), ПЛК выполнит DD1232/DD1234 с указанным в них временем замедления. Кроме того, если значение DD1232/DD1234 больше половины от общего числа выходных импульсов, ПЛК изменит эти значения до значения меньше половины от общего числа выходных импульсов.
- Количество импульсов для замедления (DD1232/DD1234) имеют 32-битное значение. Установка значения K0 отключает функцию метки.
- Y0,Y2 относительные параметры маски и метки:

Параметр / Выход	Флаг метки	Входы	Время замедления	Количество выходных импульсов для маски	Количество импульсов для замедления в функции метки	Пауза (замед-ление)	Состоя-ние паузы
Y0	M1156	X4	D1348	D1026, D1027	D1232, D1233	M1108	M1538
Y2	M1158	X6	D1349	D1135, D1136	D1234, D1235	M1110	M1540

Пример программы 1:

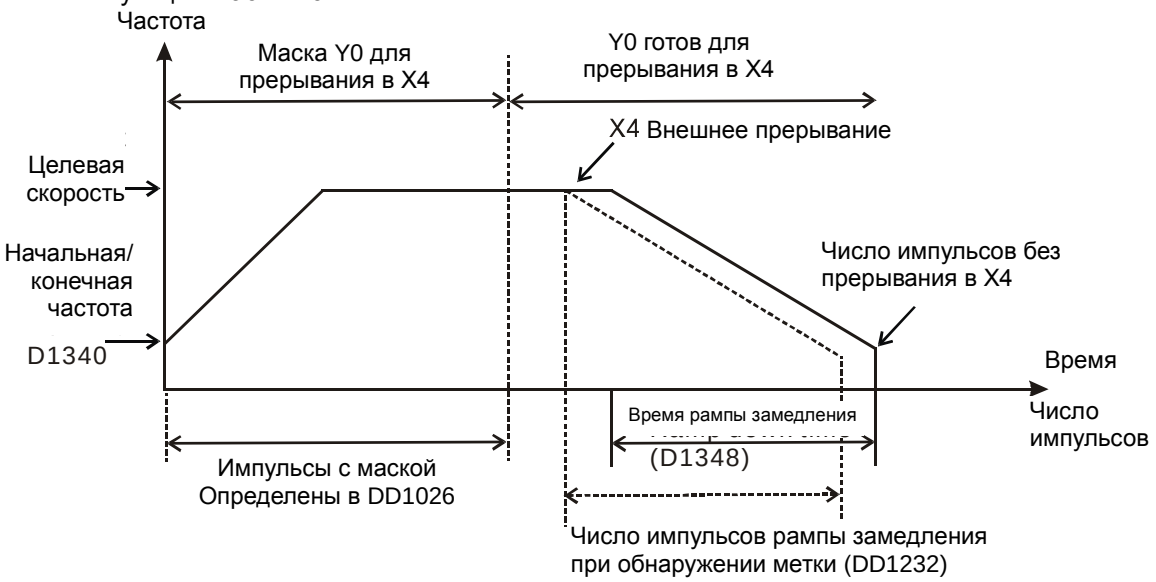


Описание:

- При срабатывании M0 Y0 становится импульсным выходом. При обнаружении внешнего прерывания на входе X4 импульсный выход будет выполнять замедление в течение 10000 импульсов, после чего прекратит подачу импульсов. M1108 будет показывать состояние паузы. Если прерывание не обнаружено, Y0 подаст 1000000 импульсов и остановится.

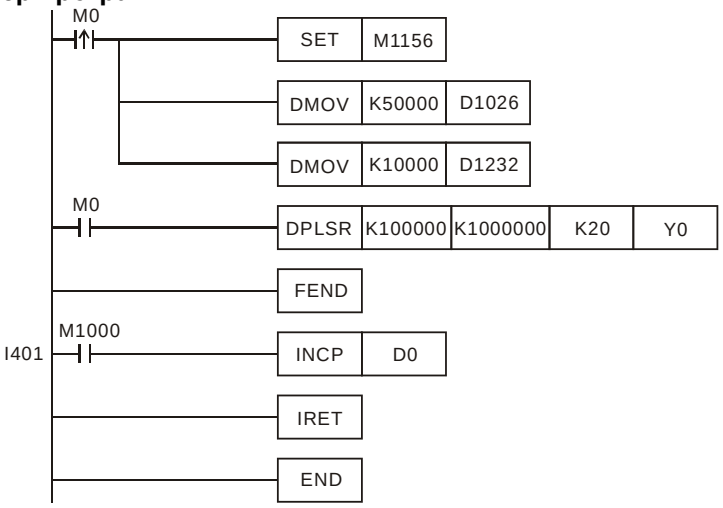
- При замедлении и останаве после обнаружения метки: M1538 будет включен и показывать паузу. Если пользователю необходимо возобновить подачу импульсов, необходимо выключить флаг M1108.

2. Функция маски Y0:



- Функция маркировки для Y0 будет включена при значениях D1026 и D1027 отличных от 0. При значениях этих операндов, равных 0 функция маски отключается. Если заданная скорость не достигнута, ПЛК очистит DD1026, чтобы отключить функцию маски. ПЛК автоматически подгоняет DD1026 под рампы разгона и торможения (чтобы было больше рампы разгона при использовании во время разгона или чтобы маска заканчивалась до начала торможения). Маски Y2 и Y0 аналогичны.

Пример программы 2:



Описание:

- При срабатывании M0 Y0 становится импульсным выходом. При обнаружении внешнего прерывания на входе X4 начнется отсчет 50000 импульсов, после чего импульсный выход будет выполнять замедление в течение 10000 импульсов, после чего прекратит подачу импульсов. M1108 будет показывать состояние паузы. Если прерывание не обнаружено, Y0 подаст 1000000 импульсов и остановится.
- Подача прерывания в диапазоне 0-50000 импульсов игнорируется, т.е. процесс будет осуществляться без замедления до достижения 50000 импульсов.

Заметки:

1. Если функция метки выполняется с функцией маски, ПЛК проверит допустимость диапазона маски, а затем произведет замедление. Если диапазон превышен, ПЛК автоматически скорректирует его.
2. Когда команда PLSR или команды позиционирования рампы ускорения/замедления работают, пользователь может проверить импульсы рампы ускорения в DD1127 и импульсы рампы замедления в DD1133.
3. Пользователь может использовать одну скорость позиционирования, когда настройки ускорения/замедления не заданы.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
60	IST	(S) (D1) (D2)	Ручное/автоматическое управление				

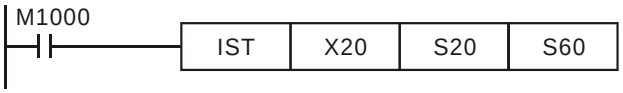
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*													IST: 7 шагов
D_1				*												
D_2				*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных для назначения режимов работы (8 последовательных адресов)
D_1: Младший шаг в автоматическом режиме **D_2**: Старший шаг в автоматическом режиме

Описание:
1. Команда для включения режима шагового управления. Диапазон **D_1** и **D_2**: S20~S911,**D_1** < **D_2**.
2. Данная команда может быть использована только один раз в программе.

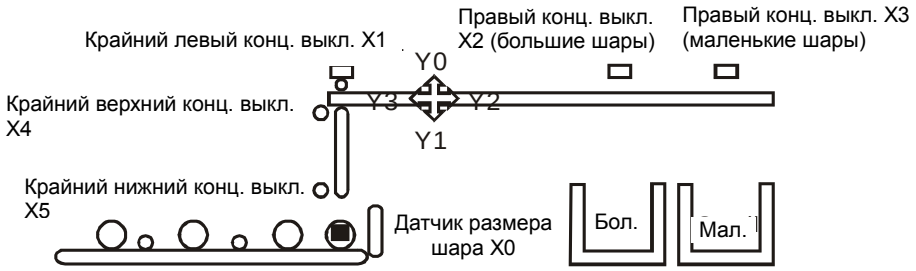
Пример программы 1:



S: X20: Ручной режим
X21: Возврат в исходную позицию
X22: Шаговый режим
X23: Выполнение одного цикла
X24: Автоматический режим
X25: Кнопка возврата в исходную позицию
X26: Кнопка запуска автоматического режима
X27: Кнопка остановки автоматического режима

1. Специальные реле при работе команды IST:
   M1040: Запрещение переходов
   M1041: Старт последующего перехода
   M1042: Импульс статуса
   M1047: Включение индикации состояния шагов
   S0: Ручной наладочный режим
   S1: Перемещение в начальную точку
   S2: Автоматический режим
2. При выполнении команды IST S10-S19 занимают начальные (нулевые точки и не могут использоваться в качестве шаговых точек). Кроме того, S0 запускает ручной режим, S1 - режим возврата в начальное состояние, а S2 - запускает автоматический режим, поэтому эти три точки должны быть запрограммированы в первую очередь.
3. При запущенном режиме S1 возврат в исходное состояние не будет выполняться, если включен любой из S10-S19.
4. При запущенном режиме S2 автоматический режим не будет выполняться, если включен M1043 или активен любой из D1-D21.

Пример программы 2:
Управление роботом-манипулятором (с помощью команды IST):
Задача: Сортировка больших и малых шаров и помещение их в соответствующие коробки. Настройка пульта управления для данной операции.
Движения робота-манипулятора: опускание манипулятора вниз, взятие шара, поднятие манипулятора, перемещение вправо, движение вниз, помещение шара в коробку, поднятие манипулятора, перемещение влево в исходную позицию.

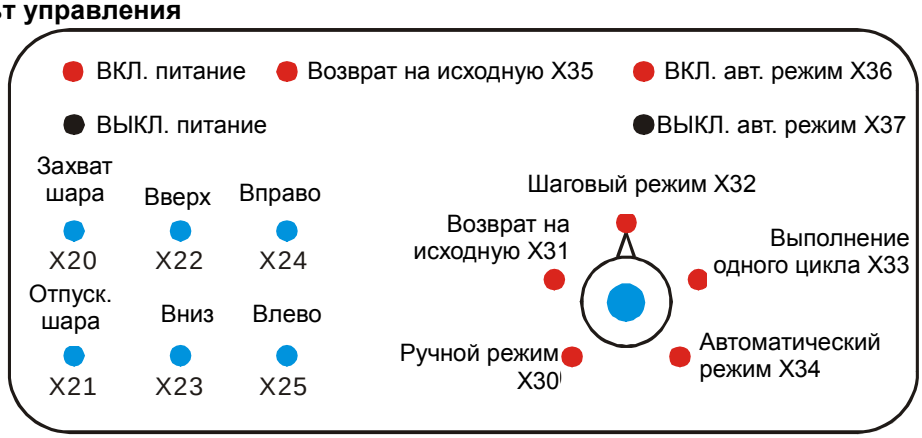


Режимы работы:
1 шаг: Нажать кнопку запуска шагового режима для вкл./выкл. внешнего сигнала.
Возврат в исходное состояние: Нажать кнопку возврата для выполнения возврата механизма в исходное положение.

Автоматический режим (Шаговый / Один цикл / Автоматический режим):

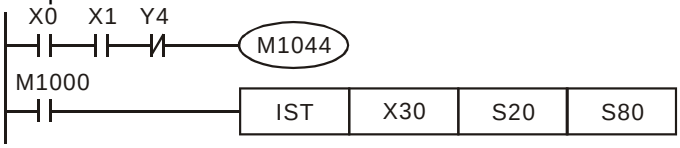
- Шаговый: операция выполняется по одному шагу при нажатии Auto ON (вкл. авт. режима).
- Один цикл: Нажать Auto ON в исходном положении, операция выполнит полный цикл и остановится в исходном положении. При нажатии Auto OFF (выкл. авт. режима) во время цикла, он будет приостановлен, после повторного нажатия Auto ON цикл возобновится до возврата в исходное положение.
- Автоматический режим: нажать Auto ON в исходном положении, операция будет выполняться непрерывно. Если нажать Auto OFF, процесс остановится в конце текущего цикла.

Пульт управления

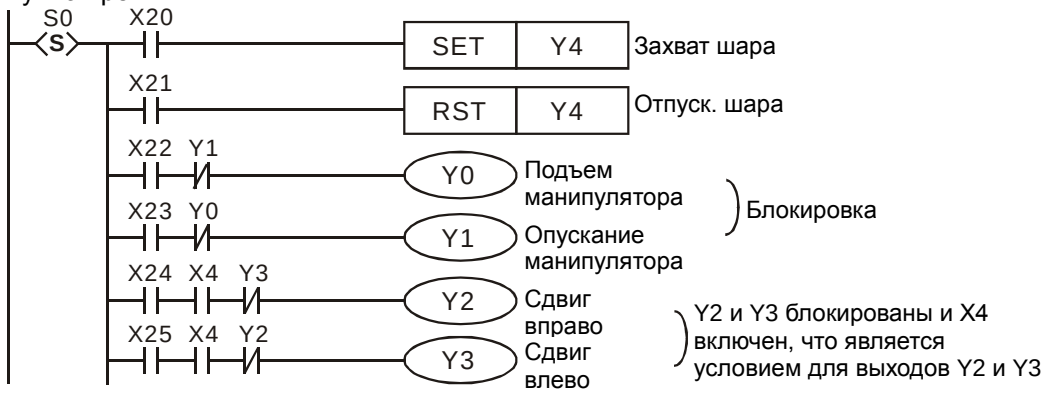


- X0: датчик размера шара.
- X1: крайнее левое положение манипулятора, X2: крайнее правое положение (для больших шаров), X3: крайнее правое положение (для малых шаров), X4: крайнее верхнее положение, X5: крайнее нижнее положение.
- Y0: перемещение вверх, Y1: перемещение вниз, Y2: перемещение вправо, Y3: перемещение влево, Y4: взять шар.

1. Старт:

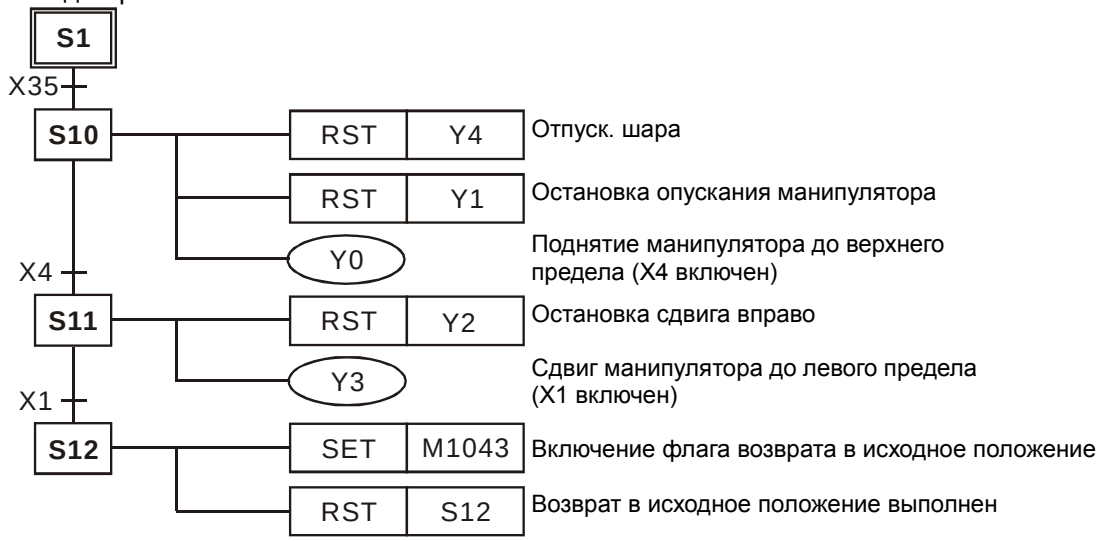


2. Ручной режим:



3. Возврат в исходную позицию:

a) SFC-диаграмма:

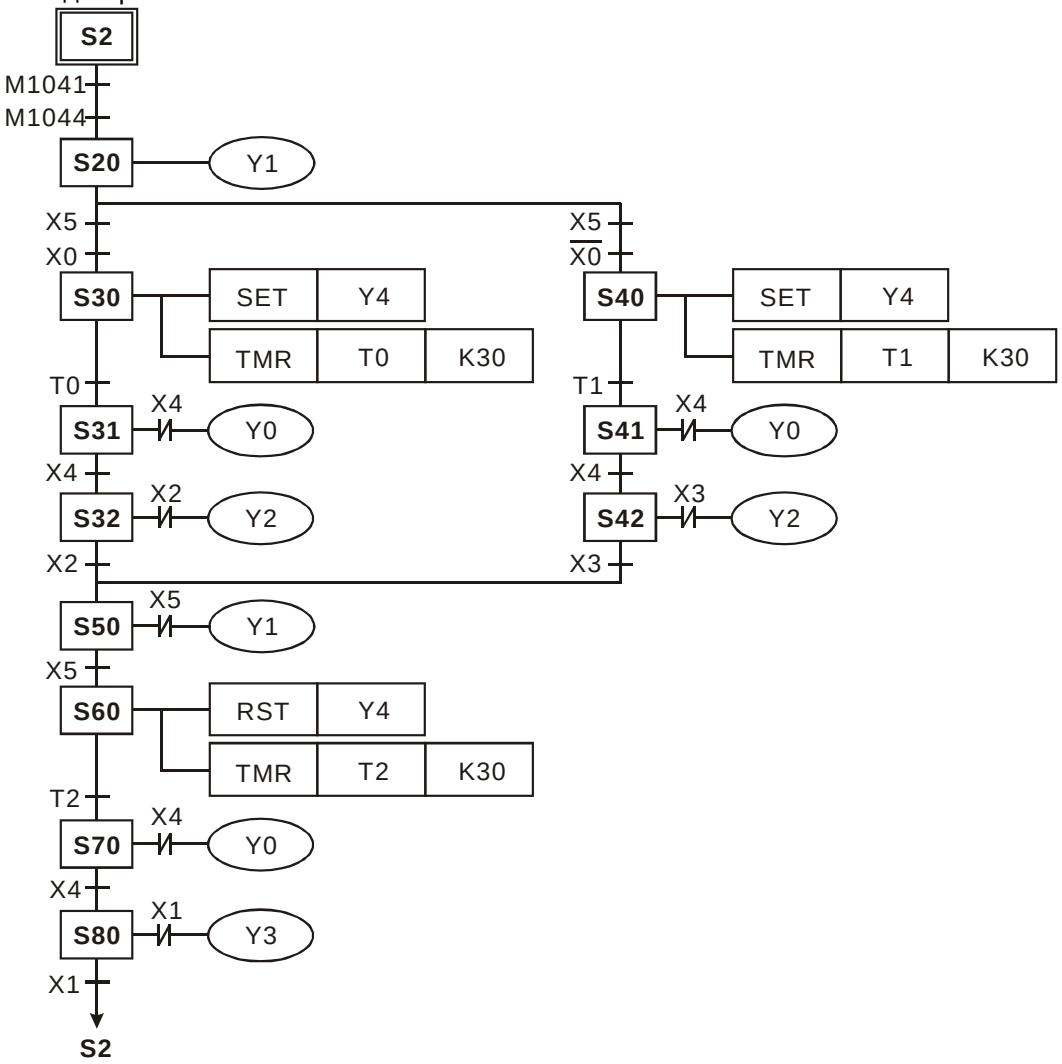


b) Релейно-контактная схема:

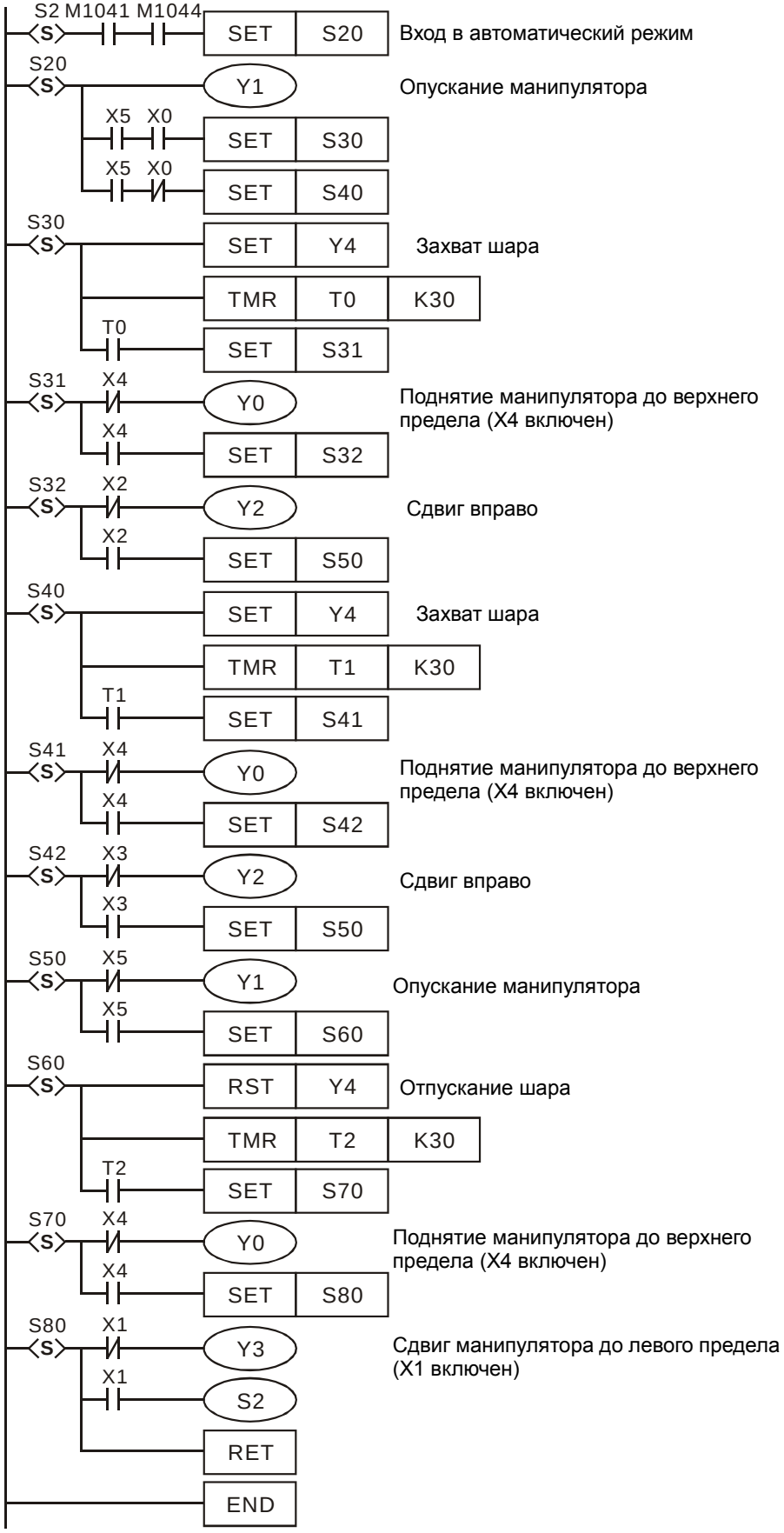


4. Автоматический режим (Шаговый / Один цикл / Автоматический режим):

a) SFC-диаграмма:



b) Релейно-контактная схема:



Описания флагов:

M1040:

Отключение шага перехода. Когда M1040 включен, все шаговые движения не производятся.

1. **Ручной режим:** M1040 остается включенным в ручном режиме.
2. **Режим возврата в исходное положение/режим одного цикла:** M1040 остается включенным в период от нажатия Auto ON (вкл. авт. режима) до нажатия Auto OFF (выкл. авт. режима).

3. **Шаговый режим:** M1040 включен до нажатия Auto ON.
4. **Автоматический режим:** Когда ПЛК включается, M1040 также включается. После нажатия Auto ON M1040 выключается.

M1041:
Шаг начала перехода. Указывает на переход от точки S2 к следующей точке.

1. **Ручной режим/режим возврата в исходное положение:** M1041 выключен.
2. **Шаговый режим/Режим одного цикла:** M1041 включен при нажатии Auto ON.
3. **Автоматический режим:** M1041 остается включенным при нажатии Auto OFF и выключается при нажатии Auto ON.

M1042:
Включение в импульсном режиме: при нажатии Auto ON ПЛК выдает один импульс для данной операции.

M1043:
Возврат в исходное положение произведен: M1043 включен, когда выполнен возврат.

M1044:
Исходное состояние: в автоматическом режиме работы M1044 должен быть включен как условие запуска шагового перехода от точки S2 до следующей точки.

M1045:
Включение функции “сброс все выходов”. Если механизм движется (не в исходной точке):

- Для ручного режима (S0) для возвращения в исходную точку (S1)
- В автоматическом режиме (S2) для перехода в ручной режим (S0)
- В автоматическом режиме (S2) для возвращения в исходную точку (S1)

И
M1045 выключен, любой из активных S между $\mathbf{D_1}$ ~ $\mathbf{D_2}$ будет сброшен, как и Y.
M1045 включен, выход Y будут сохранены, но выполняющийся шаг будет отменен.

M1046:
Индикатор состояния STL. Когда операция STL запущена, M1046 включается, если происходит шаг из любой S. Если M1047 включен, M1046 также включается. D1040-D1047 - сохраняют 8 шаговых состояний.

M1047:
Запуск STL мониторинга. При выполнении команды IST флаг M1047 включается и остается включенным в каждом цикле до конца выполнения команды. Флаг используется для контроля всех шагов (S).

D1040~D1047:
Записывают 8 шаговых состояний.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
61	D	SER	P	S_1 S_2 D n	Поиск данных в стеке				

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1							*	*	*	*	*	*	*		
S_2					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*		
N					*	*							*		

Шаги программы: SER, SERP: 9 шагов; DSER, DSERP: 17 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
$\mathbf{S_1}$: Начальный адрес стека данных $\mathbf{S_2}$: Искомое значение **D**: Начальный адрес хранения результата (занимает 5 последовательных адресов) **n**: Длина стека

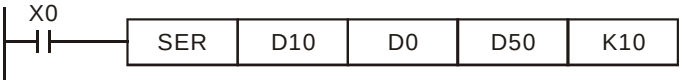
Описание:

1. Команда осуществляет поиск значения из S2 в стеке данных, начиная с адреса S1 и с длиной стека n. Результат сохраняется в 5 регистрах, начиная с D.
2. D содержит общее количество полученных результатов, D+1 содержит первый результат, D+2 содержит последний, D+3 содержит наименьшее найденное значение, D+4 - наибольшее.
3. Если операнд S2 используется с индексом F, то возможно выполнение только 16-битных команд.

4. Если команда работает в 32-битном режиме, операнды **S_1**, **S_2**, **D**, **n** находятся в 32-битных регистрах.
5. Необходимое условие для операнда n: **n** = 1~256 (16-битная команда), **n** = 1~128 (32-битная команда)

Пример программы:

1. Когда X0 включен, данные стека D10-D19 сравниваются с D0 и результат сохраняется в D50-D54. Если результата нет, содержимое D50-D52 будет равно 0.
2. В D53, D54 сохраняются данные с самым большим и самым малым значением. Если таких значений несколько, запоминается самое последнее значение.



S_1	Содержимое	Данные для поиска	Позиция	Результат
D10	88		0	
D11	100	S_2	1	равно
D12	110		2	
D13	150		3	
D14	100		4	равно
D15	300	D0=K100	5	
D16	100		6	равно
D17	5		7	наименьшее
D18	100		8	равно
D19	500		9	наибольшее

n

D	Содержимое	Описание
D50	4	Общее количество эквивалентных данных
D51	1	Номер первого эквивалентного значения
D52	8	Номер последнего эквивалентного значения
D53	7	Номер наименьшего значения
D54	9	Номер наибольшего значения

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
62	D	ABSD	S_1 S_2 D n	Абсолютный многоустановочный счетчик				

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1							*	*	*	*	*	*	*		
S_2											*	*	*		
D		*	*	*											
n					*	*									

Шаги программы:
ABSD: 9 шагов
DABSD: 17 шагов

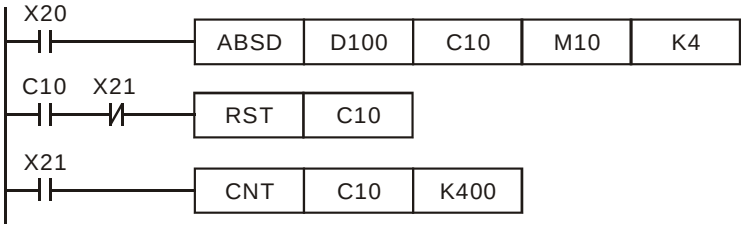
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Первый операнд таблицы данных **S_2**: Номер счетчика **D**: Первое значение результата **n**: Количество операндов результата (**n**: 1~64)

Описание:

1. Данная команда сравнивает текущее накопленное значение счетчика (S2) с таблицей операндов, имеющей первое значение S1 и n строк. Если счетчик (S2) получил накопленное значение, занесенное в таблицу, включается относящийся к нему операнд.
2. Если операнд S1 имеет значение KnX, KnY, KnM, KnS, то Kn должен иметь значение K4 для 16-битных команд и K8 для 32-битных команд.
3. Команде DABSD может в S2 использовать высокоскоростной счетчик. Однако когда текущее значение в счетчике сравнивается с заданным, результат выводится с задержкой, зависящей от времени выполнение цикла. Для немедленного вывода результата необходимо применять команду DHSZ специально предназначенную для работы с высокоскоростными счетчиками.

Пример программы:



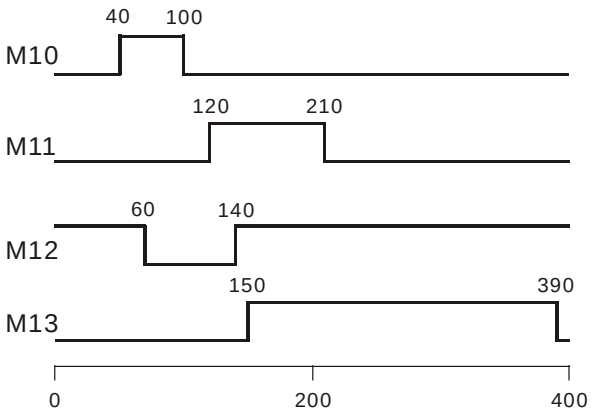
1. Перед выполнением команды ABSD необходимо заранее использовать команду MOV для задания значений D100-D107. Для нижней границы значений D должно быть четным, а для верхней - нечетным.

2. Когда X10 включен, текущее значение в счетчике C10 будет сравниваться с четырьмя группами верхних и нижних значений в D100~D107. Результат сохраняется в M10~M13.
3. Когда X10 выключен, существующие значения M10-M13 будут сохранены.
4. M10~ M13 включены, когда текущее значение C10 находится в пределах между верхней и нижней границами.

Нижняя граница	Верхняя граница	Значение счетчика C10	Выход
D100= 40	D101 = 100	40≤C10≤100	M10 = ON
D102 = 120	D103 = 210	120≤C10≤210	M11 = ON
D104 = 140	D105 = 170	140≤C10≤170	M12 = ON
D106 = 150	D107 = 390	150≤C10≤390	M13 = ON

5. Если нижняя граница имеет большее значение, чем верхняя, то при C10<60 или C10 > 140, M12 включается.

Нижняя граница	Верхняя граница	Значение счетчика C10	Выход
D100 = 40	D101 = 100	40≤C10≤100	M10 = ON
D102 = 120	D103 = 210	120≤C10≤210	M11 = ON
D104 = 140	D105 = 60	60≤C10≤140	M12 = OFF
D106 = 150	D107 = 390	150≤C10≤390	M13 = ON



API	Команда	Операнды	Функция	Контроллеры
63	INCD	S_1 S_2 D n	Инкрементный многоустановочный счетчик	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			INCD: 9 шагов
S_2												*				
D		*	*	*												
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

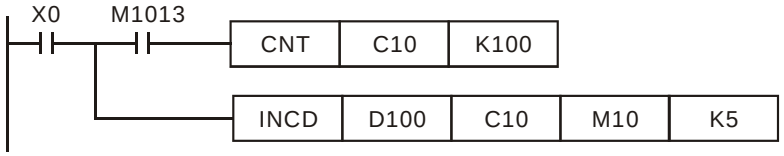
Операнды:
S_1: Первый операнд таблицы данных **S_2**: Номер счетчика **D**: Первое значение результата **n**: Количество операндов результата сравнения (**n**: 1~64)

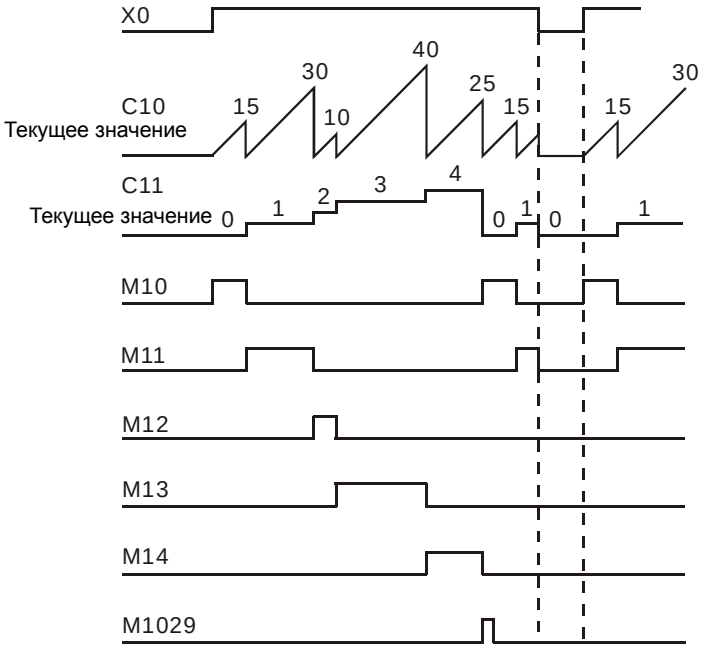
Описание:
1. Команда задает выходное значение в D в соответствии с текущим значением счетчика S2 и S2+1.
2. Текущее значение S2 сравнивается с данными, заданными в S1 (n рядов). В S2+1 записывается количество сбросов счетчика S2 при достижении текущим значением заданной точки. Когда устройство, включившее выполнение команды, выключается, содержимое S2, S2+1 будет очищено.

3. Если операнд S1 имеет значение KnX, KnY, KnM, KnS, то Kn должен иметь значение K4 для 16-битных команд.
4. Операнд S2 имеет значение C0-C198 и занимает 2 последовательных счетчика.
5. По завершении выполнения команды включается флаг M1029 (для каждого цикла).

Пример программы:

1. Перед выполнением команды INCD необходимо заранее выполнить команду MOV для задания значений D100-D104. D100 = 15, D101 = 30, D102 = 10, D103 = 40, D104 = 25.
2. Текущее значение счетчика C10 сравнивается с заданным в D100-D104. При равенстве этих значений C10 сбрасывается и вновь начинает подсчет с нуля. При этом в C11 прибавлятся 1 (один сброс).
3. Когда содержимое C11 увеличивается на 1, M10-M14 последовательно включаются. См. временную диаграмму.
4. После завершения 5 шагов сравнения, включается флаг завершения M1029 (для 1 цикла) и значение C11 сбрасывается до следующего цикла.
5. Когда X0 выключается, C10 и C11 сбрасываются до 0 и флаги M10~M14 выключаются. Когда X0 включается снова, команда выполняется сначала.





API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
64	TTMR	(D) (n)	Обучающийся таймер				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D													*			TTMR: 5 шагов
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

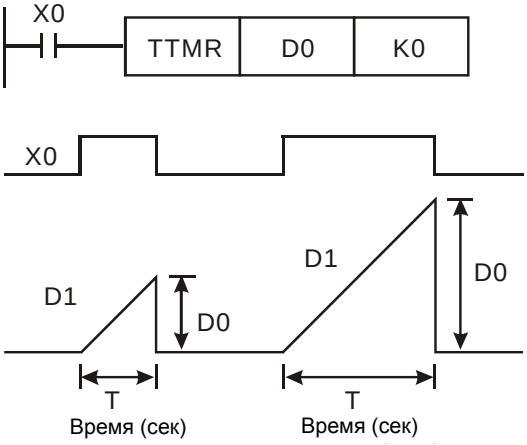
D: Хранение времени включения инструкции **n**: множитель времени (**n**: K0~K2)

Описание:

1. Время нажатия кнопки измеряется и хранится в D+1 (время в мс). Значение D+1 умножается на n и сохраняется в D (время в сек)
2. Когда **n** = K0, значение **D + 1**(100 мс) умножается на 1 и сохраняется в **D** (сек).
3. Когда **n** = K1, значение **D + 1**(100 мс) умножается на 10 и сохраняется в **D** (сек).
4. Когда **n** = K2, значение **D + 1**(100 мс) умножается на 100 и сохраняется в **D** (сек).

5. Команда TTMR может быть использована не более 8 раз в программе.

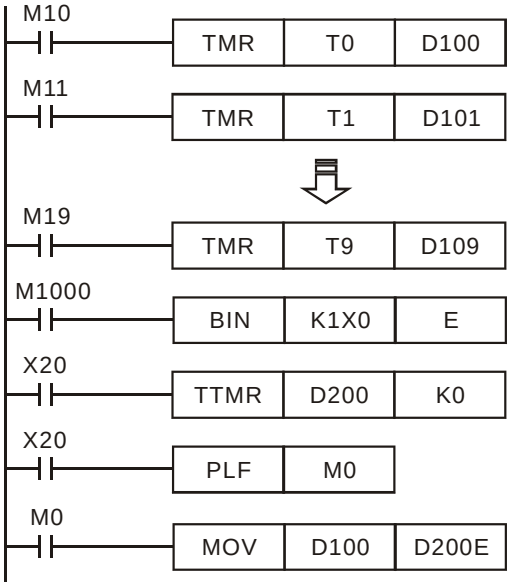
Пример программы 1:



1. Продолжительность нажатия X0 сохраняется в D1. Значение D1, умноженное на коэффициент n и помещается в D0. В этом случае кнопка может использоваться для задания значения таймера.
2. Когда X0 выключен, содержимое D1 будет сброшено, а содержимое D0 сохранится.
3. Если время нажатия X0 - T сек, соотношение между D0, D1 и **n** представлены в таблице ниже:

n	D0 (сек)	D1 (мс)
K0	T (сек) ×1	D1 = D0×10
K1	T (сек) ×10	D1 = D0
K2	T (сек) ×100	D1 = D0/10

Пример программы 2:



1. Использование команды TMR для установки времени в 10 группах.
2. Предварительная запись набора значений в D100 ~ D109.
3. Разрешение таймера 0,1 сек для таймеров T0-T9 и 1 сек для обучающегося таймера.
4. Подсоединение 1-битового DIP-переключателя к X0 ~ X3 и использование команды BIN для преобразования значения переключателя в двоичную величину и сохранения его в E.
5. Продолжительность нажатия (в сек) X20 сохраняется в D200.
6. M0 задает импульс для одного цикла, когда обучающийся таймер X20 фиксирует отпускание кнопки.
7. Набор DIP-переключателей формируют данные для отправки содержимого в D200 - D100E (D100 ~ D109).

Заметки:
Команда TTMR может быть использована не более 8 раз в программе. Если команда TTMR используется для вызова или прерывания подпрограммы, данную процедуру можно применить только 1 раз.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
65	STMR	(S) (m) (D)	Специальный таймер				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*					STMR: 7 шагов
m					*	*										
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

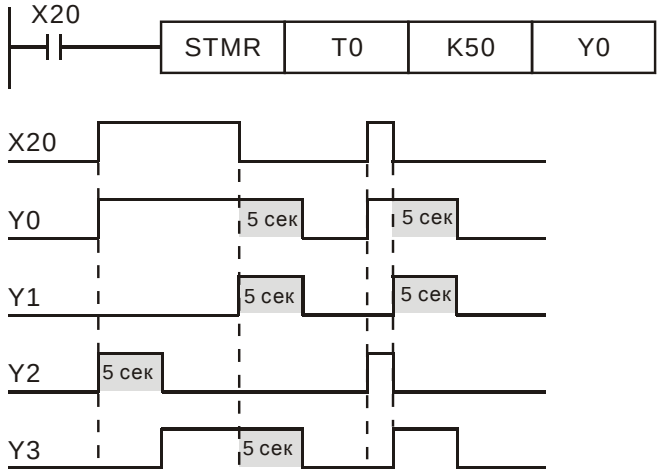
Операнды:
S: Таймер (T0~T183) **m**: Уставка таймера (**m** = 1~32,767, шаг настройки: 100 мс)
D: Начальное значение операнда выхода (занимает 4 последовательных адреса)

Описание:
1. Команда STMR предназначена для задержки выключения, формирования импульса по включению/выключению, создания импульсных схем с периодом выходного сигнала, задаваемым таймером.

- (D): включается одновременно с входным условием инструкции, а выключается после выключения входного условия через заданную в (m) задержку времени;
- (D+1): включается одновременно с выключением входных условий инструкции, а выключается через время, заданное в (m);
- (D+2): включается одновременно с включением входных условий инструкции, а выключается через время, заданное в (m);
- (D+3): включается через время (m) после включения входных условий инструкции, и выключается после выключения входного условия через заданную в (m) задержку времени;

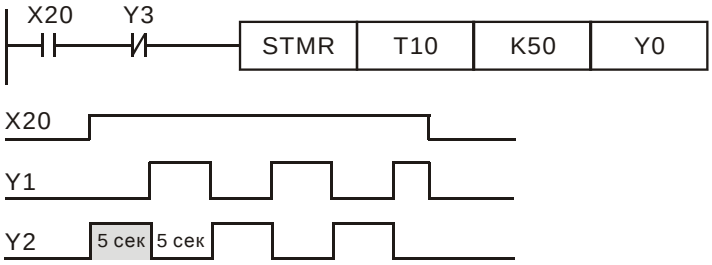
2. Таймер (**S**), используемый командой STMR, может быть использован только один раз.

Пример программы:



1. Когда X20 включен, STMR устанавливает T0 на 5 сек. специального таймера.
2. Y0 контакт с задержкой отключения, поэтому при включении X20 включится и Y0; когда X20 выключается, Y0 выключается с задержкой в 5 сек.
3. При выключении X20 на 5 сек. включится Y1.
4. Когда X20 включается, Y2 включается на 5 сек.

   Когда X20 включается, Y3 включается с задержкой в 5 сек. Когда X20 выключается, Y3 выключается с задержкой в 5 сек.



5. Применение H/3 контакта Y3 после X20 позволяет Y1 и Y2 образовывать выходной импульсный сигнал с частотой, задаваемой таймером. Когда X20 отключается, Y0,Y1 и Y3 выключаются и содержимое T10 будет сброшено.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
66	ALT	P	D	Импульсное реле (T-триггер)				

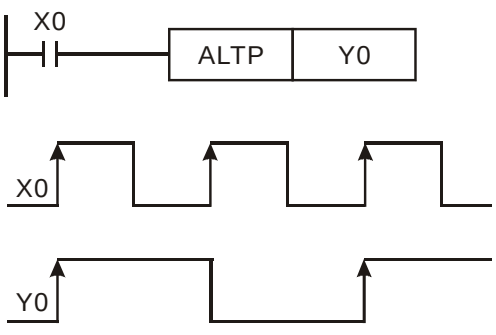
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D		*	*	*												ALT, ALTP: 3 шага

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D: Результат

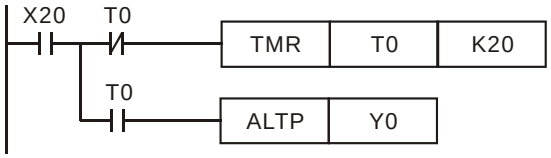
Описание:

1. Состояние D (0 или 1) меняется на противоположное при каждом выполнении команды ALT, что обычно применяется при работе в 2 режимах, например, Пуск/Стоп.
2. Команда ALT активизируется по управлению сигналом "1" и записывается в операнд, указанный в D. Новым управлением сигнала в "1" операнд отключается.
3. Эта команда обычно используется в импульсном режиме (ALTP).



Пример программы 1:
Когда X0 включается, Y0 также включен. Когда X0 снова включается, Y0 выключается.

Пример программы 2:
Создание импульсного сигнала с помощью команд ALTP и таймера.
Когда X20 включается, T0 будет генерировать импульсы каждые 2 сек., которые будут включать/выключать выход Y0.



API	Команда	Операнды	Функция	Контроллеры
67	RAMP	(S1) (S2) (D) (n)	Линейное изменение сигнала (RAMP)	ES2/EX2 | SS2 | SA2/SE | SX2

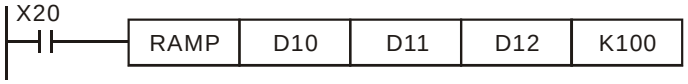
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			RAMP: 9 шагов DRAMP: 17 шагов
S_2													*			
D													*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходное значение **S_2**: Целевое значение **D**: Текущее значение RAMP-функции (занимает 2 последовательных адреса) **n**: Количество циклов (**n**: 1~32767)

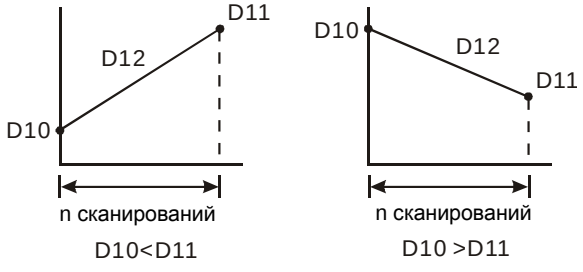
Описание:
1. Эта команда создает выходной RAMP-сигнал (линейный нарастающий или спадающий). Линейность его зависит от времени цикла программы. Поэтому ПЛК должно работать с постоянным временем цикла программы, которое должно быть установлено перед запуском команды.
2. При выполнении команды RAMP-сигнал будет изменяться от S1 до S2. Текущее значение сигнала хранится в регистрах D, а в D+1 сохраняется количество циклов. Значение в регистре D в случае, когда RAMP-сигнал достигает S2 и когда вход, управляющий командой, выключается, будут различны (см. пример и заметки).
3. Если n задает регистр D, его значение не может быть изменено во время выполнения команды.
4. Функция может быть применена совместно с функцией аналогового выхода.

Пример программы:



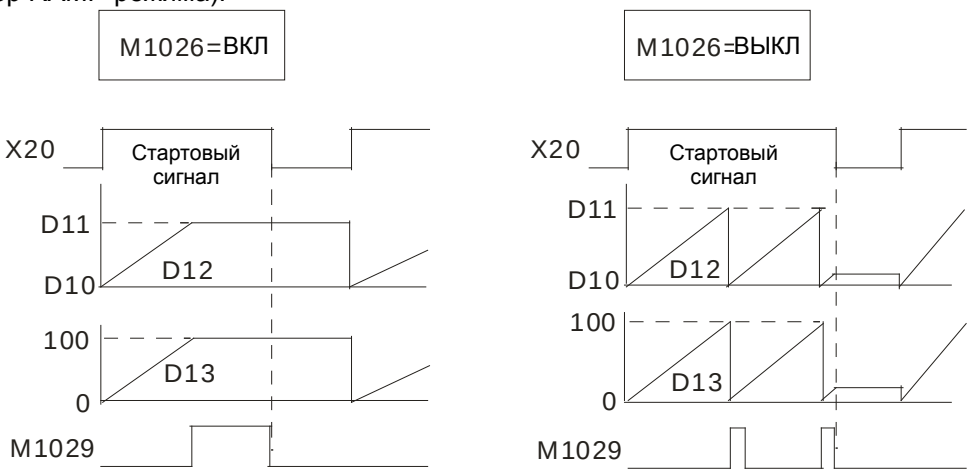
1. Перед выполнением команды необходимо включить флаг M1039 для установки времени цикла. С помощью команды MOV время записывается в специальный регистр D1039. Если время сканирования составляет 30 мс и количество циклов n=K100, время увеличения значения от D10 до D11 составит 3 сек.
2. Выключение X20 остановит выполнение команды. Когда X10 снова включится, значение D12 сбросится в исходное состояние и начнется новое выполнение команды.
3. Когда M1026 выключен, M1029 включится, показывая завершение процесса, и содержимое D12 будет сброшено до начального значения D10.
4. Установка начального и конечного значения RAMP-сигнала в D10 и D11. Когда X20 включен, значение D10 увеличивается до значения D11, текущее значение сохраняется в D12, а текущее количество циклов сохраняется в D13.

Если X20 включен,



Время сканирования указывается в D13

Заметки:
Изменение содержимого D12 в соответствии с включением/выключением флага M1026 (выбор RAMP-режима):



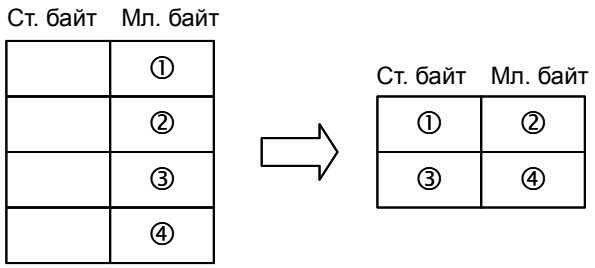
API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/ SE	SX2
68	DTM		(S1) (D) (m) (n)	Изменение и перемещение данных				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			DTM: 9 шагов
D													*			
m					*	*							*			
n					*	*							*			

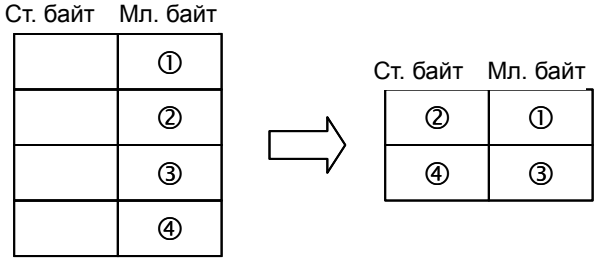
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S_1: Начальный адрес стека данных источника **D**: Начальный адрес стека данных цели **m**: Режим изменения **n**: Длина стека исходных данных

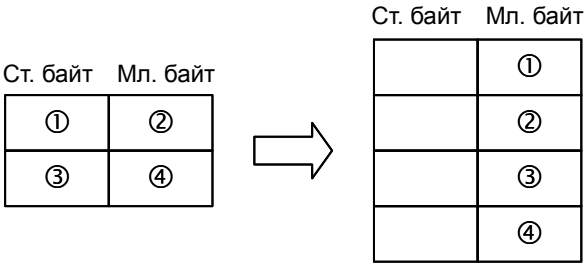
Описание:
1. При настройке параметров операнда m можно задавать значения K, H, D. Если заданное значение находится в перечне допустимых, операция изменения и перемещения будет осуществляться безошибочно.
2. Значения K, H, D можно задавать операнду n - указателю длины стека исходных данных. Доступный диапазон для n: 1-256. Если заданное значение выходит за пределы диапазона, ПЛК примет максимальное (256) либо минимальное значение (1) автоматически.
3. Параметры и настройки операнда m:



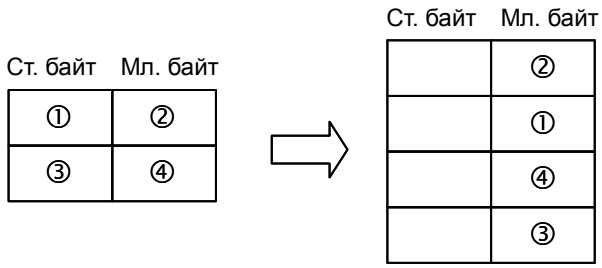
K0: При n=4, преобразуются 8-битные данные в 16-битные (старший бит, младший бит) по следующей схеме:



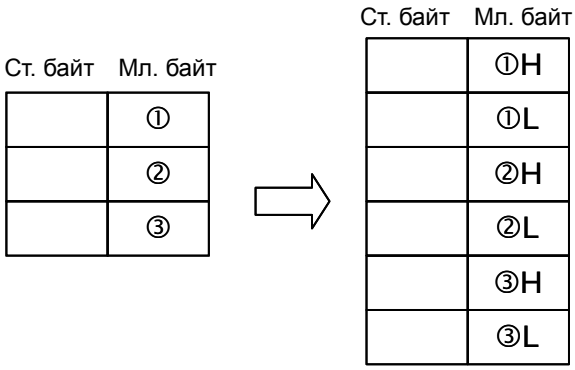
K1: При n=4, преобразуются 8-битные данные в 16-битные (младший бит, старший бит) по следующей схеме:



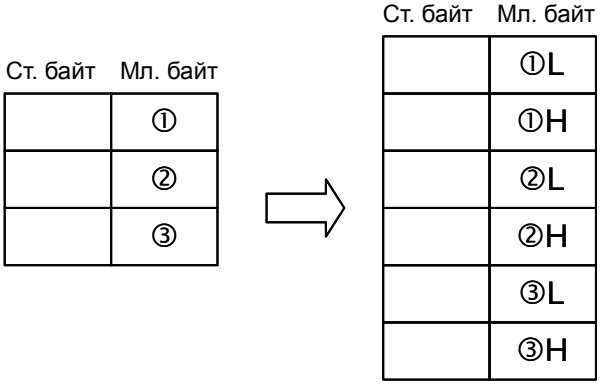
K2: При n=2, преобразуются 16-битные данные в 8-битные (старший бит, младший бит) по следующей схеме:



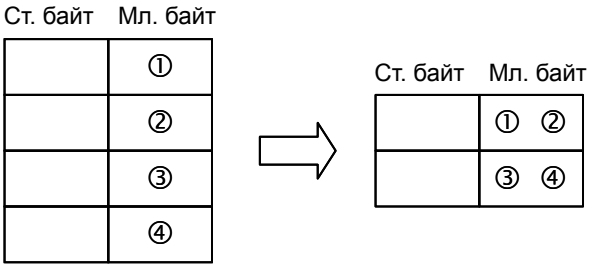
K3: При n=2, преобразуются 16-битные данные (младший бит, старший бит) в 8-битные по следующей схеме:



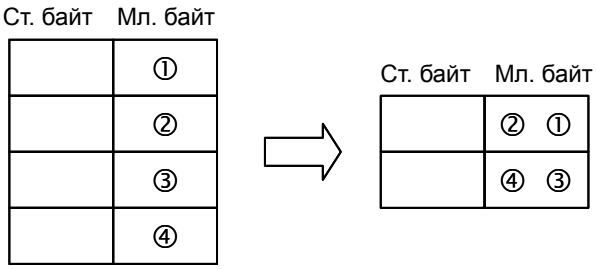
K4: При n=3, преобразуются 8-битные шестнадцатеричные данные в ASCII-данные (4 старших бита, 4 младших бита) по следующей схеме:



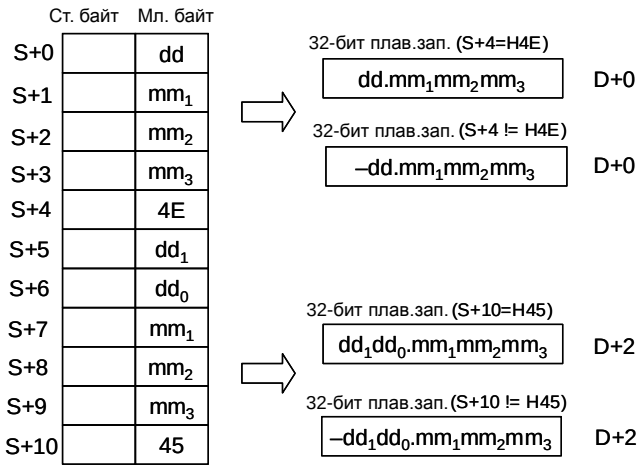
K5: При n=3, преобразуются 8-битные шестнадцатеричные данные в ASCII-данные (4 младших бита, 4 старших бита) по следующей схеме:



K6: При n=4, преобразуются 8-битные ASCII-данные (4 старших бита, 4 младших бита) в шестнадцатеричные данные по следующей схеме: (ASCII-значения для преобразования включают в себя 0 ~ 9 (0x30~0x39), A ~ F (0x41~0x46), и a ~ f (0x61~0x66).)



K7: При n=4, преобразуются 8-битные ASCII-данные (4 младших бита, 4 старших бита) в шестнадцатеричные данные по следующей схеме:



K8: Преобразование 8-битных GPS данных в 32-битные данные с плавающей запятой по следующей схеме:

K9: Расчет оптимальной частоты для выполнения команд управления позиционированием с RAMP-функциями. Команда DTM автоматически рассчитывает оптимальную выходную частоту и оптимальную начальную частоту для работы команд PLSR, DDRVI, DCLLM (работа с RAMP-функцией), исходя из количества импульсов позиционирования и времени позиционирования.

Заметки:

1. Если результат расчета превышает максимальную частоту ПЛК, выходная частота будет равна 0.
2. Если общее время разгона и замедления RAMP-операции превышает общее время позиционирования (S+2), ПЛК изменит его на "время разгона (S+3) + время замедления (S+4) +1" автоматически.

Операнды:

S+0, **S**+1: Общее количество импульсов позиционирования(32-бит)
S+2: Общее время позиционирования (мс)
S+3: Время разгона (мс)
S+4: Время замедления (мс)
D+0, **D**+1: Оптимальная максимальная выходная частота (Гц) (32-бит)
D+2: Оптимальная начальная частота (Гц)
n: Зарезервировано

K11: Преобразование местного поясного времени в местное астрономическое время. В отличие от местного поясного времени, астрономическое время имеет четкую привязку к долготе. Преобразование позволяет получить точное значение времени в пределах одного часового пояса.

Операнды:

S+0, **S**+1: Долгота (32-битное значение с плавающей запятой; Восток: положительное, Запад: отрицательное)
S+2: Часовой пояс (16-битное целое; час)
S+3~ **S**+8: Год, Месяц, День, Час, Минута, Секунда местного поясного времени (16-битное целое)
D+0~**D**+5: Год, Месяц, День, Час, Минута, Секунда местного астрономического времени (16-битное целое)
n: Зарезервировано

Пример:

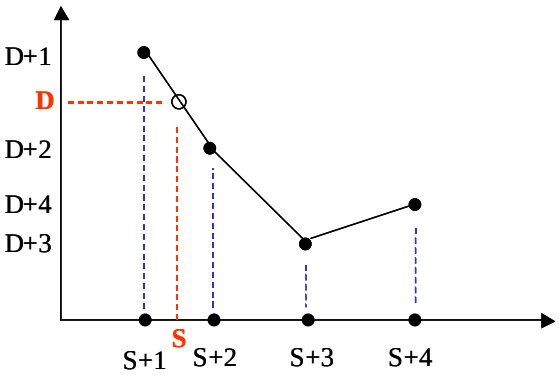
Задано: Долгота F121.55, Часовой пояс: +8, Местное поясное время: AM 8:00:00, Январь/6/2011
Результат преобразования: AM 8:06:12, Январь/6/2011

K12: Функция расчета пропорционального значения на многоточечных кривых (16-битные значения)

Операнды (16-битные):

S: входное значение
S+1, **S**+2….. **S**+n: абсциссы точек кривой. **S**+1 должно быть минимальным значением, **S**+2 должно быть больше **S**+1 и т.д. Поэтому **S**+n должно быть максимальным значением.
D: выходное значение, полученное пропорциональным расчетом.
D+1, **D** +2 … **D**+n: ординаты точек кривой.
n: установки значений многоточечных зон. Допустимый диапазон K2-K50. Если заданное значение превышает допустимый диапазон, операция выполнена не будет.

Пример кривой: (n установлен как K4)



Описание примера:

1. Когда входное значение S больше, чем S+1 (S_1 для краткости) и меньше, чем S+2 (S_2 для краткости), D+1 (D_1 для краткости) и D+2 (D_2 для краткости), D= ((S – S_1) x (D_2 – D_1) / (S_2 – S_1)) + D_1.
2. Когда входное значение S меньше, чем S+1, D= D+1; когда входное значение S больше, чем S+n, D= D+n.
3. Команда использует значения с плавающей запятой. Если десятичные знаки выходного значения не указаны, значение получается в 16-битном виде.

K13: Функция расчета пропорционального значения в многоточечных кривых (32-битные значения)

Обозначения аналогичны K12, но S и D имеют 32-битные значения.

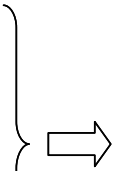
K14: Функция расчета пропорционального значения в многоточечных кривых (значения с плавающей запятой)

Обозначения аналогичны K12, но S и D имеют 32-битные значения с плавающей запятой.

K16: Комбинация строк

	Ст. байт	Мл. байт
S+0		‘A’
S+1		‘B’
S+2		‘C’
S+3		‘D’
S+4		0x00

	Ст. байт	Мл. байт
D+0		‘a’
D+1		‘b’
D+2		‘c’
D+3		0x00



	Ст. байт	Мл. байт
D+0		‘a’
D+1		‘b’
D+2		‘c’
D+3		‘A’
D+4		‘B’
D+5		‘C’
D+6		‘D’
D+7		0x00

Система ищет ETX (значение 0x00) в строке данных результата (младшие 8 бит), а затем копирует данные строки, начиная с начальной (младшие 8 бит), в конец строки данных результата. Строки будут копироваться пока ETX (значение 0x00) не будет достигнуто. Операнд n устанавливает максимальную длину данных после строки комбинации (максимум 256).
Комбинация строк осуществляется по следующей схеме:

K17: Захват строк

	Ст. байт	Мл. байт
S+0		‘a’
S+1		‘b’
S+2		‘c’
S+3		‘A’
S+4		‘B’
S+5		‘C’
S+6		‘D’
S+7		0x00



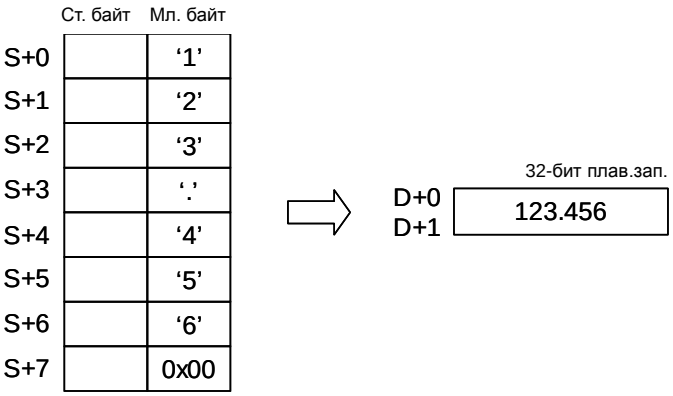
	Ст. байт	Мл. байт
D+0		‘a’
D+1		‘b’
D+2		‘c’
D+3		0x00

Система копирует строку источника данных (младшие 8 бит) с данными длиной n в регистр результата, где регистр n+1 будет иметь значение 0x00. Если значение 0x00 будет достигнуто до указанного предела n, захват будет закончен.
Захват строк осуществляется по следующей схеме:

K18: Преобразование строковых данных в данные с плавающей точкой
Система преобразует n слов (младшие 8 бит) в строках исходных данных (без десятичных) в значение с плавающей запятой и сохраняет полученное значение в операнде результата.

Заметки:
1. Операнд n устанавливает общее число знаков для значения с плавающей запятой. Максимально возможно 8 цифр, остальные будут просто проигнорированы. Например, n=K6, строковые данные 123,45678 будут преобразованы в 123,456.
2. Если есть символы, отличные от цифр 0-9, или десятичная запятая в строке источника данных, то символ перед запятой будет рассматриваться как 0, а значение после запятой, как ETX.
3. Если строка источника данных не содержит запятую, преобразованное значение с плавающей запятой (n разрядов) будет отображаться автоматически.

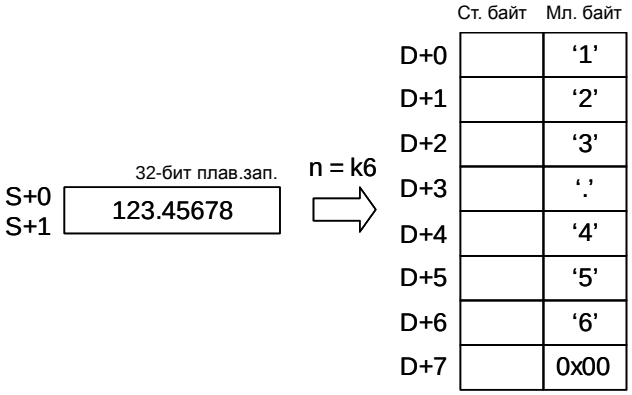
Преобразование осуществляется по следующей схеме:



K19: Преобразование данных с плавающей точкой в строковые данные
Система преобразует значение с плавающей запятой из источника данных S в строковые данные заданной длины n (без учета запятой).

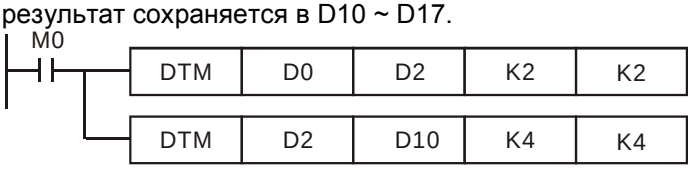
Заметки:
1. Операнд n устанавливает общее число знаков для значения с плавающей запятой, которое должно быть преобразовано. Максимально возможно 8 цифр, остальные будут просто проигнорированы. Например, n=K6, данные с плавающей запятой F123,45678 будут преобразованы в 123,456.
2. Когда знаков в исходном значении больше, чем указанное n, будет преобразовано только n знаков. Например, исходное значение F123456,78 с n=K4 будет преобразовано в данные строки 1234.
3. Если исходное десятичное значение не имеет целой части, например 0,1234, то преобразованные данные будут .1234 по первым цифрам после запятой.

Преобразование осуществляется по следующей схеме:



Пример программы 1: K2, K4

1. Когда M0 включен, преобразовываются 16-битные данные в D0, D1 в ASCII-данные в следующем порядке: Старший байт-Младший байт-Старший байт-Младший байт, и результат сохраняется в D10 ~ D17.



2. Значение D0, D1:

Регистр	D0	D1
Знач.	H1234	H5678

3. Когда первая команда DTM выполнена (m=K2), ПЛК преобразует 16-битные данные (Старший байт, младший байт) в 8-битные данные и сохраняет их в D2~D5.

Регистр	D2	D3	D4	D5
Знач.	H12	H34	H56	H78

4. После второго выполнения команды DTM (m=K4), ПЛК преобразует 8-битные шестнадцатеричные данные в ASCII-данные и сохраняет их в регистрах D10~D17.

Регистр	D10	D11	D12	D13	D14	D15	D16	D17
Знач.	H0031	H0032	H0033	H0034	H0035	H0036	H0037	H0038

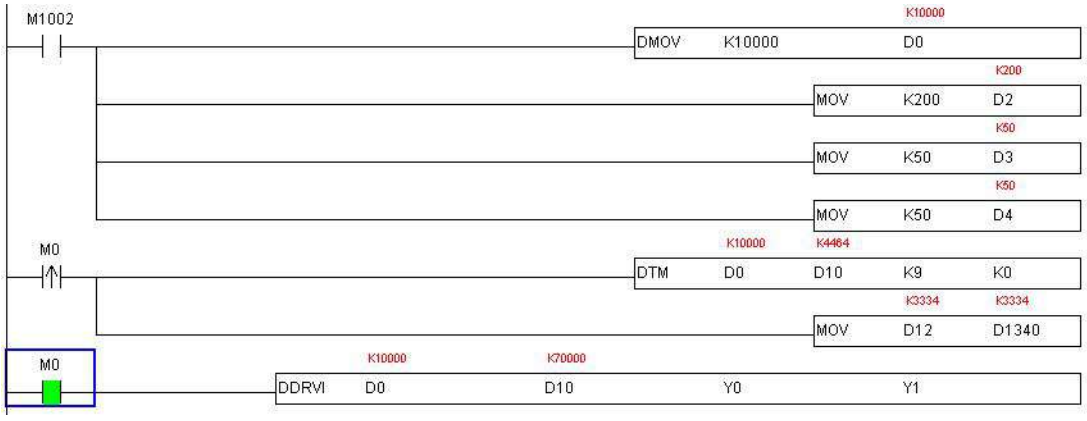
Пример программы 2: K9

1. Установка общего количества импульсов, общего времени, времени разгона и замедления RAMP-функции в источнике сигнала, начиная с D0. Выполнение команды DTM дает возможность определить оптимальную максимальную и начальную частоту для позиционирования.
2. Задание параметров:

Общее кол-во импульсов	Общее время	Время разгона RAMP-функции	Время замедления RAMP-функции
D0, D1	D2	D3	D4
K10000	K200	K50	K50

3. Результаты расчета данных оптимального позиционирования:

Оптимальная максимальная частота	Оптимальная начальная частота
D10, D11	D12
K70000	K3334



API	Команда		Операнды	Функция	Контроллеры			
69	D	SORT	S m1 m2 D n	Сортировка данных	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			SORT: 11 шагов DSORT: 21 шаг
m_1					*	*										
m_2					*	*										
D													*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

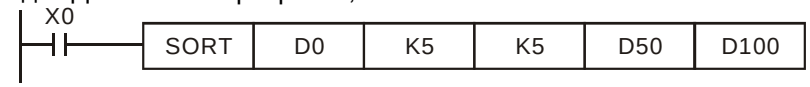
S: Начальное значение исходных данных **m_1**: количество строк в таблице сортировки (**m_1** =1~32) **m_2**: Количество столбцов в таблице (**m_2** =1~6) **D**: Начальное значение отсортированных данных **n**: Номер столбца, по которому будет сортировка (**n**=1~ **m2**)

Описание:

1. Отсортированные данные сохраняются в m1 x m2 регистрах, начиная с D. S и D могут указываться в том же регистре.
2. Желательно, чтобы номер самого правого устройства, указанный в регистре S, был 0.

3. Выполнение команды SORT заканчивается только тогда, когда в m1 будет получено заданное число. По завершении включается флаг завершения выполнения M1029.
4. Не существует ограничений на количество использований команды. Тем не менее, одновременно может выполняться только одна команда.
5. Добавлена функция сортировки одномерных данных. При m1 и m2 = 1 команда начинает выполняться и в этом случае n представляет собой количество данных (n=1~32). Данные n устройств, начиная с S, сортируются, и результат сохраняется в операнде D. Сортировка происходит за один цикл сканирования. После завершения сортировки включается флаг завершения выполнения M1029. Функция поддерживается ПЛК серий SS2 – версия V3.0 и выше, SA2 – версия V2.6 и выше, SX2 – версия V2.4 и выше, ES2/EX2/ES2-C – версия V3.2 и выше.

Пример программы 1:
Когда X0 включен, запускается процесс сортировки. После завершения сортировки включается M1029. НЕЛЬЗЯ ИЗМЕНИТЬ данные для сортировки в процессе выполнения команды. Для новой сортировки, выключите и снова включите X0.



Пример таблицы данных для сортировки

Колонки данных: m_2

Строки данных: m_1	Колонка данных				
Строка \ Колонка	1	2	3	4	5
	Группа	Англ.	Математ.	Физики	Химики
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

Сортировка данных для D100 = K3 (по математикам)

Колонки данных: m_2

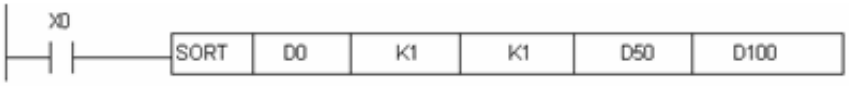
Строки данных: m_1	Колонка данных				
Строка \ Колонка	1	2	3	4	5
	Группа	Англ.	Математ.	Физики	Химики
1	(D50) 4	(D55) 70	(D60) 60	(D65) 99	(D70) 50
2	(D51) 2	(D56) 55	(D61) 65	(D66) 54	(D71) 63
3	(D52) 1	(D57) 90	(D62) 75	(D67) 66	(D72) 79
4	(D53) 5	(D58) 95	(D63) 79	(D68) 75	(D73) 69
5	(D54) 3	(D59) 80	(D64) 98	(D69) 89	(D74) 90

Сортировка данных для D100 = K5 (по химикам)

Колонки данных: m_2

Строки данных: m_1	Колонка данных				
Строка \ Колонка	1	2	3	4	5
	Группа	Англ.	Математ.	Физики	Химики
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

Пример программы 2: (сортировка одномерных данных)
Когда X0 включен, запускается процесс сортировки. После завершения сортировки включается M1029. При **m1** = K1 и **m2** = K1, одномерные данные сортируются. Значение в D100 равно K5. Значения в D0~D4 приведены ниже.



Значения в D0~D4:

Источник данных (S)	D0	D1	D2	D3	D4
Данные	75	65	98	60	79

Отсортированные данные в D50~D54:

Результат сортировки (D)	D50	D51	D52	D53	D54
Данные	60	65	75	79	98

API	Команда			Операнды	Функция	Контроллеры			
70	D	TKY		(S) (D1) (D2)	Ввод с 10-кнопочной клавиатуры	ES2/EX2	SS2	SA2/SE	SX2

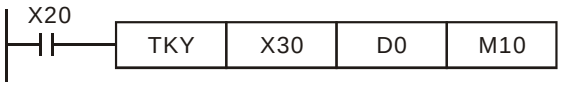
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												TKY: 7 шагов DTKY: 13 шагов
D_1								*	*	*	*	*	*	*	*	
D_2		*	*	*												

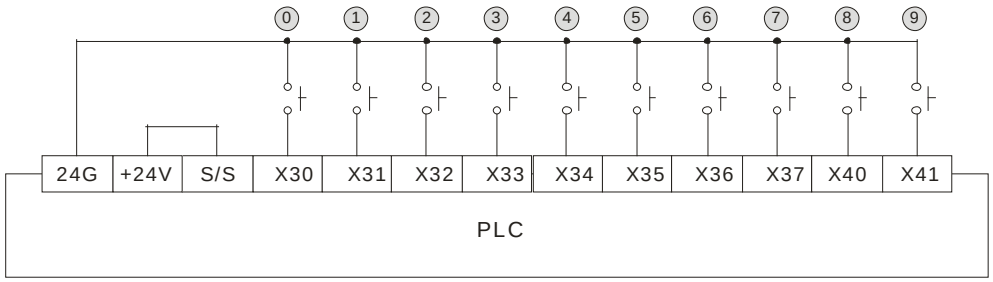
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

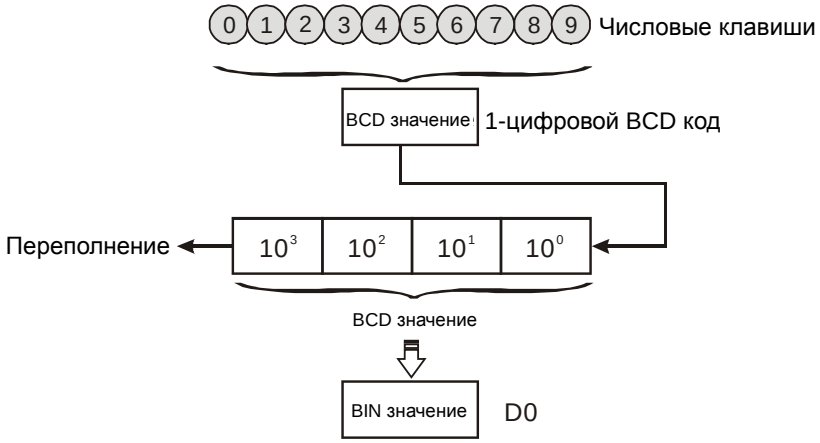
Операнды:
S: Начальный адрес клавиатуры (операнд занимает 10 последовательных адресов) **D_1**: Регистр хранения состояния клавиатуры **D_2**: Выходной сигнал (занимает 11 последовательных адресов)

Описание:
1. Данная команда при подключении к 10-значной клавиатуре определяет 10 входных точек (от 0 до 9) начиная с регистра S. D2 отображает состояние клавиш. Значение с клавиш записывается в D1 - 4 знака (макс. 9999) в 16-битном режиме и 8 знаков (макс. 99999999) в 32-битном режиме.
2. Не существует ограничений на количество использований команды. Тем не менее, одновременно может выполняться только одна команда.

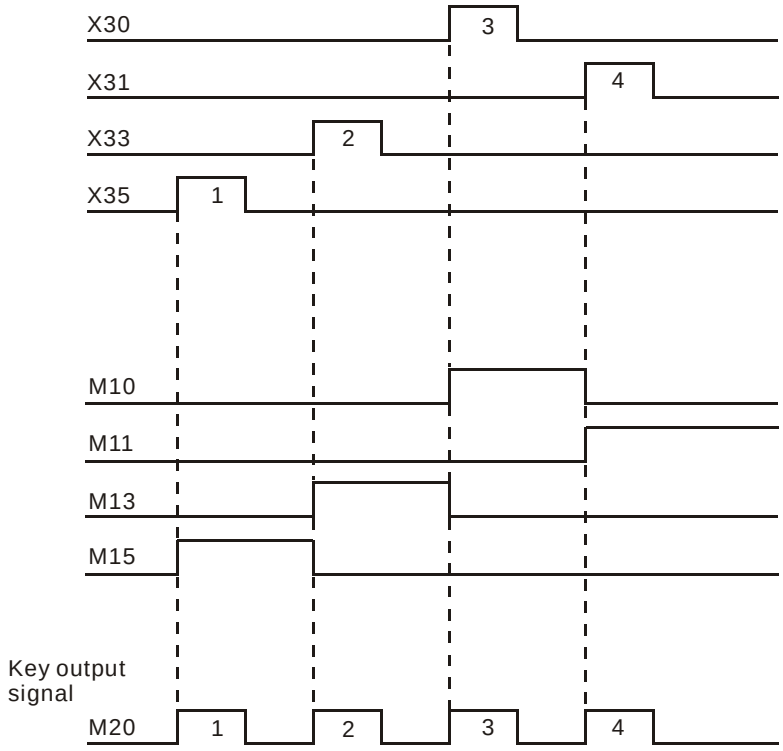
Пример программы:
1. Подключение 10 входов с клавиатуры, начиная с входа X30. Когда X20 включен, команда начнет выполняться, считывая состояние клавиш на входах X30...X41. Значение состояния клавиш будет сохраняться в D0 в двоичном виде. Реле M10 ~ M19 включаются при включении соответсвующих клавиш.







2. Как показано на временной диаграмме, четыре клавиши связаны с входами X35, X33, X30, X31. Нажатие клавиш в таком порядке формирует число 5301 и сохраняет его в D0 (максимально допустимое 9999). Если нажато больше, сохраняются только последние 4 значения.
3. Когда нажата клавиша X35, M15 остается включенным до следующего нажатия клавиши.
4. M20 включается при нажатии любой клавиши.
5. Когда X20 выключается, значение в D0 остается неизменным, а M10-M20 выключаются.



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
71	D	HKY	S D1 D2 D3	Ввод с 16-кнопочной клавиатуры				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															HKY: 9 шагов DHKY: 17 шагов
D_1		*														
D_2											*	*	*	*	*	
D_3		*	*	*												

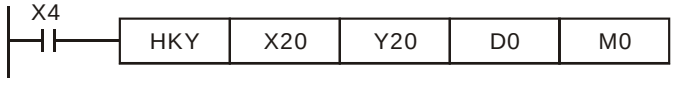
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S: Начальный адрес клавиатуры (занимает 4 последовательных значения) **D_1**: Начальный адрес выхода (занимает 4 последовательных значения) **D_2**: Хранение значений клавиш **D_3**: Состояние клавиш (занимает 8 последовательных значений)

Описание:

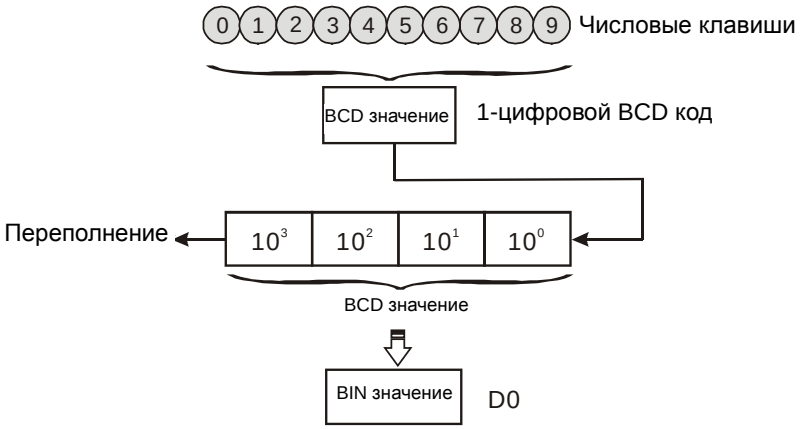
1. Эта команда считывает состояние клавиш 16-кнопочной клавиатуры мультиплексным методом с 4-х входов S и 4-х выходов D1. При сканировании матрицы значения клавиш будут сохраняться в D2. D3 сохраняет состояние клавиш A-F и показывает состояние клавиш 0-9 и A-F.
2. M1029 включается при каждом нажатии клавиши.
3. Если нажато сразу несколько клавиш, вводится только первая.
4. D2 сохраняет десятичное значение 4 знака (0-9999) для 16-битного формата и 8 знаков (0-99999999) для 32-битного формата. Если введено больше значений, обрабатываются последние введенные допустимые значения.
5. Не существует ограничений на количество использований команды, но только одна команда может выполняться во время сканирования.



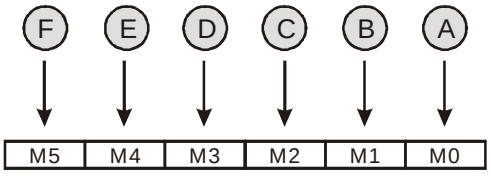
Пример программы:

1. Назначение 4 входов X20 ~ X23 и 4 выходов Y20 ~ Y23 для обработки 16-кнопочной клавиатуры. Когда X4 включен, команда начнет выполняться, и значение клавиш будет сохраняться в D0 в двоичном виде. Состояние клавиш сохраняется в M10-M19.

2. Клавиши 0~9:



3. Клавиши A~F:



a) Когда нажата клавиша A, M0 включится и останется включенным. После нажатия следующей клавиши D: M0 выключится, M3 включится и останется включенным.

b) Если одновременно нажато несколько клавиш, введется значение только первой из них.

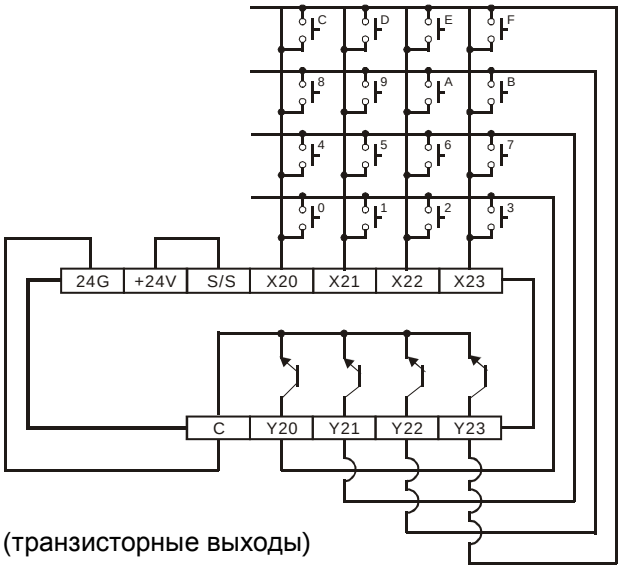
4. Состояние клавиш:

a) Когда любая из клавиш A ~ F нажата, M6 включится на время одного сканирования.

b) Когда любая из клавиш 0 ~ 9 нажата, M7 включится на время одного сканирования.

5. Когда контакт X4 выключен, значение в D0 остается неизменным, а M0-M7 выключатся.

6. Внешнее соединение:



ПЛК (транзисторные выходы)

Заметки:

1. При выполнении команды HKY необходимо 8 циклов сканирования (скан матрицы), обязательных для успешного чтения входных значений. Слишком длинные или слишком короткие циклы могут привести к некорректному чтению данных с клавиатуры. В этом случае возможны следующие решения:
   - Если цикл слишком короткий, то входы/выходы могут не успеть отреагировать и в результате будет неправильный ввод. Для решения этой проблемы необходимо скорректировать время сканирования.
   - Если цикл слишком длинный, реакция на нажатие клавиш может быть медленной. В этом случае необходимо работать с подпрограммой временного прерывания для коррекции времени работы команды.
2. Функции флага M1167:
   - Когда M1167 включен, команда HKY позволяет вводить шестнадцатеричные значения, состоящие из 0-F.
   - Когда M1167 выключен, A~F используются в качестве функциональных клавиш.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
72	DSW	(S) (D1) (D2) (n)	Ввод с цифрового переключателя				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															DSW: 9 шагов
D_1		*														
D_2											*	*	*			
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

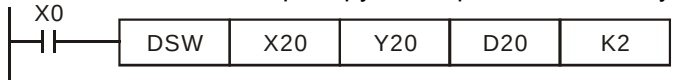
S: Начальный адрес входа **D_1**: Начальный адрес выхода **D_2**: Словный операнд, который содержит считанное значение **n**: Количество групп переключателей (**n** = 1~2)

Описание:

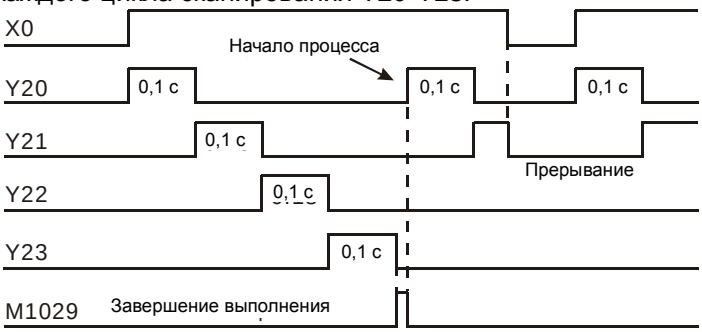
1. Эта команда считывает значение 1 или 2-х групп четырехзместных DIP-переключателей. Команда контролирует 4 или 8 входов, начиная с S, и 4 выхода начиная с D1. Значение сохраняется в D2 и значение n определяет количество групп (1 или 2) DIP-переключателей.
2. **n** = K1, **D_2** занимает 1 регистр. **n** = K2, **D_2** занимает 2 последовательных регистра.
3. Не существует ограничений на количество использований команды, но только одна команда может выполняться во время сканирования.

Пример программы:

1. Первая группа DIP-переключателей состоит из X20-X23 и Y20-Y23, вторая - X24-X27 и Y20-Y23. Когда X0 включен команда считывает значение первой группы переключателей, которое будет переведено в двоичный код и сохранено в D20, двоичное значение второй группы переключателей будет сохранено в D21.

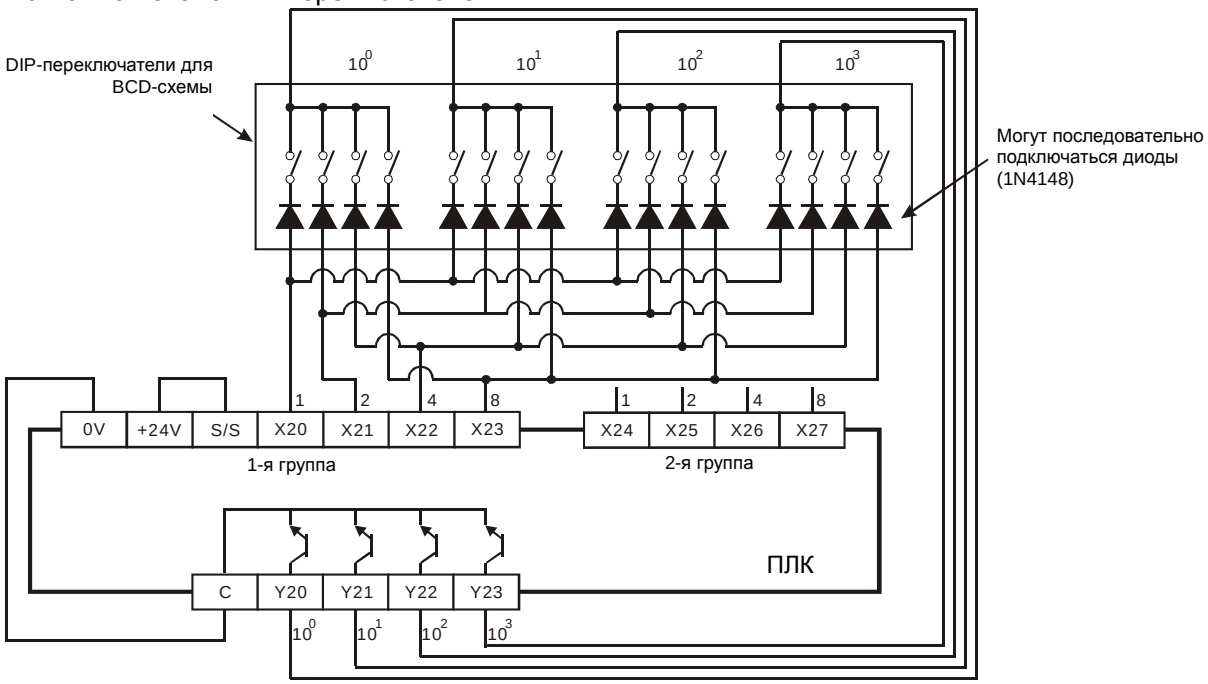


2. Когда X0 включен, выходы Y20...Y23 один за другим обрабатывают состояния соответствующих входов X. Флаг завершения M1029 включается после завершения каждого цикла сканирования Y20-Y23.

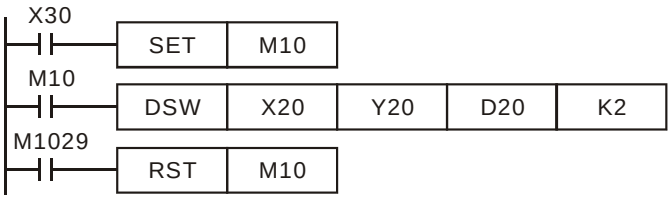


3. Для Y20 ~ Y23 используйте транзисторные выходы. Каждый контакт 1, 2, 4, 8 должен быть подключен к выходам через последовательные диоды (0,1 A/50B).

Монтажная схема DIP-переключателей:



Заметки:



Когда терминалы являются релейными выходами, программа применяется следующим образом: Когда X30 включается, команда начинает выполняться, после выключения X30 M10 остается включенным до завершения текущего цикла сканирования. Если контакт X30 кнопка без фиксации, то M10 выключится только после завершения текущего цикла сканирования. Непрерывный цикл сканирования будет применяться только при нажатии и удержании кнопки, поэтому рекомендуется использовать реле с самоподхватом, что увеличивает срок службы реле.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
73	SEGD	P	(S) (D)	Дешифратор для 7-сегментного индикатора				

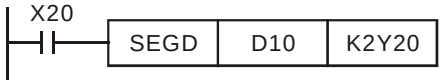
Тип Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	SEGD, SEGDP: 5 шагов
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Исходные данные для дешифровки **D**: Хранение дешифрованных данных

Описание:
Эта команда дешифрует младшие 4 бита (шестнадцатеричные данные: 0 - 9, A - F) из исходных данных **S** и сохраняет дешифрованные данные в младшие 8 бит **D**, формирующие 7-сегментный индикатор.

Пример программы:



Когда X20 включен, содержимое младших 4 бит (b0~b3) D10 дешифруются в формат 7-сегментного индикатора. Результат дешифрования сохраняется в Y20~Y27. Если исходные данные превышают 4 бита, будут дешифрованы только 4 младших бита.

Таблица преобразования сигнала для 7-сегментного индикатора:

HEX значение	Комбинация битов	Структура 7-сегментного индикатора	Состояние каждого сегмента							Данные на дисплее
			B0(a)	B1(b)	B2(c)	B3(d)	B4(e)	B5(f)	B6(g)	
0	0000		ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	
1	0001		ВЫКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	
2	0010		ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	
3	0011		ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	
4	0100		ВЫКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	
5	0101		ВКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВКЛ	
6	0110		ВКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	
7	0111		ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВЫКЛ	
8	1000		ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	
9	1001		ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВКЛ	
A	1010		ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	
B	1011		ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	
C	1100		ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	
D	1101		ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	ВЫКЛ	ВКЛ	
E	1110		ВКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	ВКЛ	
F	1111		ВКЛ	ВЫКЛ	ВЫКЛ	ВЫКЛ	ВКЛ	ВКЛ	ВКЛ	

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
74	SEGL	(S) (D) (n)	7-сегментный индикатор с запоминанием				

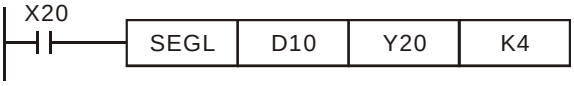
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SEGL: 7 шагов
S					*	*	*	*	*	*	*	*	*	*	*	
D		*														
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Исходные данные для значения, отображаемого 7-сегментным индикатором. **D**: Выходные данные для 7-сегментного индикатора
n: Конфигурация настроек выходного сигнала (**n** = 0~7)

Описание:
1. Эта команда занимает 8 или 12 последовательных выходов, начиная с D, для отображения данных на 1 или 2-х четырехразрядных 7-сегментных индикаторах. Данные преобразуются в BCD-код и выдаются на индикатор. Также осуществляется контроль сигналов запоминания для сохранения отображения данных на 7-сегментном индикаторе.
2. **n** указывает количество четырехразрядных индикаторов (1 или 2) и задает положительную/отрицательную логику выходов ПЛК и параметры индикатора.
3. Один индикатор (группа из 4-х разрядов) занимает 8 адресов, два индикатора (2 группы по 4) занимает 12 адресов.
4. При выполнении команды выходы сканируются по кругу. Если при выполнении команды управляющий контакт выключится и включится вновь, то сканирование возобновляется сначала.
5. Флаг: Когда команда SEGL выполнена, флаг завершения M1029 включается.
6. Не существует ограничений на количество использований команды, но во время сканирования может выполняться только одна команда.

Пример программы:



1. Когда X20 включен, команда SEGL выполняется и выходы Y24-Y27 формируют цепи опроса выходов, формирующих сигнал на 7-сегментный индикатор. Значение D10 преобразовывается в BCD-код и отправляется на 1-й индикатор (набор из 4). Значение D11 от Y30-Y33 преобразовывается и отправляется на 2-й индикатор (набор из 4). Если значение D10, D11 превышает 9999, отображается ошибка.
2. Когда X20 включен, Y24~Y27 сканируются по кругу автоматически. Каждый круг требует 12 циклов сканирования. Флаг завершения M1029 включается в конце каждого цикла, пока круг не будет завершен.
3. Для 1 четырехразрядного 7-сегментного индикатора, **n** = 0 ~ 3:

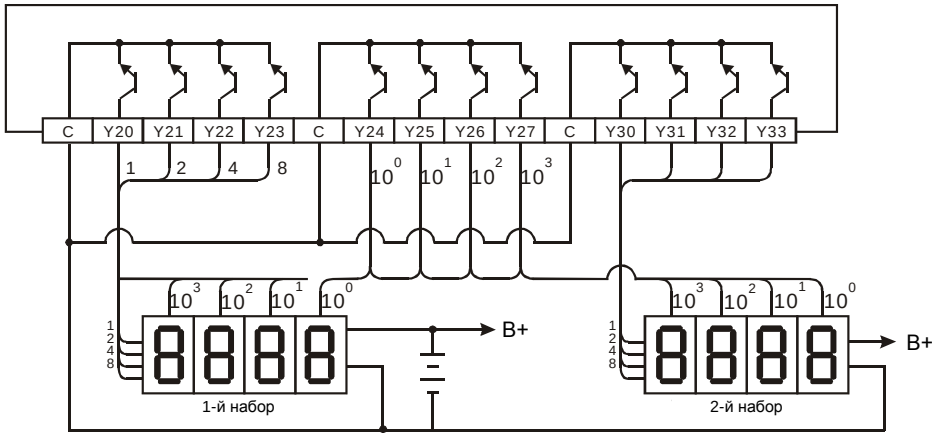
Подключите контакты 1, 2, 4, 8 индикатора параллельно к выходам Y20-Y23 ПЛК. Затем подключите контакты выбора знакоместа к Y24-Y27 ПЛК.

4. Когда X20 включен, содержимое D10 дешифруется и отправляется через выходы Y20-Y23 на 7-сегментный индикатор.
5. Для 2 четырехразрядных 7-сегментных индикаторов, **n** = 4 ~ 7

Подключите контакты 1, 2, 4, 8 индикатора параллельно к выходам Y30-Y33 ПЛК. Затем подключите контакты выбора знакоместа к Y24-Y27 ПЛК.

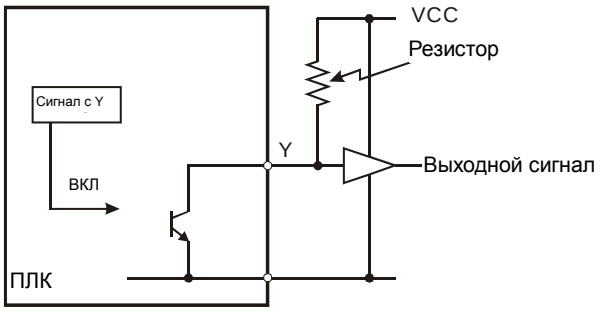
6. Содержимое D10 отправится на первый четырехразрядный индикатор, а D11 - на второй. Если D10=K1234 и D11=K4321, на индикаторах будут отображаться соответственно 1234 и 4321.

Схема соединения:



Заметки:

1. Для выполнения этой команды необходимо время сканирования не меньше 10 мс.
2. ПЛК должен применяться только с транзисторными выходами.
3. Операнд n используется для задания полярности выходного транзистора и определения числа четырехразрядных индикаторов.
4. Выход должен быть транзисторным NPN типа. Выход подключается через нагрузочный резистор к VCC (менее 30 В постоянного тока). Поэтому когда выход Y включается, сигнал имеет низкий (LOW) уровень.



5. Положительная логика (отрицательная полярность) выхода (PNP-выход):

BCD - значение				Выход Y (BCD код)				Выходной сигнал			
b_3	b_2	b_1	b_0	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

6. Отрицательная логика (положительная полярность) выхода (NPN-выход):

BCD - значение				Выход Y (BCD код)				Выходной сигнал			
b_3	b_2	b_1	b_0	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

7. Логические операции выходного сигнала:

Положительная логика (PNP-выход)		Отрицательная логика (NPN-выход)	
Управляющий сигнал (запоминание)	Сигнал управления данными	Управляющий сигнал (запоминание)	Сигнал управления данными
1	0	0	1

8. Параметры **n**:

Число групп 7-сегментных индикаторов	1 группа				2 группы			
BCD-код – сигнал управления данными	+		−		+		−	
Управляющий сигнал (запоминание)	+	−	+	−	+	−	+	−
n	0	1	2	3	4	5	6	7

'+': Положительная логика (PNP-выход); '−': Отрицательная логика (NPN-выход)

9. Полярность выходного транзистора ПЛК и полярность 7-сегментного индикатора могут быть установлены с помощью операнда **n**.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
75	ARWS	S D1 D2 n	Ввод со стрелочной клавиатуры				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												ARWS: 9 шагов
D_1											*	*	*	*	*	
D_2		*														
n					*	*										

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

S: Начальный адрес для ввода данных (занимает 4 последовательных адреса)
D_1: Хранение значения для 7-сегментного индикатора **D_2**: Выход на 7-сегментный индикатор **n**: Конфигурация настроек выходного сигнала (**n** = 0~3). См. команду SEGL.

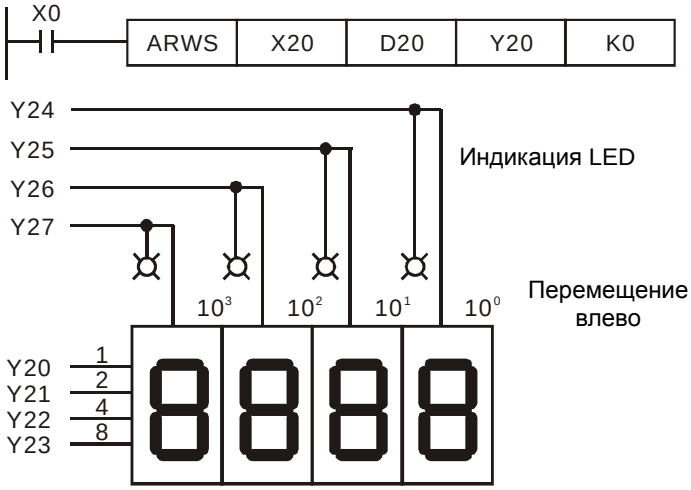
Описание:

1. Команда ARWS отображает значение из D1 на четырехзначном 7-сегментном индикаторе, автоматически преобразуя его BCD-формат. Значение в D1 изменяется с помощью стрелочной клавиатуры, после чего поменяется отображаемая на индикаторе цифра.
2. Значение D2 может быть только кратным 10, т.е. Y0, Y10, Y20…и т.д.
3. ПЛК должен применяться только с транзисторными выходами.
4. При применении этой команды необходимо исправить время сканирования или применить подпрограмму прерывания (I610/I699, I710/I799).
5. Не существует ограничений на количество использований команды, но только одна команда может выполняться одновременно.

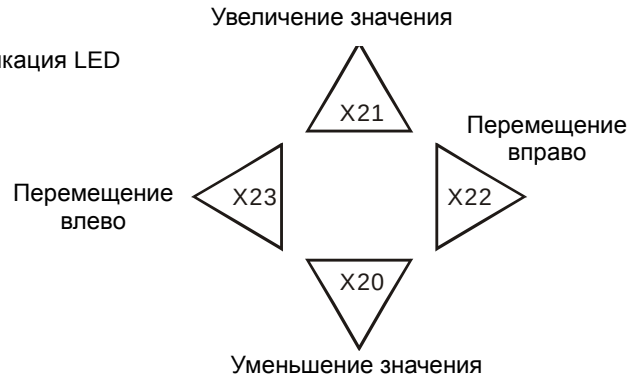
Пример программы:

1. При выполнении команды X20 определяется как счет вниз, X21 - счет вверх, X22 - сдвиг вправо, X23 - сдвиг влево. Клавиши используются для изменения заданных значений (диапазон: 0-9999), хранящихся в D20.

2. Когда X0 включается, цифра 10^3 является стартовым значением. При нажатии клавиши сдвига влево, значение индикатора изменяется в следующей последовательности: $10^3 \rightarrow 10^0 \rightarrow 10^1 \rightarrow 10^2 \rightarrow 10^3 \rightarrow 10^0$.
3. При нажатии клавиши сдвига вправо, значение индикатора изменяется в следующей последовательности: $10^3 \rightarrow 10^2 \rightarrow 10^1 \rightarrow 10^0 \rightarrow 10^3 \rightarrow 10^2$. Кроме того, светящимся светодиодом (LED, Y24 - Y27) будут отмечена текущая позиция курсора (изменяемой ячейки).
4. При нажатии клавиши счета вверх изменение происходит в следующей последовательности: 0 → 1 → 2 … → 8 → 9 → 0 →1. При нажатии клавиши счета вниз изменение происходит в следующей последовательности: 0 → 9 → 8 … → 1 → 0 → 9. Измененное значение будет отображаться на 7-сегментном индикаторе.



7-сегментный четырехразрядный индикатор



4 клавиши для изменения значений, отображаемых на индикаторе

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
76	ASC	(S) (D)	ASCII - конвертирование				

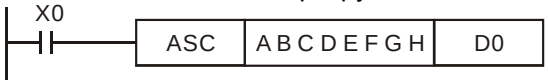
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S																ASC: 11 шагов
D											*	*	*			

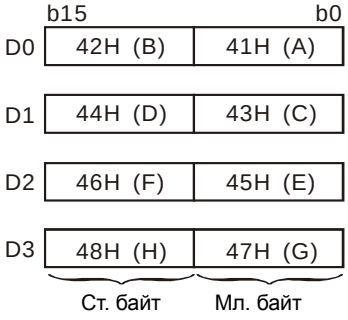
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Буквенно-цифровые данные для конвертирования в ASCII-код **D**: Хранение ASCII-кода

Описание:
1. Команда ASC конвертирует 8 буквенно-цифровых данных, содержащихся в **S,** и сохраняет результат в **D**. Значение в **S** могут быть введены в ПО WPLSoft или ISPSoft.
2. Если ПЛК подключен к 7-сегменному индикатору, то с помощью этой команды возможен текстовый вывод сообщения об ошибке может.
3. Флаг: M1161 переключает 8/16 битовый режим.

Пример программы:
Когда X0 включен, A~H конвертируется в ASCII-код и сохраняется в D0~D3.





Когда M1161 включен, каждый ASCII-код, преобразованный из буквенно-цифрового значения, будет занимать младшие 8 бит (b7-b0) регистра, старшие 8 бит не используются (значение 0).

	b15	b0
D0	00 H	41H (A)
D1	00 H	42H (B)
D2	00 H	43H (C)
D3	00 H	44H (D)
D4	00 H	45H (E)
D5	00 H	46H (F)
D6	00 H	47H (G)
D7	00 H	48H (H)
	Ст. байт	Мл. байт

API	Команда	Операнды	Функция	Контроллеры
77	PR	(S) (D)	Выдача ASCII-кодов по выходам	ES2/EX2 | SS2 | SA2/SE | SX2

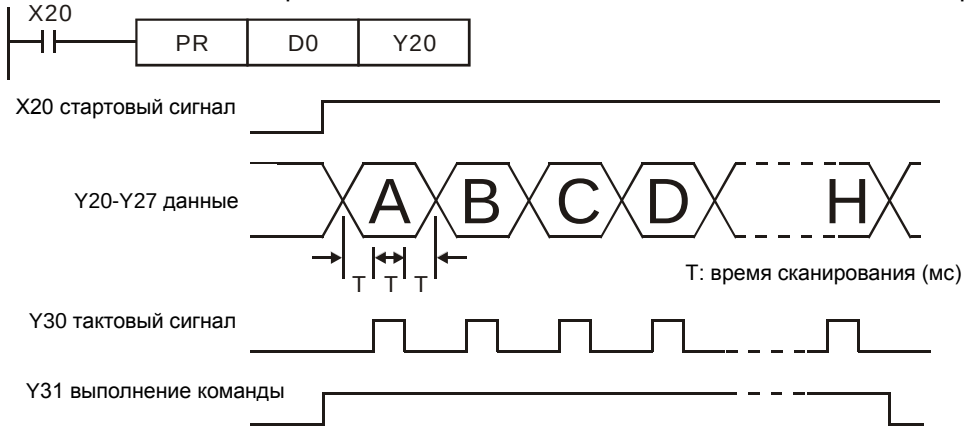
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PR: 5 шагов
S											*	*	*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Хранение ASCII-кодов (занимает 4 последовательных адреса) **D**: Выходы ASCII-кода (занимает 10 последовательных адресов)

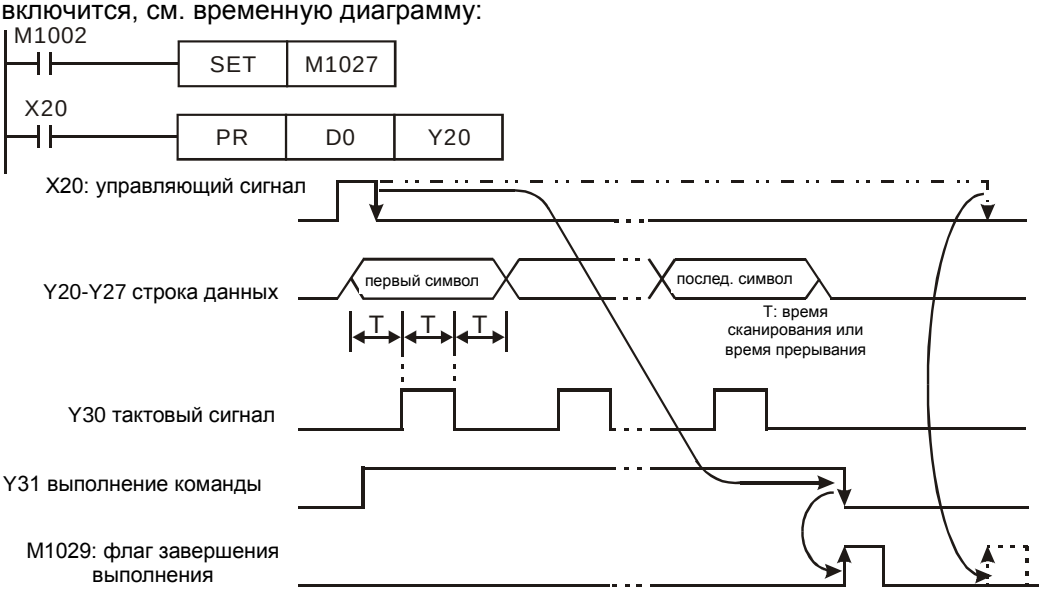
Описание:
1. Эта команда выводит ASCII-коды из 4 регистров, начиная с S, на выходы, начиная с D.
2. Команда может быть использована в программе не более 2 раз.
3. Флаги: M1029 (флаг завершения); M1027 (флаг режима).
4. $\mathbf{D_0}$ ~ $\mathbf{D_7}$ представляют биты сформированные по источнику данных S (ASCII-код), $\mathbf{D_{10}}$ – тактовый сигнал и $\mathbf{D_{11}}$ – флаг завершения.

Пример программы 1:
1. API 76 ASC используется для преобразования A ~ H в ASCII-код, который сохраняется в D0 ~ D3. После этого, команда PR выводит его на выходы по порядку.
2. Когда M1027 выключен и X20 включен, команда назначит в качестве выходов Y20 (младший бит) ~ Y27 (старший бит) и Y30 в качестве тактового, Y31 – флаг выполнения команды. В этом режиме пользователи могут выводить последовательно 8 букв.
3. Если X20 выключается во время выполнения команды, вывод данных прервется, и все выходы отключатся. При новом включении X20 вывод данных начнется с первой буквы.



Пример программы 2:
1. Команда поддерживает 8-битный режим выходных данных, когда M1027 отключен, и 16-битный, когда M1027 включен.
2. Когда M1027 и X20 включены, команда определяет в качестве выходов Y20 (младший бит) ~ Y27 (старший бит) и Y30 в качестве тактового, Y31 – флаг выполнения команды. В этом режиме, пользователи могут выводить последовательно 16 букв.
3. Значение 00H (NULL) в строке данных указывает на конец строки, дальнейшие данные обрабатываться не будут.

4. Если X20 выключается во время выполнения команды, вывод данных прервется, и все выходы отключатся. Если X20 остается включенным, флаг завершения M1029 не включится, см. временную диаграмму:



Заметки:
1. Должен использоваться ПЛК с транзисторными выходами.
2. При применении этой команды необходимо исправить время сканирования или применить подпрограмму прерывания.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
78	D	FROM	P	m_1 m_2 D n	Чтение данных из специального модуля				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m_1					*	*							*			FROM, FROMP: 9 шагов DFROM, DFROMP: 17 шагов
m_2					*	*							*			
D													*			
n					*	*							*			

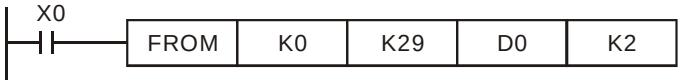
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
m_1: Адрес специального модуля **m_2**: Номер CR регистра для чтения **D**: Начальный адрес регистра хранения данных из m_2 **n**: Число регистров, считываемых одновременно

Описание:
1. ПЛК использует эту команду для чтения данных из CR (контрольного регистра) специального модуля.
2. Диапазон **m_1**: ES2/EX2/SS2: 0 ~ 7; SA2/SE/SX2: 0~107.
3. Диапазон **m_2**: ES2/EX2: 0 ~ 255; SS2: 0~48; SA2/SE/SX2: 0~499.
4. Диапазон **n:**.

Диапазон **n**	ES2/EX2	SS2	SA2/SE/SX2
16-битная команда	1~4	1~(49 - **m_2**)	1~(499 - **m_2**)
32-битная команда	1~2	1~(49 - **m_2**)/2	1~(499 - **m_2**)/2

Пример программы:
1. Чтение данных из CR №29 специального модуля №0 в регистр D0 ПЛК, и CR#30 специального модуля №0 в регистр D1 ПЛК. Одновременно считываются 2 последовательных ряда 16-разрядных данных (**n** = 2).
2. Когда X0 включен, команда выполняется; когда X0 выключен, предыдущее содержимое D0 и D1 не будет изменено.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
79	D	TO	P	(m1) (m2) (S) (n)	Запись данных в специальный модуль				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m_1					*	*							*			TO, TOP: 9 шагов DTO, DTOP: 17 шагов
m_2					*	*							*			
S					*	*							*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

m_1: Адрес специального модуля **m_2**: Номер CR регистра для чтения **S**: Данные для записи в CR **n**: Число регистров, записываемых одновременно

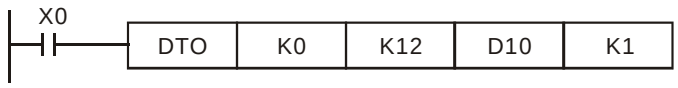
Описание:

1. ПЛК использует эту команду для записи данных в CR (контрольный регистр) специального модуля.
2. Диапазон **m_1**: ES2/EX2/SS2: 0 ~ 7; SA2/SE/SX2: 0~107.
3. Диапазон **m_2**: ES2/EX2: 0 ~ 255; SS2: 0~48; SA2/SE/SX2: 0~499.
4. Диапазон **n**:

Диапазон **n**	ES2/EX2	SS2	SA2/SE/SX2
16-битная команда	1~4	1~(49 - **m_2**)	1~(499 - **m_2**)
32-битная команда	1~2	1~(49 - **m_2**)/2	1~(499 - **m_2**)/2

Пример программы:

1. Применение 32-битной команды DTO позволяет записать содержимое D11 и D10 в CR №13 и CR №12 специального модуля №0. Одновременно записывается один ряд 32-битных данных (n=1).
2. Когда X0 включен, команда выполняется; когда X0 выключен, предыдущее содержимое D10 и D11 не будет изменено.

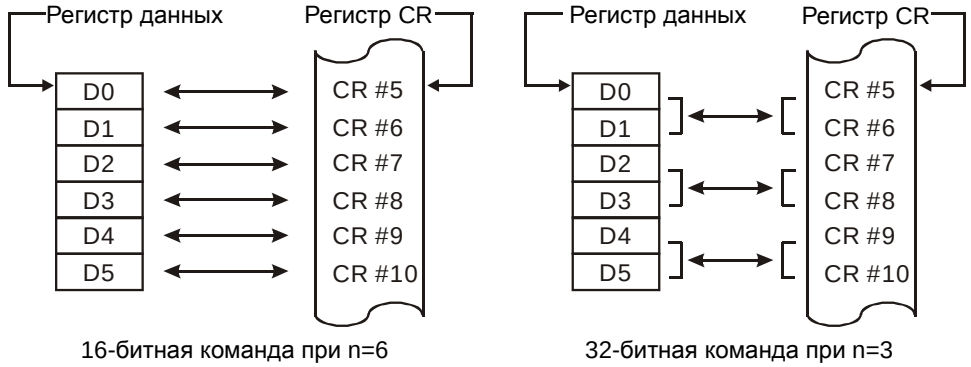


3. Условия для операндов**:**
   m_1: адрес специального модуля. Модули автоматически пронумерованы от 0 (ближайший к MPU) до 7 (самый дальний от MPU). К MPU могут подключаться не более 8 модулей, не занимая цифровых входов/выходов.
   m_2: номер CR (контрольного регистра). CR является 16-разрядной шиной памяти, встроенной в специальный модуль. Все параметры настроек и работы специального модуля заносятся в CR.
4. Команда FROM/TO считывает/записывает 1 CR в одно и то же время. DFROM/DTO считывает/записывает 2 CR в одно и то же время.

   Старшие 16 бит Младшие 16 бит

CR #10	CR #9	← Назначенный номер CR

   n: Число регистров, записываемых одновременно. **n** = 2 в 16-битной операции имеет тот же результат, что и n=1 в 32-битной.



16-битная команда при n=6 32-битная команда при n=3

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
80	RS	S m D n	Последовательная передача данных				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			RS: 9 шагов
m					*	*							*			
D													*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

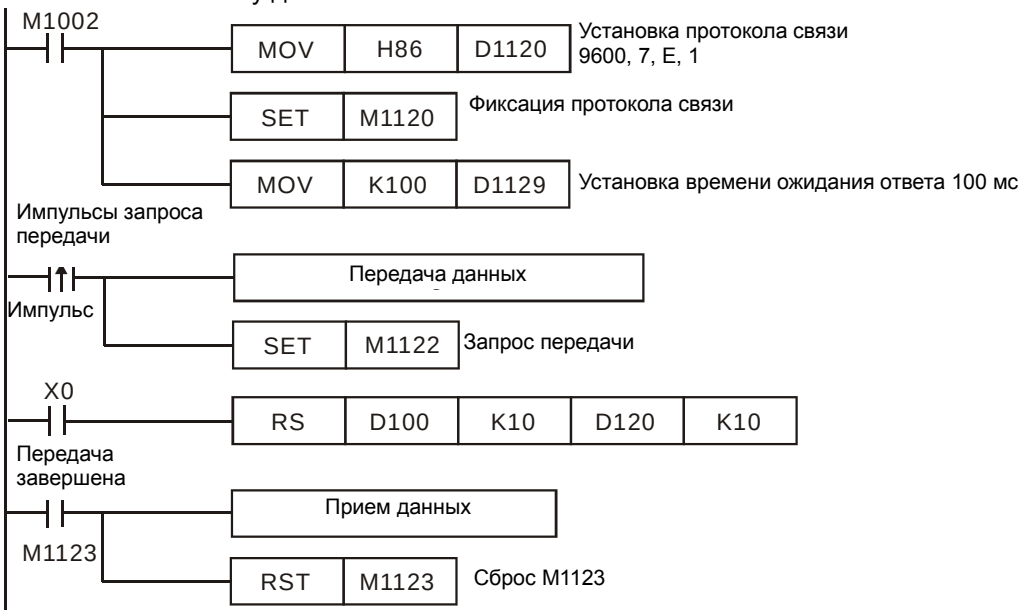
S: Стартовый адрес передаваемых данных **m**: Длина передаваемого сообщения (**m** = 0~256)
D: Стартовый адрес принимаемых данных **n**: Длина принимаемого сообщения (**n** = 0~256)

Описание:

1. Команда используется для обмена данными между ПЛК и внешними устройствами (приводы и т.п.). Пользователь предварительно задает словные данные в стартовых регистрах, начиная с S, длину передаваемого сообщения m, регистры получения данных D, длину получаемого сообщения n.
2. Команда RS поддерживает следующие порты: COM1 (RS-232), COM2 (RS-485) и COM3 (RS-485, ES2/EX2/SA2/SE).
3. Значение **m** равно K0, если данные не нужно передавать. Значение **n** равно K0, если данные не нужно принимать.
4. Изменение данных во время передачи невозможно.
5. Не существует ограничений на количество использований команды. Тем не менее, одновременно может выполняться только одна команда.
6. Для связи с устройствами, поддерживающими протокол Modbus, можно также использовать команды MODRD, MODWR и MODRW.
7. Если подключенными периферийными устройствами является оборудование Delta серии VFD, возможно применение еще нескольких команд для коммуникации: FWD, REV, STOP, RDST и RSTEF.

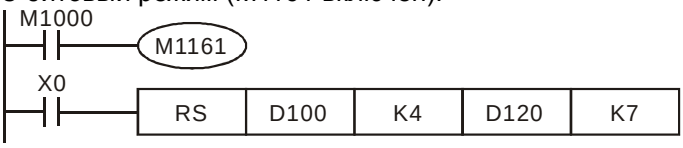
Пример программы 1: COM2 RS-485

1. Запись передаваемых данных в регистры, начиная с D100, и включение флага M1122 (отправка запроса).
2. Когда X10 включен, команда RS выполняется и ПЛК готов к обмену данными. D100 начнет передавать данные последовательно из 10 регистров. После передачи данных M1122 будет автоматически сброшен (не использовать в программе команду RST с M1122). Через 1 мс начнется прием данных, и они будут сохраняться в 10 последовательных регистрах, начиная с D120.
3. После завершения приема данных M1123 будет автоматически сброшен и ПЛК будет снова готов к обмену данными.



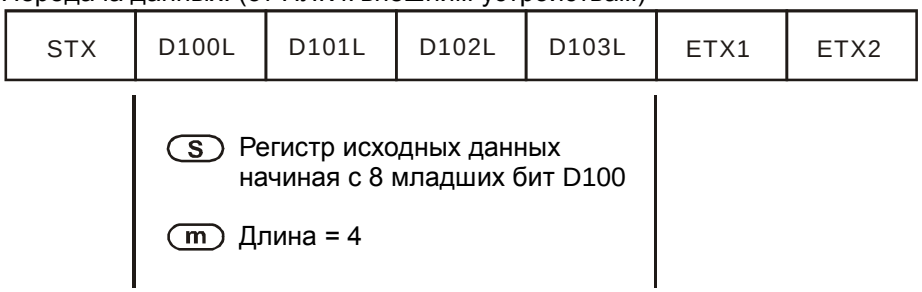
Пример программы 2: COM2 RS-485

8-битовый режим (M1161 включен):

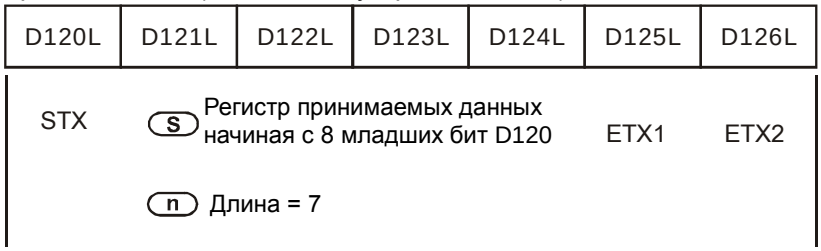


1. STX (стартовый код) и ETX (конечный код) устанавливаются с помощью M1126 и M1130 и назначаются в D1124~D1126. При выполнении команды RS, STX и ETX передаются автоматически.
2. Когда M1161 включен, будет установлен 8-битный режим передачи, а 16-битные данные будут разделены на старшие и младшие. Старшие будут игнорироваться, а младшие – передаваться.

Передача данных: (от ПЛК к внешним устройствам)

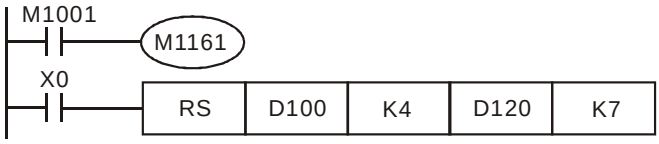


Прием данных: (от внешних устройств к ПЛК)



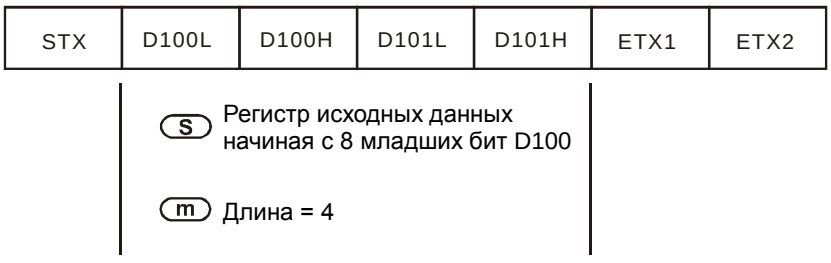
3. ПЛК принимает данные, включая коды STX и ETX, это следует учесть при задании длины принимаемых данных (операнд n).

16-битовый режим (M1161 выключен):

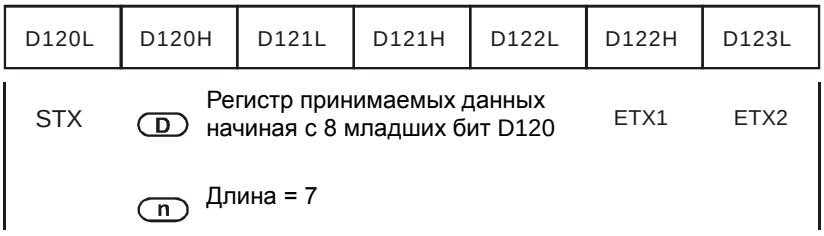


1. STX (стартовый код) и ETX (конечный код) устанавливаются с помощью M1126 и M1130 и назначаются в D1124~D1126. При выполнении команды RS, STX и ETX передаются автоматически.
2. Когда M1161 выключен, будет установлен 16-битный режим передачи, а 16-битные данные будут разделены на старшие и младшие. Старшие и младшие байты будут передаваться.

   Передача данных: (от ПЛК к внешним устройствам)



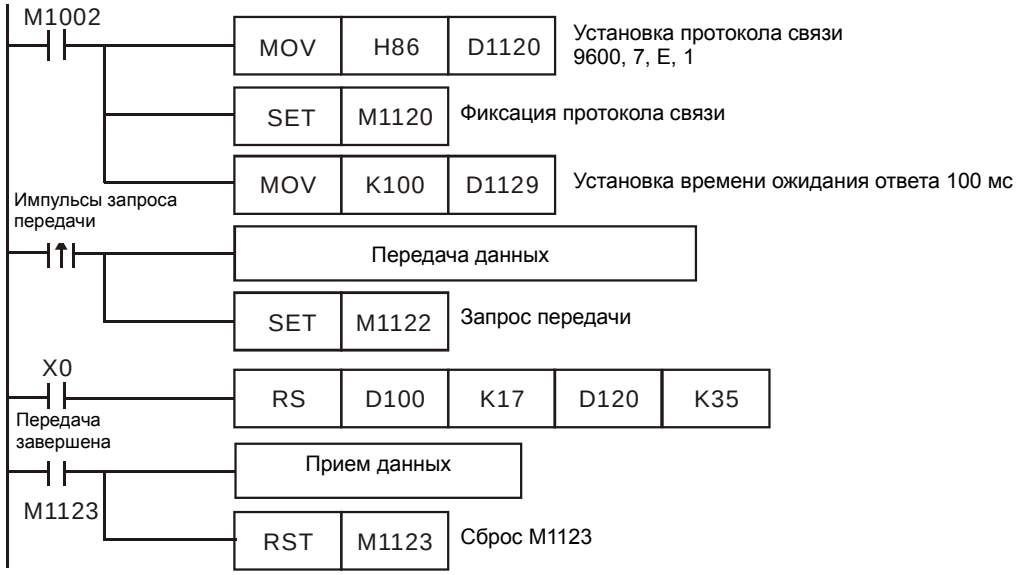
   Прием данных: (от внешних устройств к ПЛК)



3. ПЛК принимает данные, включая коды STX и ETX, это следует учесть при задании длины принимаемых данных (операнд n).

Пример программы 3: COM2 RS-485

1. Обмен данными ПЛК и преобразователя частоты серии VFD-B (ASCII-режим; ПЛК в 16-битном режиме и флаг M1161 выключен).
2. Чтение данных из 6 регистров, начиная с D100 и начальным адресом H2101, о состоянии VFD-B



ПЛК ⇨ VFD-B, ПЛК передает **“: 01 03 2101 0006 D4 CR LF “**
VFD-B ⇨ ПЛК, ПЛК принимает **“: 01 03 0C 0100 1766 0000 0000 0136 0000 3B CR LF “**

Регистры для передачи данных (ПЛК передает сообщение)

Регистр	Данные		Описание	
D100 младший байт	‘: ’	3A H	STX	
D100 старший байт	‘0’	30 H	ADR 1	Адрес: ADR (1,0)
D101 младший байт	‘1’	31 H	ADR 0	
D101 старший байт	‘0’	30 H	CMD 1	Код команды: CMD (1,0)
D102 младший байт	‘3’	33 H	CMD 0	
D102 старший байт	‘2’	32 H	Стартовый адрес данных	
D103 младший байт	‘1’	31 H		
D103 старший байт	‘0’	30 H		
D104 младший байт	‘1’	31 H		
D104 старший байт	‘0’	30 H	Число данных (слов)	
D105 младший байт	‘0’	30 H		
D105 старший байт	‘0’	30 H		
D106 младший байт	‘6’	36 H		
D106 старший байт	‘D’	44 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D107 младший байт	‘4’	34 H	LRC CHK 0	
D107 старший байт	CR	D H	Конец	
D108 младший байт	LF	A H		

Регистры для приема данных (VFD-B передает сообщение)

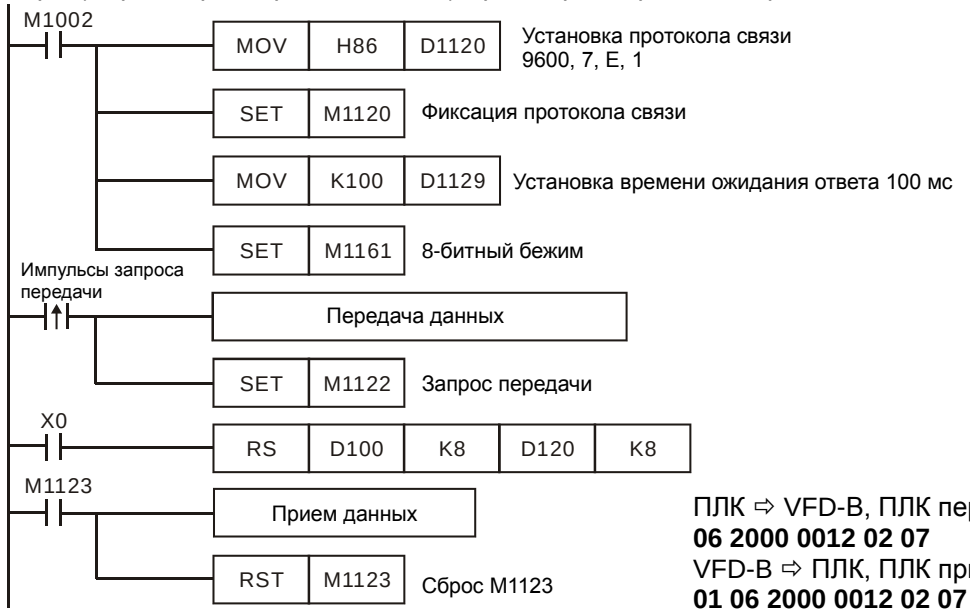
Регистр	Данные		Описание
D120 младший байт	‘: ’	3A H	STX
D120 старший байт	‘0’	30 H	ADR 1
D121 младший байт	‘1’	31 H	ADR 0
D121 старший байт	‘0’	30 H	CMD 1
D122 младший байт	‘3’	33 H	CMD 0
D122 старший байт	‘0’	30 H	Число данных (слов)
D123 младший байт	‘C’	43 H	
D123 старший байт	‘0’	30 H	Содержимое адреса 2101 H

D124 младший байт	‘1’	31 H	
D124 старший байт	‘0’	30 H	
D125 младший байт	‘0’	30 H	
D125 старший байт	‘1’	31 H	Содержимое адреса 2102 H
D126 младший байт	‘7’	37 H	
D126 старший байт	‘6’	36 H	
D127 младший байт	‘6’	36 H	
D127 старший байт	‘0’	30 H	Содержимое адреса 2103 H
D128 младший байт	‘0’	30 H	
D128 старший байт	‘0’	30 H	
D129 младший байт	‘0’	30 H	
D129 старший байт	‘0’	30 H	Содержимое адреса 2104 H
D130 младший байт	‘0’	30 H	
D130 старший байт	‘0’	30 H	
D131 младший байт	‘0’	30 H	
D131 старший байт	‘0’	30 H	Содержимое адреса 2105 H
D132 младший байт	‘1’	31 H	
D132 старший байт	‘3’	33 H	
D133 младший байт	‘6’	36 H	
D133 старший байт	‘0’	30 H	Содержимое адреса 2106 H
D134 младший байт	‘0’	30 H	
D134 старший байт	‘0’	30 H	
D135 младший байт	‘0’	30 H	
D135 старший байт	‘3’	33 H	LRC CHK 1
D136 младший байт	‘B’	42 H	LRC CHK 0
D136 старший байт	CR	D H	Конец
D137 младший байт	LF	A H	

3. Проверить состояние преобразователя частоты Delta серии VDF можно также с помощью команды API 105 RDST.

Пример программы 4: COM2 RS-485

1. Обмен данными ПЛК и преобразователя частоты серии VFD-B (RTU-режим; ПЛК в 16-битном режиме и M1161 включен).
2. Запись данных, которые будут отправлены в регистры, начиная с D100. Запись H12 (пуск вперед) в регистр с адресом H2000 (адрес параметров VFD-B).



ПЛК ⇨ VFD-B, ПЛК передает: **01 06 2000 0012 02 07**
VFD-B ⇨ ПЛК, ПЛК принимает: **01 06 2000 0012 02 07**

Регистры для передачи данных (ПЛК передает сообщение)

Регистр	Данные	Описание
D100 младший байт	01 H	Адрес
D101 младший байт	06 H	Функция
D102 младший байт	20 H	Адрес данных
D103 младший байт	00 H	
D104 младший байт	00 H	Содержание данных
D105 младший байт	12 H	
D106 младший байт	02 H	CRC CHK Low
D107 младший байт	07 H	CRC CHK High

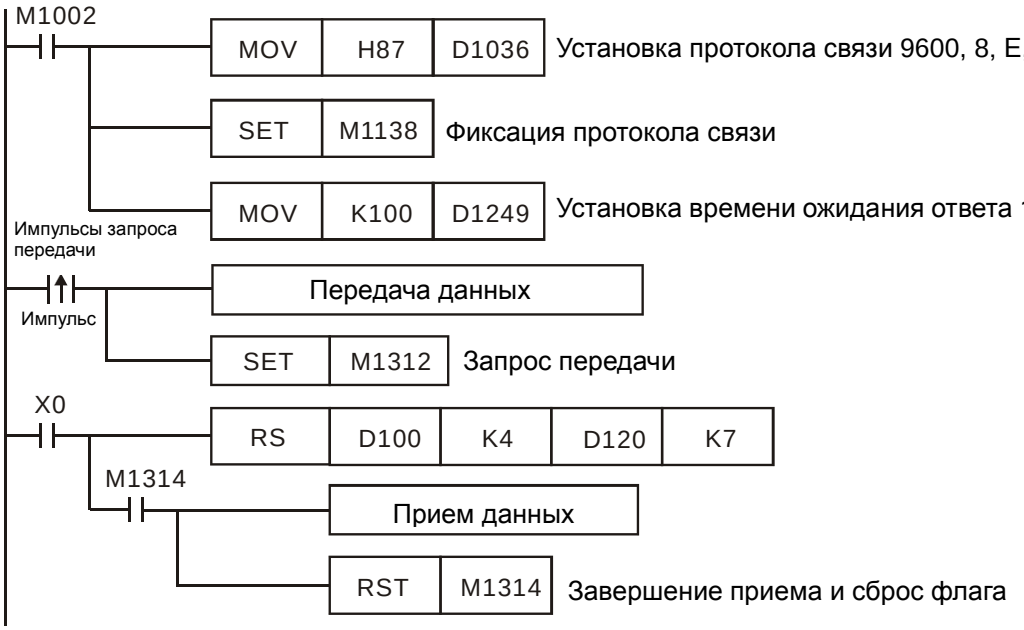
Регистры для приема данных (VFD-B передает сообщение)

Регистр	Данные	Описание
D120 младший байт	01 H	Адрес
D121 младший байт	06 H	Функция
D122 младший байт	20 H	Адрес данных
D123 младший байт	00 H	
D124 младший байт	00 H	Содержание данных
D125 младший байт	12 H	
D126 младший байт	02 H	CRC CHK Low
D127 младший байт	07 H	CRC CHK High

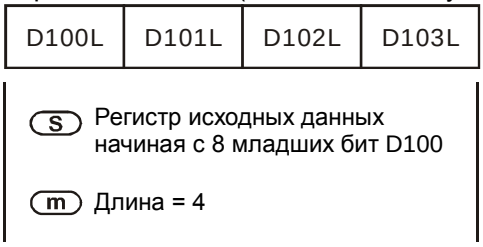
3. Реализовать функцию пуска вперед преобразователя частоты Delta серии VDF можно также с помощью команды API 102 FWD.

Пример программы 5: COM1 RS-232

1. Поддерживается только 8-битовый режим. Формат связи и скорость передачи данных определяются младшими 8 байтами D1036.
2. Настройка функций STX/ETX (M1126/M1130/D1124~D1126) не поддерживается.
3. Старшие байты 16-битных данных недоступны. Для связи используются только младшие байты.
4. Запись передаваемых данных в регистры, начиная с D100, и включение флага M1312 (отправка запроса COM1).
5. Когда X10 включен, команда RS выполняется, и ПЛК готов к обмену данными. D0 начнет передавать данные из 4 регистров непрерывно. После передачи данных M1312 будет автоматически сброшен (не использовать в программе команду RST M1312). Через 1 мс начнется прием данных, и они будут сохраняться в 7 последовательных регистрах, начиная с D20.
6. После завершения приема данных M1314 автоматически включится. После обработки полученных данных M1314 будет сброшен и ПЛК снова готов к обмену данными.



Передача данных: (ПЛК→Внешнее устройство)



Прием данных: (Внешнее устройство→ПЛК)

D120L	D121L	D122L	D123L	D124L	D125L	D126L

(S) Регистр принимаемых данных начиная с 8 младших бит D120

(n) Длина = 7

Пример программы 6: COM3 RS-485

1. Поддерживается только 8-битовый режим. Формат связи и скорость передачи данных определяются младшими 8 байтами D1109.
2. Настройка функций STX/ETX (M1126/M1130/D1124~D1126) не поддерживается.
3. Старшие байты 16-битных данных недоступны. Для связи используются только младшие байты.
4. Запись передаваемых данных в регистры, начиная с D100 и включение флага M1316 (отправка запроса COM3).
5. Когда X10 включен, команда RS выполняется, и ПЛК готов к обмену данными. D0 начнет передавать данные из 4 регистров непрерывно. После передачи данных M1318 будет автоматически сброшен (не использовать в программе команду RST M1318). Через 1 мс начнется прием данных, и они будут сохраняться в 7 последовательных регистрах, начиная с D20.
6. После завершения приема данных M1318 автоматически включится. После обработки полученных данных M1318 будет сброшен и ПЛК снова готов к обмену данными.



Передача данных: (ПЛК→Внешнее устройство)

D100L	D101L	D102L	D103L

(S) Регистр исходных данных начиная с 8 младших бит D100

(m) Длина = 4

Прием данных: (Внешнее устройство→ПЛК)

D120L	D121L	D122L	D123L	D124L	D125L	D126L

(S) Регистр принимаемых данных начиная с 8 младших бит D120

(n) Длина = 7

Заметки:

PLC COM1 RS-232: Флаги (специальные реле) и специальные регистры (Special D) для команд связи RS/MODR:

Флаг	Функция	Действие
M1138	COM1 сохраняет параметры соединения. Настройки связи будут сброшены (изменены) в зависимости от содержимого D1036 после каждого цикла. Пользователь может включить M1138, если требуется сохранить протокол связи, в этом случае параметры связи изменены не будут даже в случае изменения содержания D1036. Поддерживаемые команды связи: RS / MODRW	Установка и сброс пользователем
M1139	COM1 ASCII / RTU: выбор режима, ON: RTU-режим, OFF: ASCII- режим. Поддерживаемые команды связи: RS / MODRW	Установка и сброс пользователем
M1312	Флаг отправки запроса. Перед выполнением команды связи пользователь должен включить M1312 для запуска процедуры обмена данными. После завершения обмена ПЛК автоматически сбрасывает M1312. Поддерживаемые команды связи: RS / MODRW	Установка пользователем и сброс системой
M1313	Флаг готовности к приему данных. Когда M1313 включен, ПЛК готов к приему данных. Поддерживаемые команды связи: RS / MODRW	Установка и сброс системой
M1314	Флаг завершения приема данных. Пользователь может обрабатывать данные, пока флаг M1314 включен, после чего пользователь должен сбросить этот флаг для нового сеанса связи. Поддерживаемые команды связи: RS / MODRW	Установка системой и сброс пользователем
M1315	Флаг ошибки. Флаг включится при возникновении ошибки и код ошибки сохраняется в D1250. Поддерживаемые команды связи: RS / MODRW	Установка системой и сброс пользователем

Специальный регистр	Функция
D1036	COM1 (RS-232) протокол связи. См. пункт 4 следующей таблицы для настройки протокола.
D1167	Специальное слово для выполнения запроса прерывания (I140) на COM1 (RS232). Поддерживаемые команды связи: RS
D1121	Адреса связи COM1 (RS-232) и COM2 (RS-485)
D1249	COM1 (RS-232) Настройка ожидания соединения (мс). Если пользователь устанавливает значение ожидания в D1249 и время получения данных превышает время ожидания, включится M1315 и код ошибки K1 будет сохранен в D1250. M1315 сбрасывается пользователем.
D1250	COM1 (RS-232) код ошибки. Поддерживаемые команды связи: MODRW

PLC COM2 RS-485: Флаги (специальные реле) и специальные регистры (Special D) для команд связи RS / MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW.

Флаг	Функция	Действие
M1120	Сохранение прокола связи. Если 1120 выключен, то протокол, установленный в D1120, будет сбрасываться по окончании каждого цикла. Если M1120 включен, заданный протокол связи сохраняется	Установка и сброс пользователем
M1121	Готовность передачи данных. Если M1121 выключен, идет передача данных по RS-485.	Установка и сброс системой
M1122	Запрос передачи данных. Для начала обмена данными пользователь должен включить M1122 (импульс). M1122 будет автоматически сброшен после завершения передачи данных.	Установка пользователем и сброс системой
M1123	Завершение приема данных. M1123 автоматически включится после завершения приема данных. Сброс M1123 осуществляется пользователем. Поддерживаемые команды связи: RS	Установка системой и сброс пользователем
M1124	Ожидание приема данных. Когда M1124 включен, ПЛК ожидает приема данных.	Установка и сброс системой
M1125	Сброс связи. Когда M1125 включен, соединение будет сброшено, после чего пользователь выключает M1125	Установка и сброс пользователем
M1126	Выбор STX/ETX (пользователь/система). См. Пункт 5 следующей таблицы. M1126 поддерживается только командой RS.	
M1130	Выбор STX/ETX (пользователь/система). См. Пункт 5 следующей таблицы. M1126 поддерживается только командой RS.	
M1127	COM2 (RS-485) Завершение приема данных. Команда RS не поддерживается. Поддерживаемые команды связи: MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW	Установка системой и сброс пользователем
M1128	Индикация приема/передачи	Установка и сброс системой

Флаг	Функция	Действие
M1129	Превышение времени ожидания приема. M1129 включится, если время ожидания превысит значение, установленное в D1129.	Установка системой и сброс пользователем
M1131	В ASCII-режиме M1131 включен, только когда команды MODRD/RDST/MODRW конвертируют данные в шестнадцатеричный формат. Поддерживаемые команды связи: MODRD / RDST / MODRW	Установка и сброс системой
M1140	MODRD/MODWR/MODRW Ошибка приема данных Поддерживаемые команды связи: MODRD / MODWR / MODRW	
M1141	MODRD/MODWR/MODRW Ошибка команд Поддерживаемые команды связи: MODRD / MODWR/ MODRW	
M1142	Ошибка приема данных от VFD-A Поддерживаемые команды связи: FWD / REV / STOP / RDST / RSTEF	
M1143	Выбор режима ASCII / RTU. ON: RT-режим, OFF: ASCII-режим. Поддерживаемые команды связи: RS / MODRD / MODWR / MODRW (При M1177 = ВКЛ. также поддерживаются FWD / REV / STOP / RDST / RSTEF.	Установка и сброс пользователем
M1161	Режим 8/16-бит. ON: 8-битовый режим. OFF: 16-битовый режим Поддерживаемые команды связи: RS	Установка и сброс пользователем
M1177	Запуск команды связи для оборудования Delta серии VFD. ON: VFD-A (по умолчанию), OFF: остальные модели VFD Поддерживаемые команды связи: FWD / REV / STOP / RDST / RSTEF	

Специальный регистр	Функция
D1038	Время задержки ответа на COM2, COM3 RS-485, когда ПЛК находится в ведомом (slave) режиме, Диапазон: 0~10,000. (ед. 0,1 мс). Для использования EASY PLC LINK в COM2, D1038 может устанавливаться для следующей передачи данных с задержкой (ед. 1 цикл)
D1050 ~ D1055	Конвертирование данных для передачи данных по протоколу Modbus. ПЛК автоматически конвертирует ASCII данные из D1070~D1085 в шестнадцатеричные 16-битовые данные и сохраняет их в D1050~D1055 Поддерживаемые команды связи: MODRD / RDST
D1070 ~ D1085	Обратная связь (ASCII) по протоколу Modbus. Когда ПЛК получает сигнал обратной связи, данные будут сохранены в регистрах D1070-D1085 и затем преобразуются в шестнадцатеричный формат в других регистрах. Команда RS не поддерживается.
D1089 ~ D1099	Переданные данные по Modbus. Когда команда связи (MODRD) пересылает данные, они сохраняются в D1089~D1099. Команда RS не поддерживается.
D1120	COM2 (RS-485) Протокол связи. См. Пункт 4 следующей таблицы.
D1121	COM1 (RS-232) и COM2 (RS-485) Адрес связи ПЛК в ведомом (slave) состоянии.
D1122	COM2 (RS-485) Остаточные слова переданных данных.
D1123	COM2 (RS-485) Остаточные слова принятых данных.
D1124	COM2 (RS-485) Определение стартового слова (STX) См. Пункт 3 следующей таблицы. Поддерживаемые команды связи: RS
D1125	COM2 (RS-485) Определение стопового слова 1 (ETX1) См. Пункт 3 следующей таблицы. Поддерживаемые команды связи: RS
D1126	COM2 (RS-485) Определение стопового слова 2 (ETX2) См. Пункт 3 следующей таблицы. Поддерживаемые команды связи: RS
D1129	COM2 (RS-485) Настройка времени ожидания (мс). Если время получения данных превышает установленное пользователем в D1129 время ожидания, M1129 включится и код ошибки K1 будет сохранен в D1130. M1129 сбрасывается вручную после очищения статуса ожидания.
D1130	COM2 (RS-485) Код ошибки отклика Modbus. Команда RS не поддерживается Поддерживаемые команды связи: MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF / MODRW
D1168	Специальное слово для выполнения операции прерывания (I150) на COM2 (RS-485). Поддерживаемые команды связи: RS
D1256 ~ D1295	Для COM2 RS-485 команды MODRW. D1256~D1295 регистры для хранения переданных/принятых данных. Поддерживаемые команды связи: MODRW
D1296 ~ D1311	Для COM2 RS-485 команды MODRW. D1296~D1311 сохраняют конвертированные в шестнадцатеричный формат данные из D1070 ~ D1085 (ASCII). Поддерживаемые команды связи: MODRW

PLC COM3 RS-485: Флаги (специальные реле) и специальные регистры (Special D) для команд связи RS / MODRW and FWD / REV / STOP / RDST / RSTEF when M1177 = ON.

Флаг	Функция	Действие
M1136	COM3 Сохранение параметров связи. Настройки связи будут сброшены (изменены) в зависимости от содержимого D1109 после каждого цикла. Пользователь может включить M1136, если требуется сохранить протокол связи, в этом случае параметры связи изменены не будут даже в случае изменения содержания D1109.	Установка и сброс пользователем

Флаг	Функция	Действие
M1320	COM3 ASCII / RTU Выбор режима. ON: RTU-режим, OFF: ASCII-режим.	
M1316	COM3 Запрос передачи данных. Перед выполнением команды связи пользователь должен включить M1316 (импульс) для запуска процедуры обмена данными. После завершения обмена ПЛК автоматически сбрасывает M1316.	Установка пользователем и сброс системой
M1317	Готовность приема данных. Когда M1317 включен, ПЛК готов принимать данные.	Установка и сброс системой
M1318	COM3 Завершение приема данных.	Установка системой и сброс пользователем
M1319	COM3 Ошибка приема данных. Флаг включится при возникновении ошибки и код ошибки сохраняется в D1252.	Установка системой и сброс пользователем

Специальный регистр	Функция
D1038	Время задержки ответа на COM2, COM3 RS-485, когда ПЛК находится в ведомом (slave) режиме, Диапазон: 0~10,000. (ед. 0,1 мс). Для использования EASY PLC LINK в COM2, D1038 может устанавливаться для следующей передачи данных с задержкой (ед. 1 цикл)
D1109	COM3 (RS-485) протокол связи. См. Пункт 4 следующей таблицы
D1169	Специальное слово для выполнения запроса прерывания (I160) на COM3 (RS-485). Поддерживаемые команды связи: RS
D1252	COM3 (RS-485) Настройка времени ожидания (мс). Если время получения данных превышает установленное пользователем в D1252 время ожидания, M1319 включится и код ошибки K1 будет сохранен в D1253. M1319 сбрасывается вручную после очищения статуса ожидания.
D1253	COM3 (RS-485) Код ошибки связи.
D1255	COM3 (RS-485) Адрес связи ПЛК в ведомом (slave) режиме.

Таблица соответствия между COM-портами и параметрами/настройками связи

	COM1	COM2	COM3	Описание функций
Настройки протокола	M1138	M1120	M1136	Сохранение параметров связи.
	M1139	M1143	M1320	ASCII/RTU выбор режима
	D1036	D1120	D1109	Протокол связи
	D1121	D1121	D1255	Адрес связи ПЛК
Запрос передачи данных	-	M1161	-	Выбор 8/16-бит режимов
	-	M1121	-	Индикация статуса передачи
	M1312	M1122	M1316	Запрос передачи данных
	-	M1126	-	Выбор STX/ETX пользователь/система (RS)
	-	M1130	-	Выбор STX/ETX пользователь/система (RS)
	-	D1124	-	STX (RS)
	-	D1125	-	ETX1 (RS)
	-	D1126	-	ETX2 (RS)
	D1249	D1129	D1252	Настройка времени ожидания (мс)
	-	D1122	-	Остаточные слова переданных данных
	-	D1256 ~ D1295	-	Хранение переданных данных (команда MODRW)
	-	D1089 ~ D1099	-	Хранение переданных данных (команды MODRD / MODWR / FWD / REV / STOP / RDST / RSTEF)
Получение данных	M1313	M1124	M1317	Готовность приема данных
	-	M1125	-	Сброс связи
	-	M1128	-	Индикация приема/передачи
	-	D1123	-	Остаточные слова полученных данных
	-	D1070 ~ D1085	-	Хранение данных обратной связи Modbus. Команда RS не поддерживается.
	D1167	D1168	D1169	Специальное слово управления прерыванием I140/I150/I160 (RS)
Получение данных завершено	M1314	M1123	M1318	Завершение приема данных
	-	M1127	-	COM2 (RS-485) прием/передача/преобразование данных выполнено. (Команда RS не поддерживается)
	-	M1131	-	Включается, когда команды MODRD/RDST/MODRW преобразуют данные из ASCII в 16-ный формат
	-	D1296 ~ D1311	-	Сохранение преобразованных командой MODRW данных.
	-	D1050 ~ D1055	-	Сохранение преобразованных командой MODRD данных

	COM1	COM2	COM3	Описание функций
Ошибки	M1315	-	M1319	Ошибка получения данных
	D1250	-	D1253	Код ошибки связи
	-	M1129	-	COM2 (RS-485) ожидание получения
	-	M1140	-	COM2 (RS-485) MODRD/MODWR/MODRW. Ошибка получения данных
	-	M1141	-	MODRD/MODWR/MODRW параметры ошибки (код находится в полученных данных). Код сохраняется в D1130
	-	M1142	-	Ошибка получения данных от VFD-A (FWD/REV/STOP/RDST/RSTEF)
	-	D1130	-	COM2 (RS-485) Код ошибки в Modbus

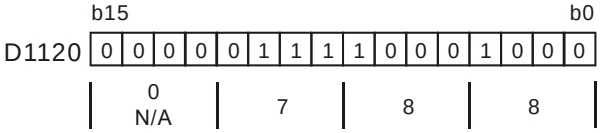
Настройки протокола связи: D1036(COM1 RS-232) / D1120(COM2 RS-485) / D1109(COM3 RS-485)

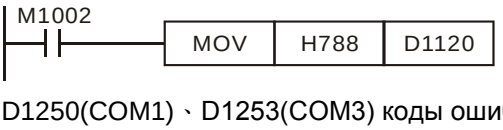
	Содержание		
b0	Длина данных	**0:** 7 бит данных	**1:** 8 бит данных
b1 b2	Паритет	00: нет паритета 01: паритет по нечету 11: паритет по чету	
b3	Стоповый бит	**0:** 1 бит	**1:** 2 битов
b4 b5 b6 b7	Скорость (бит/с)	0001(H1):110 bps 0010(H2): 150 bps 0011(H3): 300 bps 0100(H4): 600 bps 0101(H5): 1200 bps 0110(H6): 2400 bps 0111(H7): 4800 bps 1000(H8): 9600 bps 1001(H9): 19200 bps 1010(HA): 38400 bps 1011(HB): 57600 bps 1100(HC): 115200 bps 1101(HD): 500000 bps (COM2 / COM3) 1110 (HE): 31250 bps (COM2 / COM3) 1111 (HF): 921000 bps (COM2 / COM3)	
b8 (D1120)	STX	0: нет	1: D1124
b9 (D1120)	ETX1	0: нет	1: D1125
b10 (D1120)	ETX2	0: нет	1: D1126
b11~b15	Не определено		

Когда команда RS применяется для связи ПЛК с периферийным устройством по COM2 RS-485, STX(начало текста) и ETX(конец текста) должны быть установлены в формате связи. В этом случае b8-b10 устанавливаются в 1, так что пользователь может установить STX/ETX как управление пользователем/системой, что определяется с помощью M1126, M1130 и D1124-D1126. Установка M1126 и M1130:

		M1130	
		0	1
M1126	0	D1124: определено пользователем D1125: определено пользователем D1126: определено пользователем	D1124: H 0002 D1125: H 0003 D1126: H 0000（no setting）
	1	D1124: определено пользователем D1125: определено пользователем D1126: определено пользователем	D1124: H 003A（':'） D1125: H 000D（CR） D1126: H 000A（LF）

Пример установки формата связи D1120:
Формат связи:
Скорость: 9600, 7, N, 2
STX : ": "
ETX1 : "CR"
ETX2 : "LF"
В D1120 должно быть записано значение H788:





Когда применяются STX, ETX1 и ETX2, следует иметь в виду состояние M1126 и M1130.

D1250(COM1)、D1253(COM3) коды ошибки связи:

Значение	Описание ошибки

Значение	Описание ошибки
H0001	Ожидание связи
H0002	Ошибка контрольной суммы
H0003 H0004 H0005	Исключительная ситуация Ошибка командного кода / ошибка данных Ошибка длины данных

Таблица соотношений D1167~D1169 с указателями прерывания (только младшие 8 бит)

COM-порт	I1_0 прерыв.	Специальный регистр D
COM1	I140	D1167
COM2	I150	D1168
COM3	I160	D1169

Возьмем стандартный формат MODBUS для примера:

ASCII-режим

Наименование	Описание
STX	Стартовый символ = ‘: ’ (3AH)
Address Hi Address Lo	Коммуникационный адрес: 8-битовый адрес состоит из 2 ASCII-кодов
Function Hi Function Lo	Функциональный код: 8-битовый код состоит из 2 ASCII-кодов
DATA (n-1) ……. DATA 0	Содержание данных: n × 8-битовые данные состоят из 2n ASCII-кодов
LRC CHK Hi LRC CHK Lo	LRC контрольная сумма: 8-битовая контрольная сумма состоит из 2 ASCII-кодов
END Hi END Lo	Конечный символ: END Hi = CR (0DH), END Lo = LF(0AH)

Протокол связи Modbus работает в режиме ASCII, т.е. каждый бит состоит из 2 ASCII символов. Например, 64Hex это "64" в ASCII, состоящий из "6" (36Hex) и "4" (34Hex). Каждому символу 0-9, A-F также соответствует свой ASCII-код.

Значение	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’
ASCII-код	30H	31H	32H	33H	34H	35H	36H	37H

Значение	‘8’	‘9’	‘A’	‘B’	‘C’	‘D’	‘E’	‘F’
ASCII-код	38H	39H	41H	42H	43H	44H	45H	46H

Стартовое слово (STX): ‘: ’ (3AH)

Адрес:
‘0’ ‘0’: Передача на все устройства
‘0’ ‘1’: на устройство по адресу 01
‘0’ ‘F’: на устройство по адресу 15
‘1’ ‘0’: на устройство по адресу 16
… и т.д., макс. адрес: 254 (‘FE’)

Код команды:
‘0’ ‘3’: чтение содержимого из нескольких регистров
‘0’ ‘6’: занесение одного слова в конкретный регистр
‘1’ ‘0’: занесение содержимого в несколько регистров

Символы данных: Данные, передаваемые пользователю
LRC контрольная сумма: LCR контрольная сумма представляет собой 2 дополнений отрицания суммы, добавляемого в адрес символов данных.
Пример: 01H + 03H + 21H + 02H + 00H + 02H = 29H 2 дополнений отрицания 29H = D7H.
Конечное слово (END): Зафиксируем END как END Hi = CR (0DH), END Lo = LF (0AH)

Пример: Чтение 2 непрерывных слов данных, хранящихся в регистрах устройства по адресу 01H (см. таблицу ниже). Стартовый регистр по адресу 2102H.

Запрос:

STX	‘: ’
Адрес	‘0’
	‘1’
Код команды	‘0’
	‘3’
Стартовый адрес	‘2’
	‘1’
	‘0’
	‘2’
Размер данных (по словам)	‘0’
	‘0’
	‘0’
	‘2’
LRC Контрольная сумма	‘D’
	‘7’
END	CR
	LF

Ответ:

STX	‘: ’
Адрес	‘0’
	‘1’
Код команды	‘0’
	‘3’
Размер данных (по словам)	‘0’
	‘4’
Содержимое стартового адреса 2102H	‘1’
	‘7’
	‘7’
	‘0’
Содержимое адреса 2103H	‘0’
	‘0’
	‘0’
	‘0’
LRC Контрольная сумма	‘7’
	‘1’
END	CR
	LF

RTU- режим

Наименование	Описание

Наименование	Описание
START	См. следующее описание
Address	Коммуникационный адрес: n 8-бит двоичный
Function	Код команды: n 8-бит двоичный
DATA (n-1)	Данные: n × 8-бит данные
…….	
DATA 0	
CRC CHK Low	CRC контрольная сумма: 16-бит CRC состоит из 2 8-бит двоичных данных
CRC CHK High	
END	См. следующее описание

START/END:

RTU таймер задержки:

Скорость (бит/с)	RTU таймер задержки (мс)	Скорость (бит/с)	RTU таймер задержки (мс)
300	40	9,600	2
600	21	19,200	1
1,200	10	38,400	1
2,400	5	57,600	1
4,800	3	115,200	1

Адрес:
00 H: Передача на все устройства
01 H: на устройство по адресу 01
0F H: на устройство по адресу 15
10 H: на устройство по адресу 16
… и.т.д., макс адрес: 254 (‘FE’)

Код команды:
03 H: чтение содержимого из нескольких регистров
06 H: занесение одного слова в конкретный регистр
10 H: занесение содержимого в несколько регистров

Символы данных: Данные, передаваемые пользователю

CRC контрольная сумма: Начиная с адреса и заканчивая содержимым данных. Расчет выглядит следующим образом:

1. Шаг 1: Установка 16-битовый регистра (CRC регистр) = FFFFH
2. Шаг 2: Команда XOR для первого 8-бит байту командгого сообщения и младшего байта 16-бит регистра CRC. Результат размещается в CRC регистре.
3. Шаг 3: Сдвиг вправо CRC регистра побитно и заполнение "0" старшего бита..
4. Шаг 4: Проверка сдвига значения младшего бита (бит 0). Если значение Бит 0 равно "0" повторите шаг 3, если не равен "0", команда XOR работает со сдвинутым значением A001H и сохраняет результат в регистр CRC.
5. Шаг 5: Повторитен шаги 3 – 4 до завершения обработки всех 8 бит.
6. Шаг 6: Повторите шаги 2 – 5 для обработки всех сообщений. Заключительное значение в регистре CRC - контрольная сумма CRC.

Пример:
Чтение 2 непрерывных слов, сохраненных в регистрах устройства с адресом 01H (см. таблицу). Начальный регистр имеет адрес 2102H

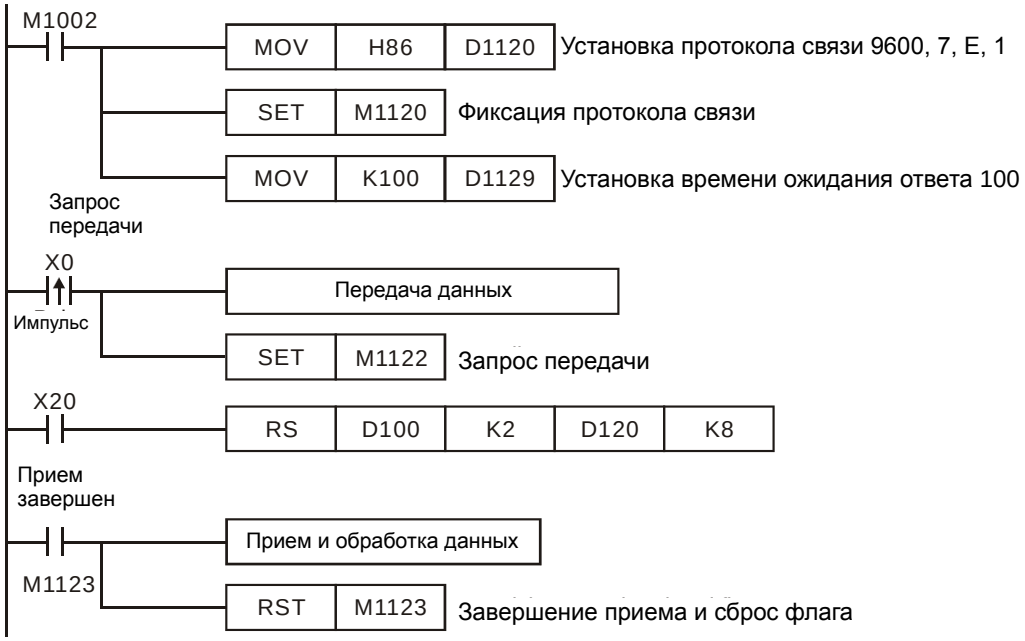
Командное сообщение:

Наименование	Данные (Hex)
Адрес	01 H
Функция	03 H
Адрес начального регистра	21 H
	02 H
Размер данных (пословно)	00 H
	02 H
CRC CHK Lo	6F H
CRC CHK Hi	F7 H

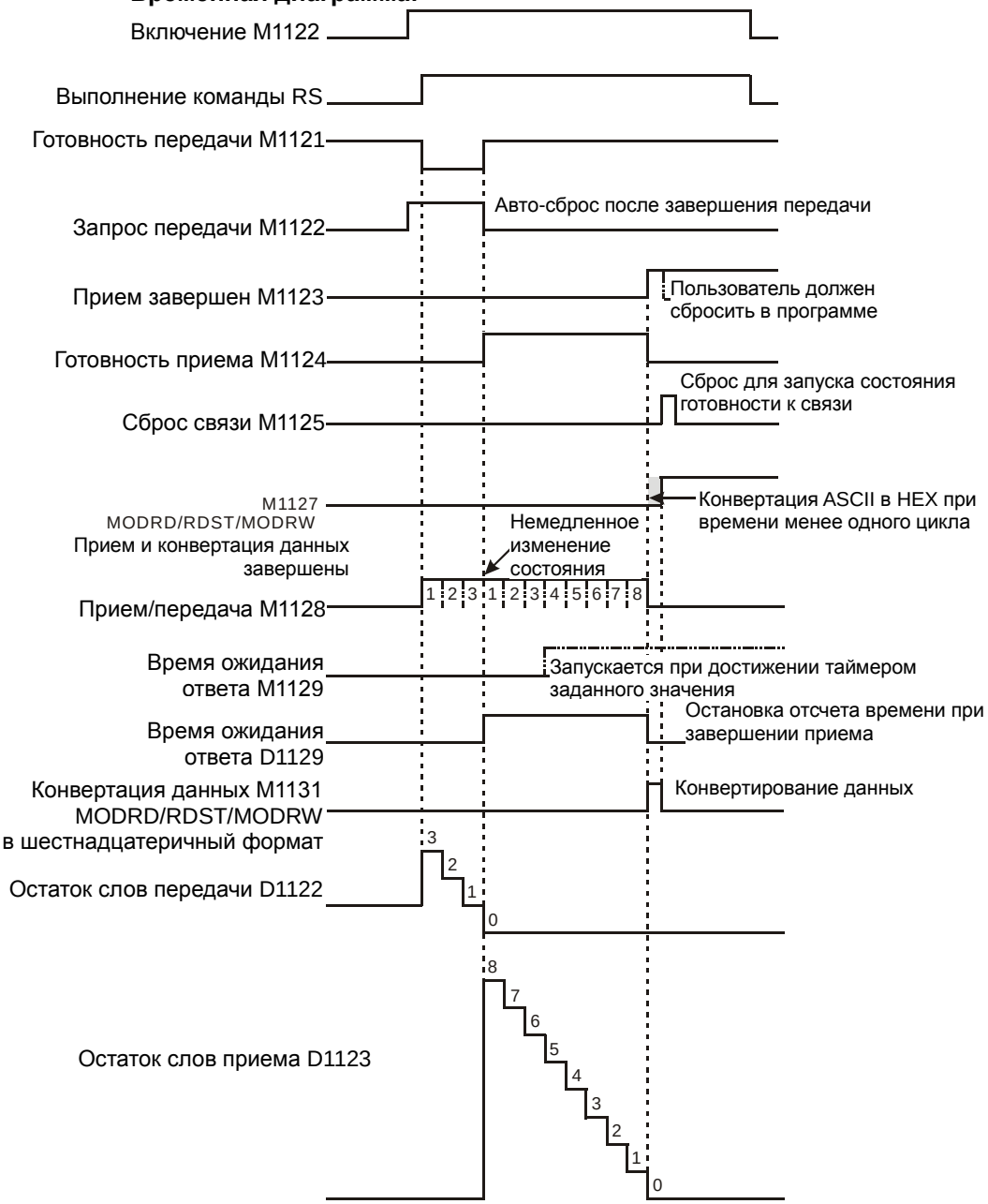
Ответное сообщение:

Наименование	Данные (Hex)
Адрес	01 H
Функция	03 H
Размер данных (пословно)	04 H
Содержимое адреса данных 2102H	17 H
	70 H
Содержимое адреса данных 2103H	00 H
	00 H
CRC CHK Lo	FE H
CRC CHK Hi	5C H

Пример программы соединения по RS-485:



Временная диаграмма:



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
81	D	PRUN		(S) (D)	Пересылка данных в восьмеричном формате				

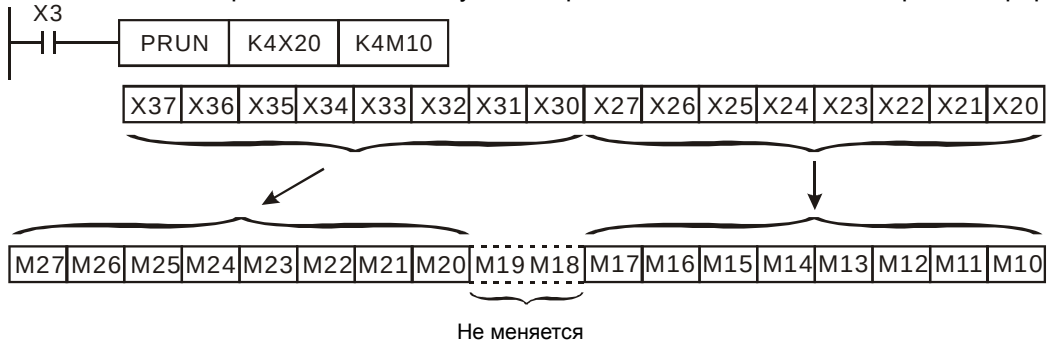
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*		*							PRUN, PRUNP: 5 шагов DPRUN, DPRUNP: 9 шагов
D								*	*							

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

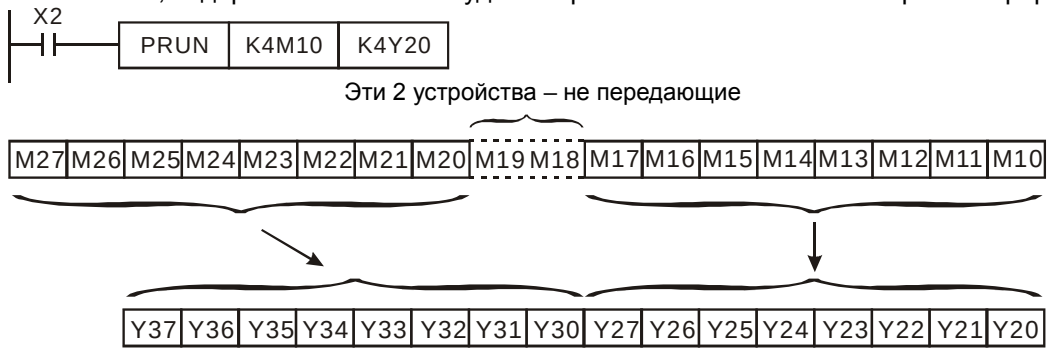
Операнды:
S: Источник данных **D**: Результат

Описание:
1. Эта команда пересылает содержимое **S** в **D** в восьмеричном режиме.
2. Стартовое значение X, Y, M в формате KnX, KnY, KnM должно быть кратным 10, т.е. X20, M20, Y20.
3. Когда операнд **S** определен как KnX, операнд **D** должен быть определен как KnM.
4. Когда операнд **S** определен как KnM, операнд **D** должен быть определен как KnY.

Пример программы 1:
Когда X3 включен, содержимое в K4X20 будет направлено в K4M10 в восьмеричной форме.



Пример программы 2:
Когда X2 включен, содержимое в K4M10 будет направлено в K4Y20 в восьмеричной форме.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
82		ASCI	P	(S) (D) (n)	Преобразование Hex в ASCII				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			ASCI, ASCIP: 7 шагов
D								*	*	*	*	*	*			
n					*	*										

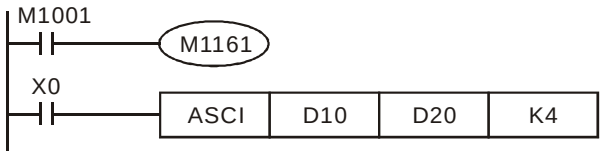
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат **n**: Количество шестнадцатеричных цифр для преобразования (**n** = 1~256)

Описание:

1. 16-битовый режим преобразования: Когда M1161 выключен, команда преобразует каждую цифру шестнадцатеричных данных из **S** в ASCII-коды и отправляет их старшие 8 бит и младшие 8 бит в **D**. **n** = число преобразованных цифр.
2. 8-битовый режим преобразования: Когда M1161 включен, команда преобразует каждую цифру шестнадцатеричных данных из **S** в ASCII-коды и отправляет их в младшие 8 бит в **D**. **n** число преобразованных цифр (все старшие 8 бит **D** = 0).
3. Флаг: M1161 (переключение режимов 8/16-бит)
4. Допустимое значение шестнадцатеричных данных: 0~9, A~F

Пример программы 1:

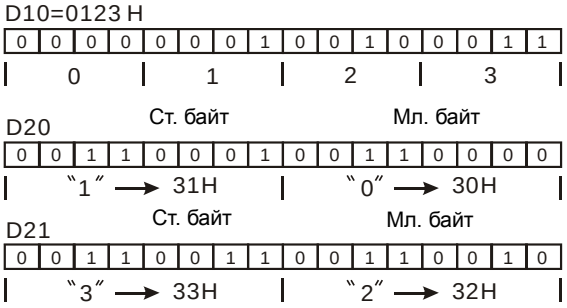


1. M1161 выключен, 16-битное преобразование.
2. Когда X0 включен, преобразуются 4 шестнадцатеричных значения (цифры) в D10 в ASCII-коде и помещаются в регистры, начиная с D20.

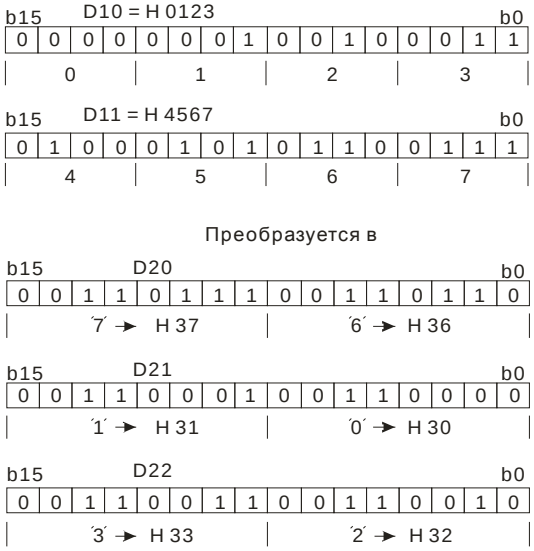
3. Принимаем значения:

(D10) = 0123 H	‘0’ = 30H	‘4’ = 34H	‘8’ = 38H
(D11) = 4567 H	‘1’ = 31H	‘5’ = 35H	‘9’ = 39H
(D12) = 89AB H	‘2’ = 32H	‘6’ = 36H	‘A’ = 41H
(D13) = CDEF H	‘3’ = 33H	‘7’ = 37H	‘B’ = 42H

4. Когда **n** = 4, битовая структура будет такой:



5. Когда **n** = 6, битовая структура будет такой:



6. Когда **n** = 1 до 16:

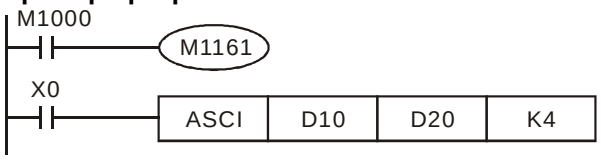
D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20 младший бит	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D20 старший бит		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D21 младший бит			“3”	“2”	“1”	“0”	“7”	“6”
D21 старший бит				“3”	“2”	“1”	“0”	“7”
D22 младший бит					“3”	“2”	“1”	“0”
D22 старший бит						“3”	“2”	“1”
D23 младший бит							“3”	“2”
D23 старший бит								“3”
D24 младший бит								
D24 старший бит								
D25 младший бит								
D25 старший бит								
D26 младший бит								
D26 старший бит								
D27 младший бит								
D27 старший бит								

Не конвер-тируется

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20 младший бит	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D20 старший бит	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D21 младший бит	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D21 старший бит	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D22 младший бит	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”

D22 старший бит	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D23 младший бит	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D23 старший бит	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D24 младший бит	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D24 старший бит		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D25 младший бит			“3”	“2”	“1”	“0”	“7”	“6”
D25 старший бит				“3”	“2”	“1”	“0”	“7”
D26 младший бит					“3”	“2”	“1”	“0”
D26 старший бит			Не конвер-тируется			“3”	“2”	“1”
D27 младший бит							“3”	“2”
D27 старший бит								“3”

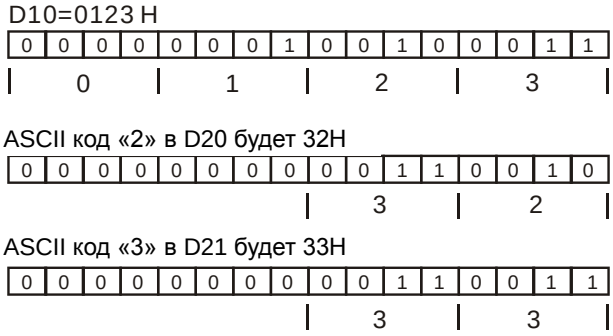
Пример программы 2:



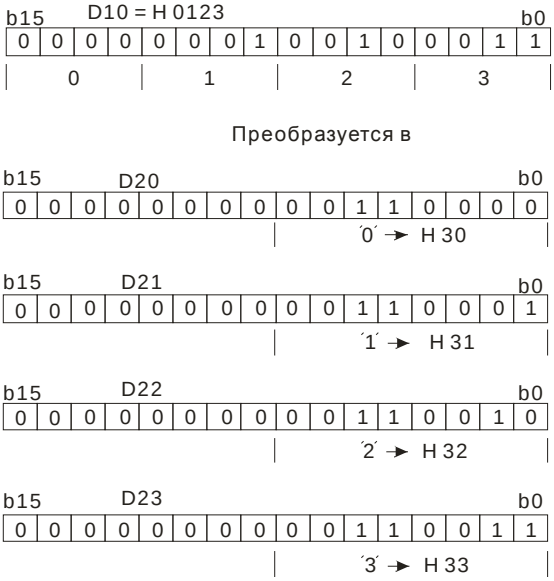
1. M1161 включен, 8-битовое преобразование.
2. Когда X0 включен, преобразуется 4 шестнадцатеричных значения (цифры) в D10 в ASCII-коды и перемещает результат в регистры, начиная с D20.
3. Принимаем значения:

(D10) = 0123 H	‘0’ = 30H	‘4’ = 34H	‘8’ = 38H
(D11) = 4567 H	‘1’ = 31H	‘5’ = 35H	‘9’ = 39H
(D12) = 89AB H	‘2’ = 32H	‘6’ = 36H	‘A’ = 41H
(D13) = CDEFH	‘3’ = 33H	‘7’ = 37H	‘B’ = 42H

4. Когда **n** = 2, битовая структура будет такой:



5. Когда **n** = 4, битовая структура будет такой:



6. Когда **n** = 1 ~ 16:

D \ n	K1	K2	K3	K4	K5	K6	K7	K8
D20	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D21		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D22			“3”	“2”	“1”	“0”	“7”	“6”
D23				“3”	“2”	“1”	“0”	“7”
D24					“3”	“2”	“1”	“0”
D25						“3”	“2”	“1”
D26							“3”	“2”
D27								“3”
D28				Не конвер-тируется				
D29								
D30								
D31								
D32								
D33								
D34								
D35								

D \ n	K9	K10	K11	K12	K13	K14	K15	K16
D20	“B”	“A”	“9”	“8”	“F”	“E”	“D”	“C”
D21	“4”	“B”	“A”	“9”	“8”	“F”	“E”	“D”
D22	“5”	“4”	“B”	“A”	“9”	“8”	“F”	“E”
D23	“6”	“5”	“4”	“B”	“A”	“9”	“8”	“F”
D24	“7”	“6”	“5”	“4”	“B”	“A”	“9”	“8”
D25	“0”	“7”	“6”	“5”	“4”	“B”	“A”	“9”
D26	“1”	“0”	“7”	“6”	“5”	“4”	“B”	“A”
D27	“2”	“1”	“0”	“7”	“6”	“5”	“4”	“B”
D28	“3”	“2”	“1”	“0”	“7”	“6”	“5”	“4”
D29		“3”	“2”	“1”	“0”	“7”	“6”	“5”
D30			“3”	“2”	“1”	“0”	“7”	“6”
D31				“3”	“2”	“1”	“0”	“7”
D32			Не конвер-тируется		“3”	“2”	“1”	“0”
D33						“3”	“2”	“1”
D34							“3”	“2”
D35								“3”

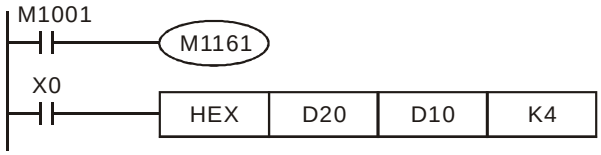
API	Команда			Операнды	Функция	Контроллеры			
83		HEX	P	(S) (D) (n)	Преобразование ASCII в HEX	ES2/EX2	SS2	SA2/SE	SX2

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат **n**: Количество символов для преобразования (**n** = 1~256)

Описание:
1. 16-битовый режим преобразования: Когда M1161 выключен, команда преобразует **n** символов в ASCII-коде, начиная с **S,** в шестнадцатеричные данные в битовом режиме и отправляет их в старший и младший бит **D**. **n** = число преобразованных символов.
2. 8-режим преобразования: Когда M1161 включен, команда преобразует **n** символов (только младшие биты) в ASCII-коде, начиная с **S,** в шестнадцатеричные данные в битовом формате и отправляет их в младшие биты **D**. **n** = число преобразованных символов (все старшие 8 бит в **D** = 0).
3. Диапазон шестнадцатеричных данных: 0~9, A~F

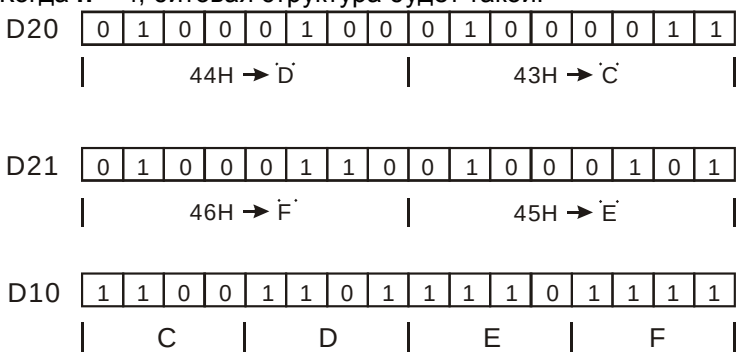
Пример программы 1:



1. M1161 выключен: 16-битовое преобразование.
2. Когда X0 включен, преобразованные 4 символа в ASCII-коде сохраняются в регистры D20~ D21 в шестнадцатеричном формате и результат отправляется в битовом режиме в регистр D10. **n** = 4
3. Значения регистров:

S	ASCII-код	HEX	S	ASCII-код	HEX
D20 младший бит	H 43	“C”	D24 младший бит	H 34	“4”
D20 старший бит	H 44	“D”	D24 старший бит	H 35	“5”
D21 младший бит	H 45	“E”	D25 младший бит	H 36	“6”
D21 старший бит	H 46	“F”	D25 старший бит	H 37	“7”
D22 младший бит	H 38	“8”	D26 младший бит	H 30	“0”
D22 старший бит	H 39	“9”	D26 старший бит	H 31	“1”
D23 младший бит	H 41	“A”	D27 младший бит	H 32	“2”
D23 старший бит	H 42	“B”	D27 старший бит	H 33	“3”

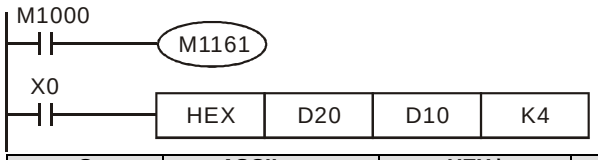
4. Когда **n** = 4, битовая структура будет такой:



5. Когда **n** = 1 ~ 16:

n \ D	D13	D12	D11	D10
1	Регистры не определены или равны «0»			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

Пример программы 2:

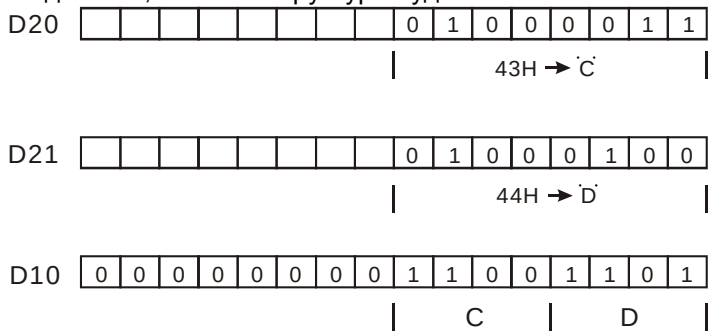


1. M1161 включен: 8-битовое преобразование.

2. Значения регистров:

S	ASCII-код	HEX \	S	ASCII-код	HEX
D20	H 43	“C”	D25	H 39	“9”
D21	H 44	“D”	D26	H 41	“A”
D22	H 45	“E”	D27	H 42	“B”
D23	H 46	“F”	D28	H 34	“4”
D24	H 38	“8”	D29	H 35	“5”
D30	H 36	“6”	D33	H 31	“1”
D31	H 37	“7”	D34	H 32	“2”
D32	H 30	“0”	D35	H 33	“3”

3. Когда **n** = 2, битовая структура будет такой



4. Когда **n** = 1 ... 16:

n \ D	D13	D12	D11	D10
1	Регистры не определены или равны «0»			***C H
2				**CD H
3				*CDE H
4				CDEF H
5			***C H	DEF8 H
6			**CD H	EF89 H
7			*CDE H	F89A H
8			CDEF H	89AB H
9		***C H	DEF8 H	9AB4 H
10		**CD H	EF89 H	AB45 H
11		*CDE H	F89A H	B456 H
12		CDEF H	89AB H	4567 H
13	***C H	DEF8 H	9AB4 H	5670 H
14	**CD H	EF89 H	AB45 H	6701 H
15	*CDE H	F89A H	B456 H	7012 H
16	CDEF H	89AB H	4567 H	0123 H

API	Команда			Операнды	Функция	Контроллеры			
84		CCD	P	S D n	Расчет контрольной суммы	ES2/EX2	SS2	SA2/SE	SX2

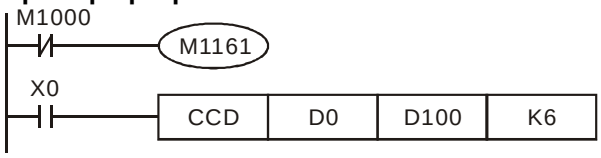
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			CCD, CCDP: 7 шагов
D									*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Исходные данные **D**: Адрес сохранения контрольной суммы **n**: Число проверяемых байт (**n** = 1~256)

Описание:
1. Эта команда рассчитывает контрольную сумму для обеспечения достоверности передачи данных.
2. 16-битовый расчет: Если M1161 выключен, **n** байт данных, начиная с младшего байта **S,** будут суммированы, контрольная сумма будет сохранена в **D**, а четные байты сохраняются в **D+1**.
3. 8-битовый расчет: Если M1161 включен, **n** байт данных, начиная с младшего байта **S** (используются только младшие байты), будут суммированы, контрольная сумма сохраняется в **D**, а четные байты сохраняются в **D+1**.

Пример программы 1:

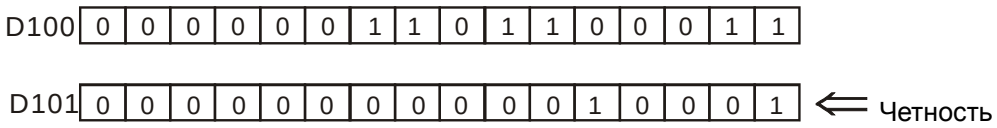


1. M1161 включен, 16-битовый расчет.
2. Когда X0 включен, 6 байт от младшего в D0 до старшего в D2 суммируются, контрольная сумма сохраняется в D100, а четные байты сохраняются D101.

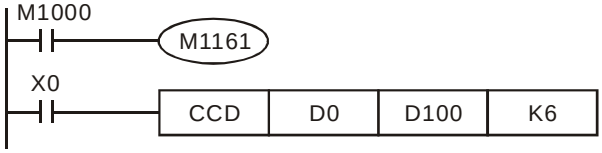
(S)	Данные
D0 мл. байт	K100 = 0 1 1 0 0 1 0 0
D0 ст. байт	K111 = 0 1 1 0 1 1 1 (1)
D1 мл. байт	K120 = 0 1 1 1 1 0 0 0
D1 ст. байт	K202 = 1 1 0 0 1 0 1 0
D2 мл. байт	K123 = 0 1 1 1 1 0 1 (1)
D2 ст. байт	K211 = 1 1 0 1 0 0 1 (1)
D100	K867
D101	0 0 0 1 0 0 0 (1)

всего

При четном результате отображается 0, при четном отображается 1



Пример программы 2:
1. M1161 включен, 8-битовый расчет.
2. Когда X0 включен, 6 байт от младшего в D0 до младшего в D5 суммируются, контрольная сумма сохраняется в D100, а четные байты сохраняются D101.



(S)	Данные
D0 мл. байт	K100 = 0 1 1 0 0 1 0 0
D1 мл. байт	K111 = 0 1 1 0 1 1 1 ①
D2 мл. байт	K120 = 0 1 1 1 1 0 0 0
D3 мл. байт	K202 = 1 1 0 0 1 0 1 0
D4 мл. байт	K123 = 0 1 1 1 1 0 1 ①
D5 мл. байт	K211 = 1 1 0 1 0 0 1 ①
D100	K867
D101	0 0 0 1 0 0 0 ①

всего

При четном результате отображается 0, при четном отображается 1

D100	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1
D101	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

⇐ Четность

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
85	VRRD	P	(S) (D)	Чтение значения с потенциометра				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRRD, VRRDP: 5 шагов
D								*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

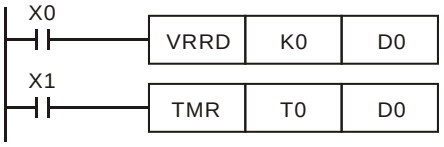
Операнды:

S: Номер потенциометра (0~1) **D**: Сохранение результата чтения

Описание:

1. Команда VRRD используется для чтения значения с двух потенциометров, встроенных в ПЛК. Читаемые значения преобразуются в диапазон 0 ~ 255 и сохраняются в **D**.
2. Потенциометр можно использовать для установки значения таймера.

Пример программы:



1. Когда X0 включен, значение потенциометра №0 будет считано, преобразовано в 8-битный двоичный формат (0~255) и сохранено в D0.
2. Когда X1 включен, таймер, применяющий установленное в D0 значение, начнет отсчет времени.

Заметки:

Значение	Функция
M1178	Включение VR0
M1179	Включение VR1
D1178	VR0 значение
D1179	VR1 значение

SX2 имеет 2 встроенных потенциометра, которые можно применять со специальными регистрами D и специальными реле M.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
86	VRSC	P	(S) (D)	Чтение положения потенциометра				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*										VRSC, VRSCP: 5 шагов
D								*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Номер потенциометра (0~1) **D**: Хранение значения положения

Описание:
Команда VRSC считывает положение (0~10) двух потенциометров ПЛК и сохраняет значение в **D** как целое, т.е. округляя значение.

Пример программы 1:



Когда X0 включен, команда VRSC считывает положение (0 -10) потенциометра № 0 и сохраняет его в D10.

Пример программы 2:
Потенциометр в качестве цифрового переключателя: Положение 0~10 потенциометра соотносится с M10~M20, поэтому только один из M10~M20 может быть включен одновременно. Когда M10~M20 включены, используйте команду DECO (API 41) для дешифровки значения положения в M10~M25.

1. Когда X0 включен, положение (0~10) потенциометра 1 будет считано и сохранено в D1.
2. Когда X1 включен, команда DECO дешифрует значение градации (0~10) в M10~M25.



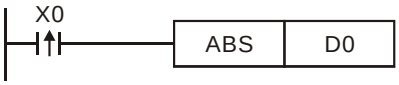
API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
87	D	ABS	P	(D)	Абсолютное значение				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*	*	*	ABS, ABSP: 3 шага DABS, DABSP: 5 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Описание
1. Команда определяет абсолютное значение **D.**
2. Команда обычно используется в импульсном режиме. (ABSP, DABSP).
3. Если операнд D используется с индексом F, возможно выполнение только 16-битных команд.

Пример программы:



Когда X0 включается, команда ABS определяет абсолютное значение (модуль) данных в D0.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
88	D	PID		(S1) (S2) (S3) (D)	ПИД-регулятор				

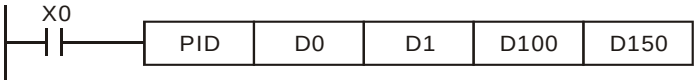
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			PID : 9 шагов DPID: 17 шагов
S_2													*			
S_3													*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

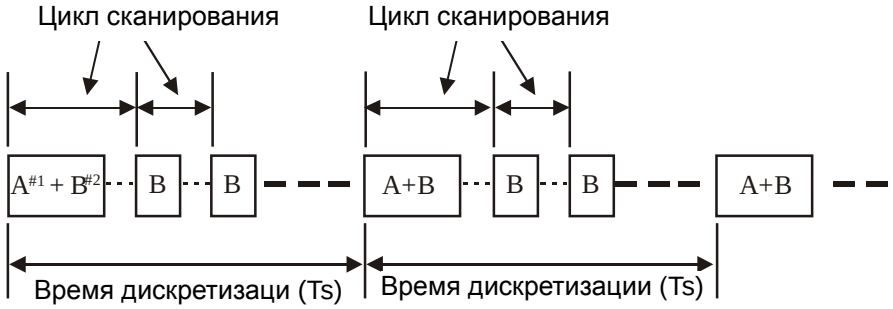
Операнды:
S_1: Заданное значение контура регулирования (SV) **S_2**: Действительное значение контура регулирования (PV) **S_3**: Параметры регулирования (для 16-битной команды – 20 последовательных адресов, для 32-битной команды – 21 последовательный адрес)
D: Выходное значение регулятора (MV)

Описание:
1. Команда выполняет ПИД-регулирование. Операция будет выполняться по достижении времени дискретизации. ПИД-регулирование имеет пропорциональную, интегральную и дифференциальную составляющую. ПИД-регулирование широко применяется во многих образцах механического, гидравлического, пневматического и электронного оборудования.
2. После установления всех параметров, команда может быть выполнена, и результаты будут сохранены в D. Операнд D должен быть регистром данных без запоминания. При необходимости использовать область регистров данных с запоминанием необходимо их обнулять в начале программы.

Пример программы:



1. Настройте параметры перед выполнением команды.
2. Когда X0 включен, команда начнет выполняться, и выходное значение регулятора будет записываться в D150. Когда X0 выключится, выполнение команды прервется, и данные в D150 останутся неизменными.
3. Временная диаграмма ПИД-регулирования (макс. время операции приблизительно 80 мкс)



#1 – время расчета ПИД-регулятора в цикле сканирования (прибл.72 мкс)
#2 – время ПИД-регулятора в цикле сканирования без расчета (прибл. 8 мкс)

Заметки:
1. Не существует ограничений на количество использований команды. Тем не менее, номер регистра в **S_3**~ **S_3**+19 не должен повторяться.
2. Для 16-битной команды, **S_3** занимает 20 регистров. В примере программы диапазон **S_3** - D100 ~ D119.
3. Перед выполнением команды пользователь должен задать параметры в указанную область регистров командой MOV. Если регистры необходимо зафиксировать, используйте команду MOVP для задания параметров только один раз.
4. Параметры регулирования **S_3** для 16-битной команды:

Регистр	Функция	Значение	Описание
S_3:	Время дискретизации (T_S)	1~2 000 (ед. изм.: 10 мс)	Временной интервал между расчетами ПИД-регулирования и обновлением параметров MV. Если T_S = 0, команда PID выполняться не будет. Если T_S меньше времени 1 цикла программы, команда PID будет выполняться в каждом цикле программы, т.е минимальное значение T_S должно быть больше времени цикла программы.
S_3+1:	Коэффициент пропорциональности (K_P)	0~30 000(%)	Коэффициент умножения для пропорционального регулятора.
S_3+2:	Интегральный коэффициент (K_I)	0~30 000(%)	Коэффициент умножения для интегрального регулятора (компенсация накопленного расхождения SV и PV). Для режима управления K0~K5.
	Интегральная постоянная времени (T_I)	0~30 000 (мс)	Для режима управления K10

Регистр	Функция	Значение	Описание
S₃+3:	Дифференциальный коэффициент (K_D)	-30 000~30 000 (%)	Коэффициент умножения для дифференциального регулятора (Скорость изменения процесса расхождения). Для режима управления K0~K5
	Дифференциальная постоянная времени (T_D)	-30 000~30 000 (мс)	Для режима управления K10
S₃+4:	Режим регулирования	0: Автоматическое регулирование 1: Прямое регулирование (E = SV - PV). 2: Обратное регулирование (E = PV - SV). 3: Автонастройка параметров только для регулирования температуры (не поддерживается в 32-битном режиме). После автонастройки коэффициентов KP, KI и KD в регистр **S₃**+4 автоматически запишется K4. 4: Специальный для регулирования температуры (не поддерживается в 32-битном режиме) 5: Автоматический режим с управлением верхней/нижней границей MV - при достижении границ накопление интегрального значения останавливается. 10: T_I / T_D режим с управлением верхней/нижней границей MV - при достижении границ накопление интегрального значения останавливается.	
S₃+5:	Зона нечувствительности (E)	0~32 767	E – допустимое отклонение между SV и PV. Если **S₃** +5 = 5, то при нахождении E в диапазоне от -5 до 5, MV = 0. Когда **S₃** +5 = K0, функция не работает
S₃+6:	Верхнее ограничение выхода (MV)	-32 768~32 767	Пример: если **S₃**+6 установлен в 1 000, то MV максимально равен 1 000 даже при необходимости превышения этого значения. **S₃**+6 должен быть больше или равен **S₃**+7.
S₃+7:	Нижнее ограничение выхода (MV)	-32 768~32 767	Пример: Если **S₃**+7 установлен в -1 000, MV минимально равен -1 000 даже при необходимости меньшего значения.
S₃+8:	Верхнее ограничение интегрирования	-32 768~32 767	Пример: Если **S₃**+8 установлен в 1 000, то MV максимально будет 1 000 даже при необходимости превышения этого значения и интегрирование выполняться не будет. **S₃**+8 должен быть больше или равен **S₃** +9;
S₃+9:	Нижнее ограничение интегрирования	-32 768~32 767	Пример: Если **S₃**+9 установлен в -1,000, то MV минимально равно -1 000 даже при необходимости меньшего значения и интегрирование выполняться не будет.
S₃+10, 11:	Накопленное значение интегрирования	Доступный диапазон 32-битных чисел с плавающей запятой	Как правило, справочные величины. Могут использоваться в дальнейшей обработке. Эти параметры можно удалять или изменить.
S₃ +12:	Значение предыдущего PV	-32 768~32 767	
S₃+13 ~ **S₃**+19	Используются только системой!		

5. Для **S₃**+1~3, когда установленные параметры превышают диапазон, в качестве заданного значения будет выбрана верхняя/нижняя граница.
6. Если заданное напрвление Вперед/Назад превысит допустимый диапазон, параметр будет установлен в 0.
7. Команда PID может включаться в подпрограммы обработки прерывания, STL программу или программы перехода CJ.
8. Максимальное отклонение времени дискретизации T_S = - (1 время цикла + 1 мс) ~ + (1 время цикла). Если отклонение влияет на параметры выхода, необходимо исправить время цикла или выполнять команду PID внутри программы прерывания.
9. Параметр PV команды PID должен оставаться постоянным до окончания выполнения программы. Если с командой PID используются модули аналогового ввода, следует внимательно выставить параметры аналогового/цифрового преобразования для этих модулей.
10. Для 32-битной команды **S₃** занимает 21 регистр. В примере программы, указанное значение **S₃** находится в D100 ~ D120. Перед выполнением команды пользователь должен задать параметры в указанную область регистров командой MOV. Если регистры необходимо зафиксировать, используйте команду MOVP для задания параметров только один раз.
11. Таблица параметров **S₃** для 32-битной команды:

Регистр	Функция	Значение	Описание
S3:	Время дискретизации (T_S)	1~2 000 (ед. изм.: 10 мс)	Временной интервал между расчетами ПИД-регулирования и обновлением параметров MV. Если T_S = 0, команда PID выполняться не будет. Если T_S меньше времени 1 цикла программы, команда PID будет выполняться в каждом цикле программы, т.е минимальное значение T_S должно быть больше времени цикла программы.
S3+1:	Коэффициент пропорциональности (K_P)	0~30 000(%)	Коэффициент умножения для пропорционального регулятора.
S3+2:	Интегральный коэффициент (K_I)	0~30 000(%)	Коэффициент умножения для интегрального регулятора (компенсация накопленного расхождения SV и PV). Для режима управления K0~K2, K5.
	Интегральная постоянная времени (T_I)	0~30 000 (мс)	Для режима управления K10
S3+3:	Дифференциальный коэффициент (K_D)	-30 000~30 000 (%)	Коэффициент умножения для дифференциального регулятора (Скорость изменения процесса расхождения). Для режима управления K0~K2, K5
	Дифференциальная постоянная времени (T_D)	-30 000~30 000 (мс)	Для режима управления K10
S3+4:	Режим регулирования	0: Автоматическое регулирование 1: Прямое регулирование (E = SV - PV). 2: Обратное регулирование (E = PV - SV). 5: Автоматический режим с управлением верхней/нижней границей MV - при достижении границ накопление интегрального значения останавливается. 10: T_I / T_D режим с управлением верхней/нижней границей MV - при достижении границ накопление интегрального значения останавливается.	
S3+5, 6:	Зона нечувствительности (E), 32-бит	0~ 2 147 483 647	E – допустимое отклонение между SV и PV. Если **S3** +5 = 5, то при нахождении E в диапазоне от -5 до 5, MV = 0. Когда **S3** +5 = K0, функция не работает
S3+7, 8:	Верхнее ограничение выхода (MV), 32-бит	-2 147 483 647~ 2 147 483 647	Пример: если **S3**+7, 8 установлен в 1 000, то MV максимально равен 1 000 даже при необходимости превышения этого значения. **S3**+7, 8 должен быть больше или равен **S3**+9, 10.
S3+9, 10:	Нижнее ограничение выхода (MV), 32-бит	-2 147 483 647~ 2 147 483 647	Пример: Если **S3**+9, 10 установлен в -1 000, MV минимально равен -1 000 даже при необходимости меньшего значения.
S3+11, 12:	Верхнее ограничение интегрирования, 32-бит	-2 147 483 647~ 2 147 483 647	Пример: Если **S3**+11, 12 установлен в 1 000, то MV максимально будет 1 000 даже при необходимости превышения этого значения и интегрирование выполняться не будет. **S3**+11,12 должен быть больше или равен **S3** +13, 14;
S3+13, 14:	Нижнее ограничение интегрирования, 32-бит	-2 147 483 647~ 2 147 483 647	Пример: Если **S3**+13, 14 установлен в -1,000, то MV минимально равно -1 000 даже при необходимости меньшего значения и интегрирование выполняться не будет.
S3+15, 16:	Накопленное значение интегрирования, 32-бит	Доступный диапазон 32-битных чисел с плавающей запятой	Как правило, справочные величины. Могут использоваться в дальнейшей обработке. Эти параметры можно удалять или изменить.
S3 +17, 18:	Значение предыдущего PV, 32-бит	-2 147 483 647~ 2 147 483 647	
S3+13 ~ **S3**+19	Используются только системой!		

12. Описание 32-битного значения **S3** и 16-битного значения **S3** почти одинаковы. Разница заключается в возможности установки значений **S3**+5 ~ **S3**+20.

PID-уравнения:

1. Когда режим регулирования (**S3**+4) определяется как K0, K1, K2 и K5:
   - В этом режиме управления команда PID может выполняться в режимах: автоматическом, прямого и обратного регулирования и автоматическом с контролем верхней/нижней границы MV. Прямое/обратное регулирование зафиксировано в **S3**+4. Другие настройки зафиксированы в регистрах **S3** ~ **S3**+5.
   - PID–уравнение для режимов регулирования K0~K2:

где:

$$MV = K_P * E(t) + K_I * E(t)\frac{1}{S} + K_D * PV(t)S$$

MV : Выходное значение ПИД-регулятора

K_P : Коэффициент пропорциональности

$E(t)$: Отклонение

$PV\ (t)$: Текущее измеренное значение (сигнал обратной связи)

$SV\ (t)$: Заданное значение

K_D : Дифференциальный коэффициент

$PV(t)S$: Значение дифференцирования PV(t)

K_I : Интегральный коэффициент

$E(t)\frac{1}{S}$: Значение интегрирования E(t)

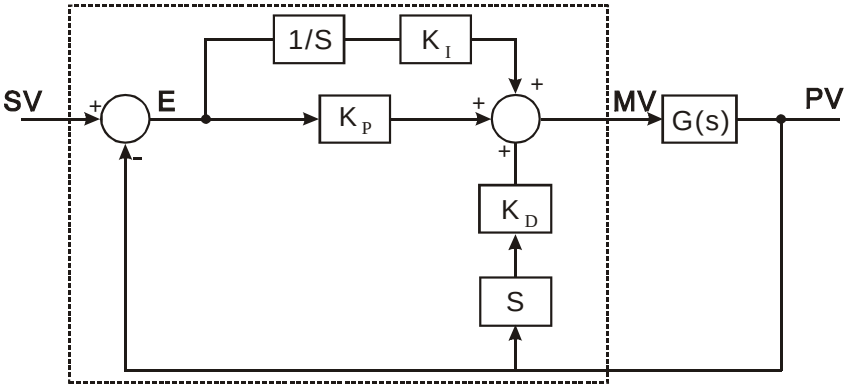
- Когда E(t) меньше 0 в режимах управления "прямое/обратное регулирование", E(t) будет иметь значение “0"

Режим	PID-уравнение
Прямое, автоматическ.	E(t) = SV – PV
Обратное	E(t) = PV – SV

- Схема регулирования:

На схеме ниже S является производной операции «(PV-предыдущее PV)/время дискретизации». 1/S является интегральной операцией, получаясь из "предыдущее интегральное значение + (отклонение x время дискретизации). G(S) является регулируемым устройством.

ПИД-команда выделена пунктиром



- Из уравнения видно, что данная операция отличается от основной работы ПИД-регулятора с применением дифференцирования значения. Чтобы избежать слишком большого дифференцирования, команда PID автоматически снижает значение выхода MV.

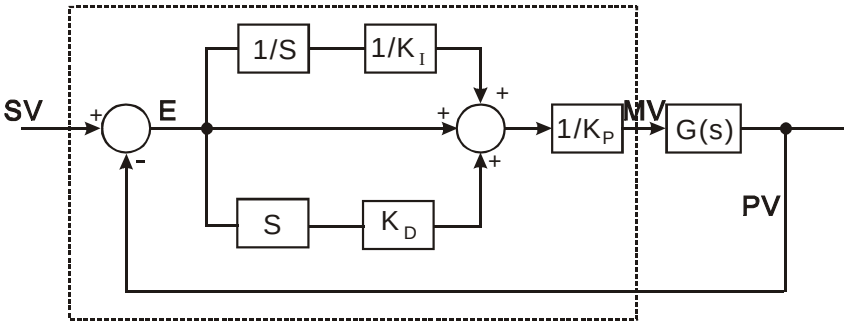
2. Когда режим регулирования (**S_3**+4) определен как K3 и K4:
Вид уравнения только для регулирования температуры:

$$MV = \frac{1}{K_P}\left[E(t) + \frac{1}{K_I}\left(E(t)\frac{1}{S}\right) + K_D * E(t)S\right], \text{ где } \mathrm{E(t) = SV(t) - PV(t)}$$

- Схема регулирования:

Т.к. этот режим предназначен только для регулирования температуры, команда PID должна применяться вместе с командой GPWM.
См. **Приложение 3** для более подробной информации

ПИД-команда выделена пунктиром



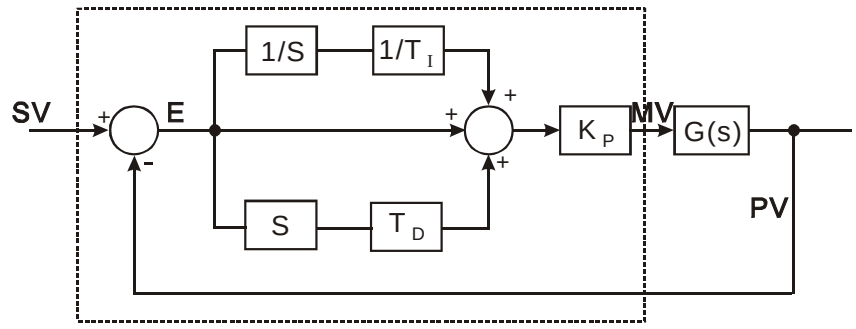
- Это уравнение предназначено исключительно для регулирования температуры. Поэтому, когда время дискретизации (T_S) установлено как 4 сек (K400), выходной диапазон (MV) будет K0 ~ K4000 и время цикла используемой совместно команды GPWM также должно не превышать 4 сек (K4000).
- Пользователь может выбрать автоматическую настройку параметров K3. В этом случае после установки всех параметров (метод регулирования автоматически установится как K4), пользователь может менять параметры вручную.

3. Когда режим регулирования (**S_3**+4) установлен как K10:
   - **S_3**+2 (K_I) и **S_3**+3 (K_D) в этом режиме будут включены как настройки интегральной постоянной времени (T_I) и дифференциальной постоянной времени (T_D).
   - Когда выходное значение MV достигает верхней/нижней границы, накопленное интегральное значение не будет соответственно увеличиваться/уменьшаться.
   - Уравнение для этого режима выглядит следующим образом:

$$MV = K_P \times \left[E(t) + \frac{1}{T_I}\int E(t)dt + T_D \frac{d}{dt}E(t) \right], \text{ где } E(t) = SV(t) - PV(t)$$

   - Схема регулирования:

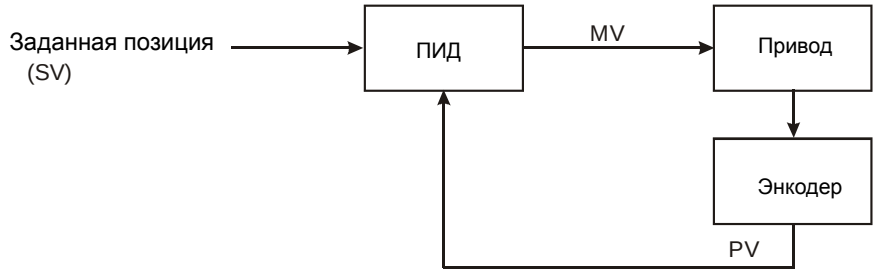
ПИД-регулирование (выделено пунктиром)



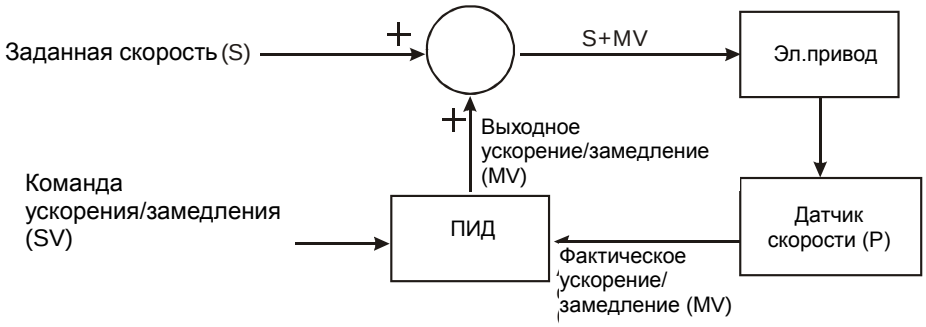
Замечания:

1. **S_3** + 3 может быть только в диапазоне 0 ~ 30 000.
2. Необходимо следить за правильностью выбора параметров настройки функций регулирования. Например, если выбрана автонастройка для регулирования температуры (**S_3** + 4 = K3), команда не может быть использована для управления двигателем и т.п.
3. При настройке трех основных параметров, K_P, K_I и K_D (**S_3** + 4 = K0 ~ K2), отрегулируйте K_P в первую очередь, установив K_I и K_D в 0. Далее переходите к настройке K_I и K_D, начав с небольших значений (см. пример 4 для корректировки настроек). K_P = 100 означает 100%, т.е. пропорциональное увеличение для отклонения - 1. K_P < 100% приведет к снижению отклонения, а K_P > 100% к увеличению.

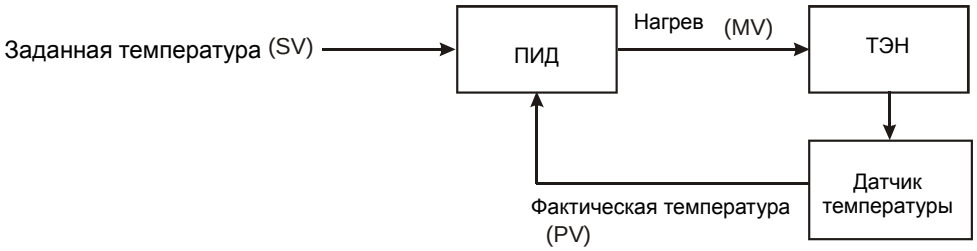
Пример 1: Блок схема при управлении позиционированием (**S_3**+4 = 0)



Пример 2: Блок-схема при управлении скоростью привода переменного тока (**S_3**+4 = 0)

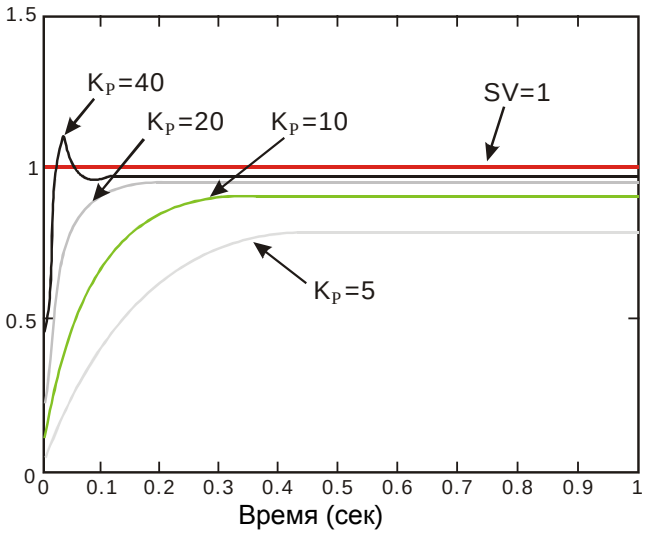


Пример 3: Блок схема при управлении регулирования температуры (**S_3**+4 = 1)



Пример 4: Последовательность настройки параметров ПИД-регулятора
Рассмотрим систему регулирования, описанную уравнением $G(s)=\frac{b}{s+a}$ (для большинства моделей электроприводов переменного тока), SV = 1, и время дискретизации (T_S) = 10 мс. Настройка параметров:

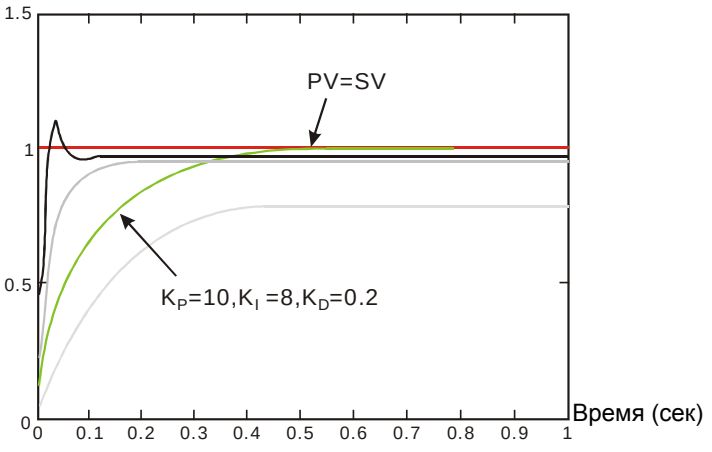
Шаг 1:
Установим K_I и K_D равными 0, а K_P равным 5, 10, 20, 40. Запишем при этом состояния SV и PV, см.график:



Шаг 2:
Когда K_P = 40, малое время отклика и, соответственно, перерегулирование.
Когда K_P = 20, также быстрый отклик, но без перерегулирования. Однако кратковременные импульсы MV будут приводить к частым старт/стоповым режимам и пусковым нагрузкам.
Когда K_P = 10, система имеет достаточно быстрый и плавный отклик. Выбираем этот режим.
Когда K_P = 5, очень медленный отклик.

Шаг 3:

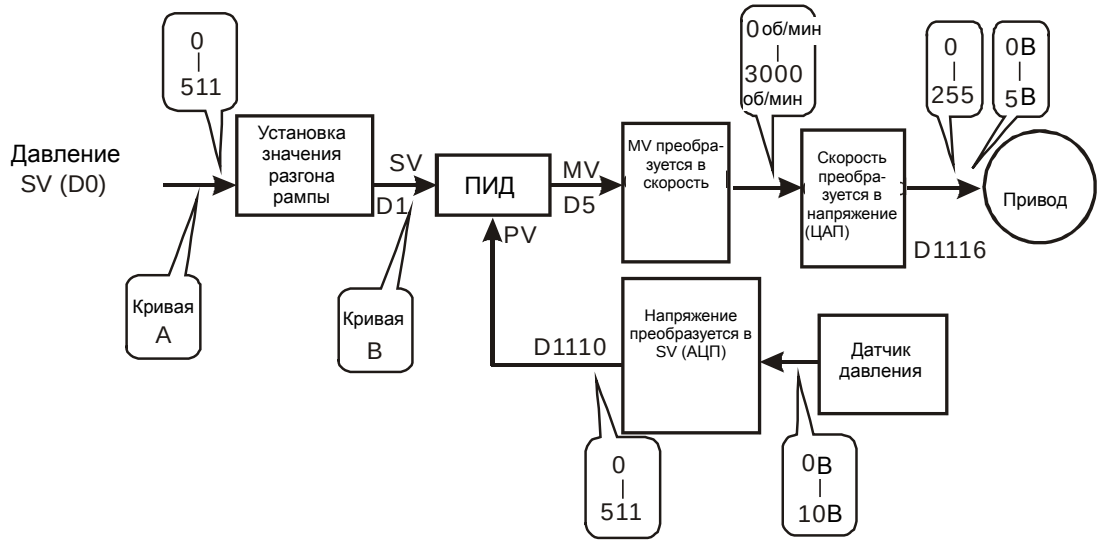
Выбрав K_P = 10, настроим K_I – градация 1, 2, 4, 8. K_I не должен быть больше K_P. Затем настроим K_D в градации 0.01, 0.05, 0.1, 0.2. K_D должен быть не больше 10% от K_P. Получим значения PV и SV (см. ниже):

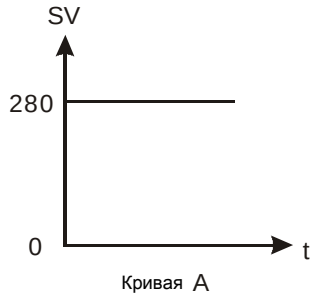


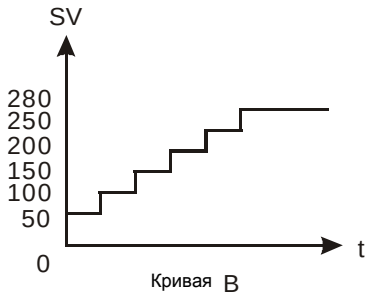
Пример применения 1:
ПИД-регулятор для системы регулирования давления (блок-схема пример 1)
Цель регулирования: Включение системы управления при достижении заданного давления.
Характеристика системы: Заданное значение должно достигаться постепенно для предотвращения перегрузок, возможных при быстрой реакции.
Рекомендуемое решение:

Решение 1: Большое время дискретизации

Решение 2: Использовать команду задержки и блок-схему ниже:



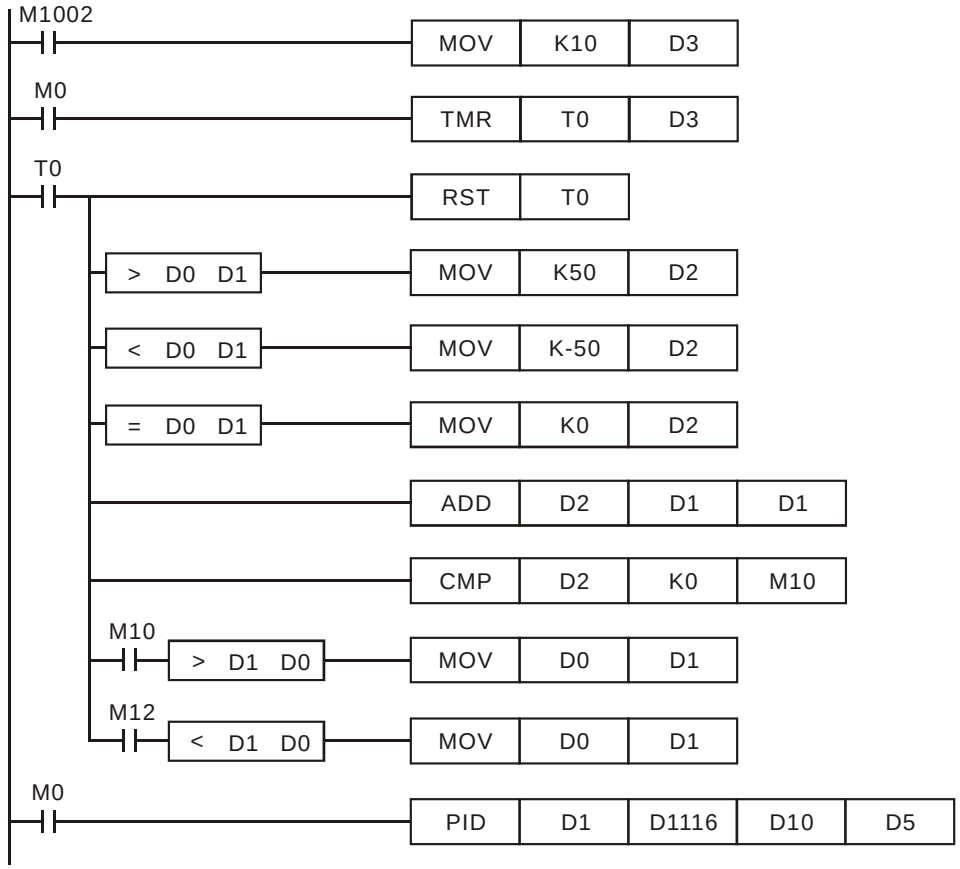




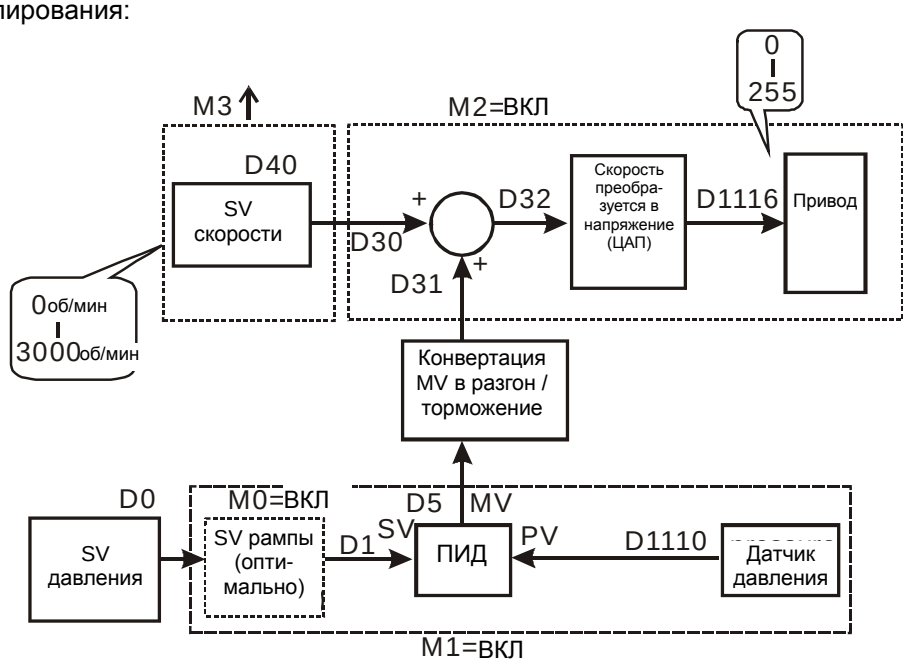
D2 хранит значение увеличения сигнала для каждого шага
D3 хранит длительность каждого шага

Значения в D2 и D3 при необходимости можно изменять

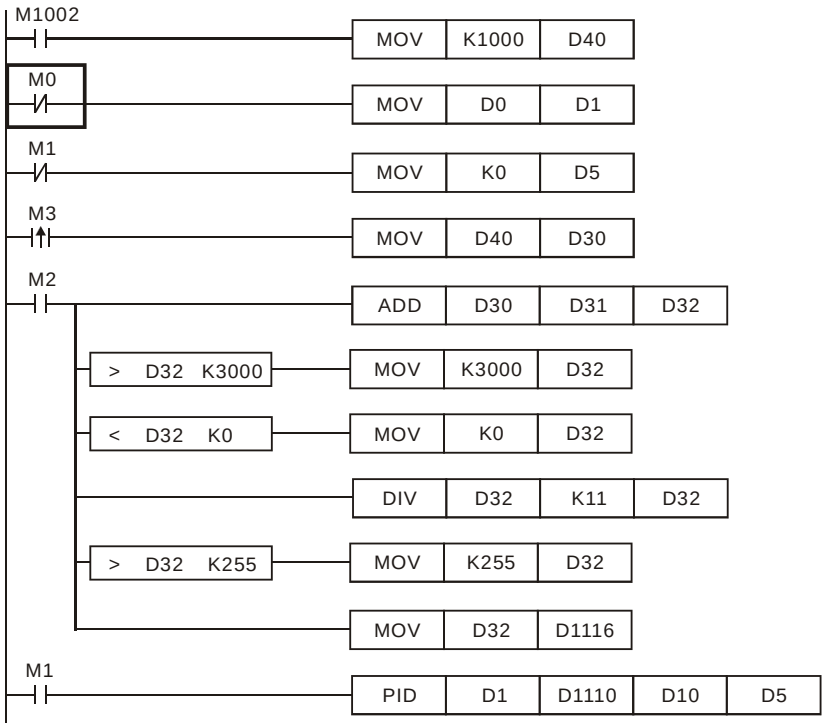
Программа, реализующая команду задержки:



Пример применения 2:
Одновременное управление двумя параметрами - управление скоростью и регулирование давления (диаграмма Пример 2)
Цель регулирования: Система имеет разомкнутый контур скорости и замкнутый контур давления.
Характеристика системы: Так как системы регулирования скорости и давления не связаны, необходимо построить сначала разомкнутый контур регулирования скорости, а потом закрытый контур регулирования давления. При избыточном изменении сигнала в контуре давления, можно применять рамповую функцию SC (подробнее в Приложении 1). См. схему регулирования:

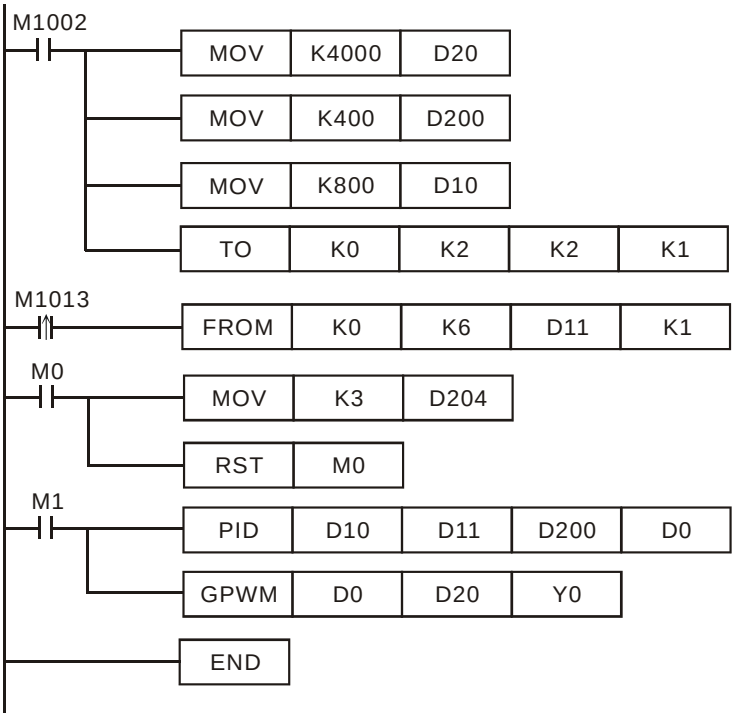


Пример программы:

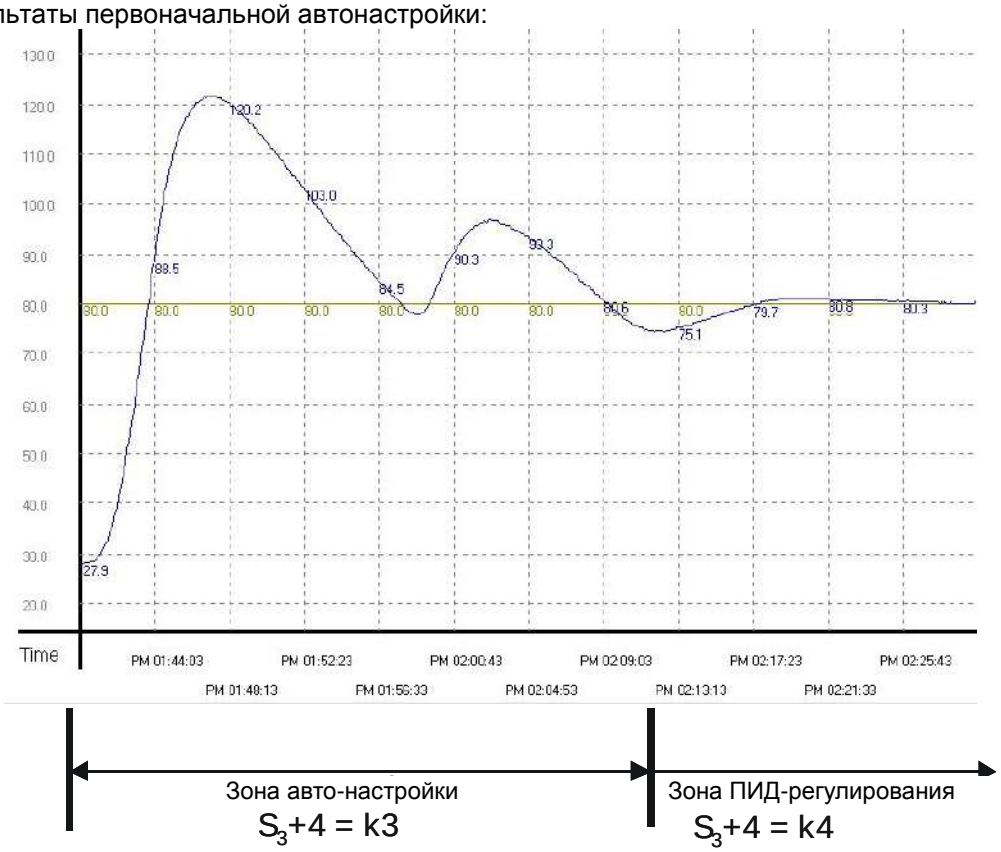


Пример применения 3:
Автонастройка регулирования температуры
Цель регулирования: Расчет оптимальных параметров для ПИД-регулирования температуры
Управление свойствами: Рекомендуется первоначально осуществить автонастройку ПИД для регулировки температуры (S_3+4 = K3). После автонастройки команда автоматически перейдет в режим ПИД-регулятора температуры (S_3+4 = K4). В качестве примера представлен процесс регулирования температуры нагрева печи.

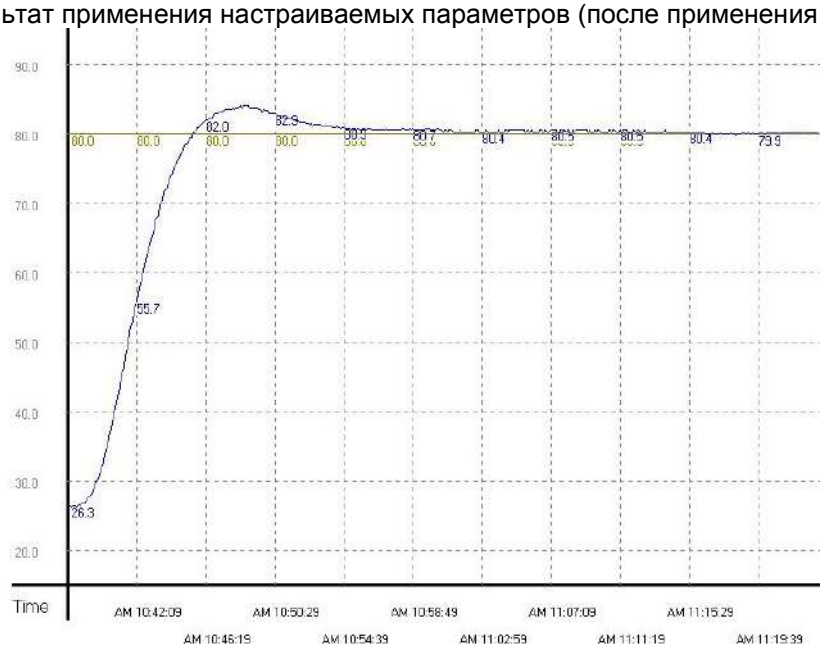
Пример программы:



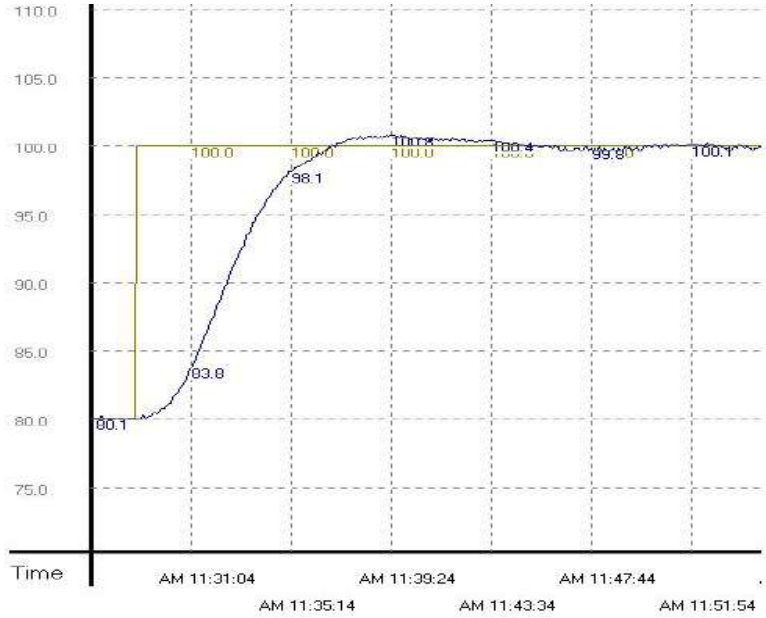
Результаты первоначальной автонастройки:



Результат применения настраиваемых параметров (после применения автонастройки):



Из графика видно, что регулирование температуры после автонастройки работает нормально и требует не более 20 мин для проведения процесса. Далее, изменяем заданную температуру от 80°C до 100°C и получаем следующий результат:



Видно, что когда параметр 100°C, процесс регулирования температуры занимает всего 20 мин, аналогично работе с 80°C.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
89	PLS	(S)	Создание импульса по переднему фронту				

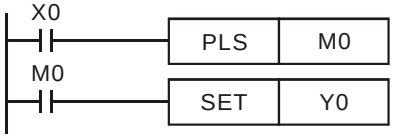
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PLS: 3 шага
S		*	*													

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

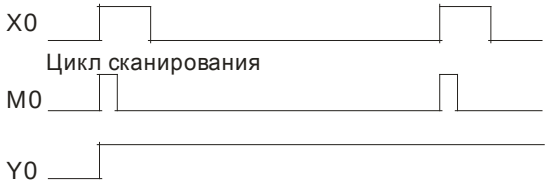
Операнды:
S: Выход для импульса

Описание:
Команда PLS служит для генерации одного импульса – опознание переднего фронта сигнала - независимо от продолжительности входного сигнала для включения соответствующего операнда. Операнд остается включенным на протяжении 1 цикла программы (скана).

Пример программы:
Релейно-контактная схема:



Временная диаграмма:



Коды команды:		Операция:
LD	X0	; Загрузка Н/О контакта X0
PLS	**M0**	; Задание на выходе M0 импульса по переднему фронту
LD	M0	; Загрузка Н/О контакта M0
SET	Y0	; Включение Y0

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
90	LDP	(S)	Логическое выражение с опросом по переднему фронту				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LDP: 3 шага
S	*	*	*	*							*	*				

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Опрашиваемый вход для определения переднего фронта сигнала

Описание:
Команда LDP используется для программирования импульсного начала логической связи, поэтому должна стоять в начале цепи. Инструкция LDP включается по переднему фронту сигнала и остается включенной один цикл программы (скана).

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LDP	**X0**	; Начало регистрации переднего фронта X0
AND	X1	; Последовательный Н/О контакт X1
OUT	Y1	; Включение выхода Y1

Заметки: Если связанный с передним фронтом контакт активен до включения питания ПЛК, контакт включится после включения питания ПЛК.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
91	LDF	(S)	Логическое выражение с опросом по заднему фронту				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LDF: 3 шага
S	*	*	*	*							*	*				

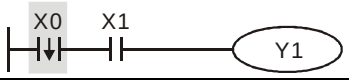
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

S: Опрашиваемый вход для определения заднего фронта сигнала

Описание:
Команда LDF используется для программирования импульсного начала логической связи, поэтому должна стоять в начале цепи. Инструкция LDF включается по заднему фронту сигнала и остается включенной один цикл программы (скана).

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LDF	**X0**	; Начало регистрации заднего контакта X0
AND	X1	; Последовательный Н/О контакт X1
OUT	Y1	; Установка катушки Y1

API	Команда	Операнды	Функция	Контроллеры
92	ANDP	(S)	«И» с опросом по переднему фронту	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ANDP: 3 шага
S	*	*	*	*							*	*				

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Опрашиваемый вход для определения переднего фронта сигнала

Описание:
Последовательное подключение входа с опросом по переднему фронту.

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LD	X0	Загрузка контакта X0
ANDP	**X1**	Последовательное подключение X1 с опросом по переднему фронту
OUT	Y1	Установка катушки Y1

API	Команда	Операнды	Функция	Контроллеры
93	ANDF	(S)	«И» с опросом по заднему фронту	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ANDF: 3 шага
S	*	*	*	*							*	*				

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Опрашиваемый вход для определения заднего фронта сигнала

Описание:
Последовательное подключение входа с опросом по заднему фронту.

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LD	X0	Загрузка контакта X0
ANDF	**X1**	Последовательное подключение X1 с опросом по заднему фронту
OUT	Y1	Установка катушки Y1

API	Команда	Операнды	Функция	Контроллеры
94	ORP	(S)	«ИЛИ» с опросом по переднему фронту	ES2/EX2 | SS2 | SA2/SE | SX2

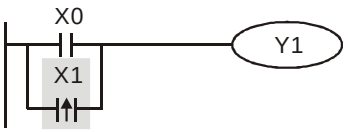
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ORP: 3 шага
S	*	*	*	*							*	*				

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Опрашиваемый вход для определения переднего фронта сигнала

Описание:
Параллельное подключение входа с опросом по переднему фронту.

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LD	X0	Загрузка контакта X0
ORP	**X1**	Параллельное подключение X1 с опросом по переднему фронту
OUT	Y1	Установка катушки Y1

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
95	ORF	(S)	«ИЛИ» с опросом по заднему фронту				

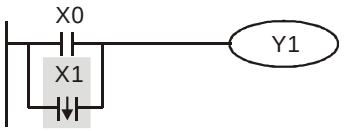
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*							*	*				ORF: 3 шага

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Опрашиваемый вход для определения заднего фронта сигнала

Описание:
Параллельное подключение входа с опросом по заднему фронту.

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LD	X0	Загрузка контакта X0
ORF	**X1**	Параллельное подключение X1 с опросом по заднему фронту
OUT	Y1	Установка катушки Y1

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
96	TMR	(S1) (S2)	16-битный таймер				

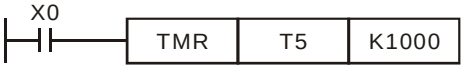
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*					TMR: 5 шагов
S_2					*								*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Адрес таймера (T0~T255) S_2: Заданное значение (K0~K32767, D0~D9999)

Описание:
При выполнении команды TMR будет вестись отсчет времени, пока значение таймера не достигнет заданного значения, после чего будет сгенерирован соответствующий сигнал.

Пример программы:
Релейно-контактная схема:



Коды команды: Операция:
LD X0 Загрузка Н/О контакта X0
TMR T5 K1000 Установка заданного значения таймера T5 = K1000

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
97	CNT	S_1 S_2	16-битный счетчик				

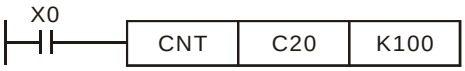
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1												*				CNT: 5 шагов
S_2					*								*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Адрес 16-битного счетчика (C0~C199) **S_2**: Заданное значение (K0~K32 767, D0~D9 999)

Описание:
1. Команды CNT служит для суммирования входных импульсов. При достижении заданного значения будет сгенерирован соответствующий сигнал, после чего входные импульсы уже не влияют на текущее значение и состояние счетчика. Состояние счетчика и таймера может применяться в следующих шагах программы как состояние входного сигнала.
2. Для сброса счетчика используется команда RST.

Пример программы:
Релейно-контактная схема:



Коды команды: Операция:
LD X0 Загрузка Н/О контакта X0
CNT C20 K100 Установка заданного значения счетчика C20 = K100

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
97	DCNT	S_1 S_2	32- битный счетчик				

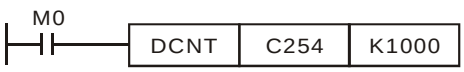
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1												*				DCNT: 9 шагов
S_2					*								*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Адрес 32-битного счетчика (C200~C254)
S_2: Заданное значение (K-2 147 483 648~K2 147 483 647, D0~D9 999)

Описание:
1. Команда DCNT запускает 32-битные счетчики C200 - C254.
2. Для общего подсчета в счетчиках C200~C231(SS2/SA2/SE/SX2: C200~C232) к текущему значению будет прибавляться или отниматься на 1 в соответствии с режимом, установленным флагами M1200-M1231.
3. Высокоскоростные счетчики C232~C254(SS2/SA2/SE/SX2: C233~C254) запускаются с помощью импульса. Для дополнительной информации см. раздел **2.12 C (Счетчики).**
4. Когда команда DCNT завершает работу, счетчик остановится, а его значение будет сохранено. Используйте команду RST для сброса значений счетчика и его состояния, а команду DMOV для перемещения этих значений в регистры. Для высокоскоростных счетчиков C232-C254 используйте специальные внешние входы для сброса значений счетчиков и их состояний.

Пример программы:
Релейно-контактная схема:



Коды команды:			Операция:
LD	M0		Загрузка Н/О контакта M0
DCNT	**C254**	**K1000**	Значение счетчика C254 = K1000

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
98	INV	-	Инверсия				

Операнд	Описание	Шаги программы
НЕТ	Инвертирует текущий результат внутренних операций ПЛК	INV: 1 шаг

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Описание:

Команда INV инвертирует результат предыдущих операций.

Пример программы:
Релейно-контактная схема:



Коды команды:		Операция:
LD	X0	Загрузка Н/О контакта X0
INV		Инвертирование результата операции
OUT	Y1	Установка катушки Y1

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
99	PLF	(S)	Создание импульса по заднему фронту				

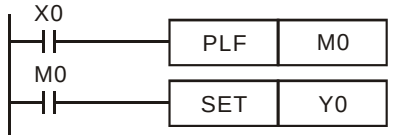
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S		*	*													PLF: 3 шага

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

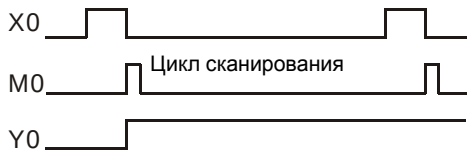
Операнды:
S: Выход для импульса

Описание:
Команда PLS служит для генерации одного импульса – опознание заднего фронта сигнала - независимо от продолжительности входного сигнала для включения соответствующего операнда. Операнд остается во включенном состоянии напротяжении одного цикла программы (скана).

Пример программы:
Релейно-контактная схема:



Временная диаграмма:



Коды команды:		Операция:
LD	X0	Загрузка Н/О контакта X0
PLF	**M0**	Создание импульса по заднему фронту на M0
LD	M0	Загрузка Н/О контакта M0
SET	Y0	Включение Y0

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
100	MODRD	S_1 S_2 n	Чтение данных MODBUS				

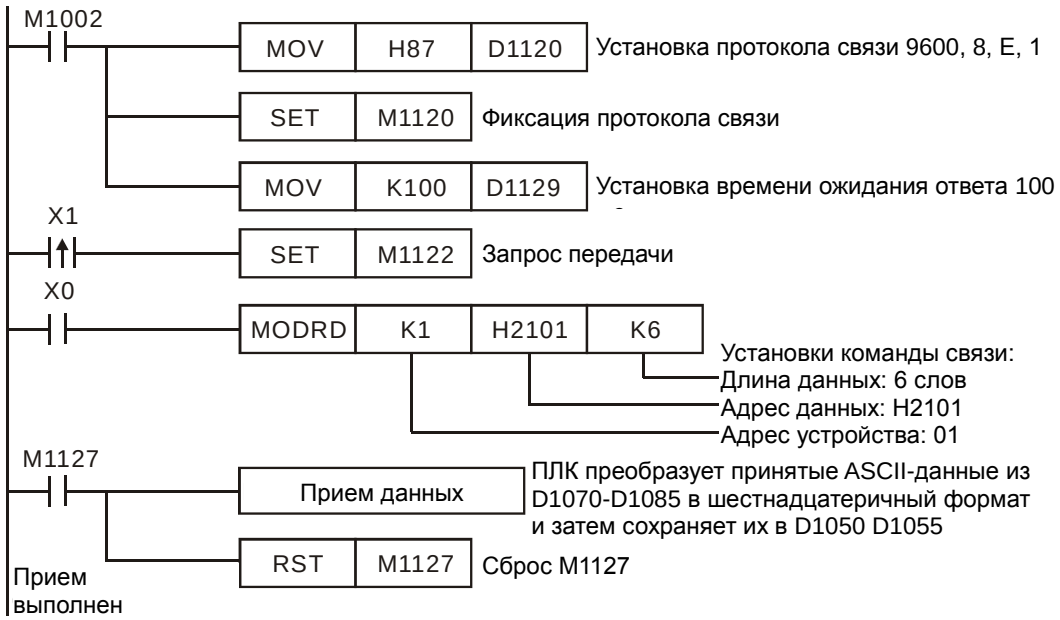
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRD: 7 шагов
S_2					*	*							*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Адрес устройства (K0~K254) **S_2**: Адрес данных **n**: Длина данных (K1<**n**<K6)

Описание:
1. Команда MODRD поддерживает интерфейс COM2 (RS-485).
2. Команда MODRD предназначена для чтения данных с внешних устройств, поддерживающих протокол Modbus (ASCII или RTU) по встроенному в устройства Delta VFD (кроме серии VFD-A) интерфейсу RS-485.
3. Если адрес **S_2** установлен неверно, возникнет ошибка, код которой будет записан в регистр D1130 и будет включен флаг ошибки M1141.
4. Данные, полученные от периферийного устройства, сохраняются в регистры D1070 - D1085. После завершения приема данных ПЛК автоматически проверяет их достоверность, в случае ошибки будет включен флаг M1140.
5. При передаче данных в ASCII-режиме ПЛК конвертирует данные в шестнадцатеричный формат и сохраняет их в D1050 ~ D1055. Регистры D1050 ~ D1055 в RTU-режиме не используются.
6. После M1140 или M1141 = 1, запрос на чтение данных будет отправлен повторно и, если данные будут приняты без ошибок, флаги M1140 и M1141 будут сброшены.
7. Не существует ограничений на количество использований команды. Тем не менее, только одна команда может выполняться на одном COM-порте.
8. Импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF) нельзя использовать с командой MODRD, иначе данные будут приняты некорректно.
9. Описание флагов и специальных регистров см. **Заметки** в описании команды API 80 RS.

Пример программы 1:
Связь ПЛК с преобразователем частоты VFD-B (ASCII-режим, M1143 выключен)



ПЛК → VFD-B , ПЛК передает: “01 03 2101 0006 D4”

VFD-B → ПЛК , ПЛК принимает: “01 03 0C 0100 1766 0000 0000 0136 0000 3B”

Регистры передаваемых данных (отправленные сообщения)

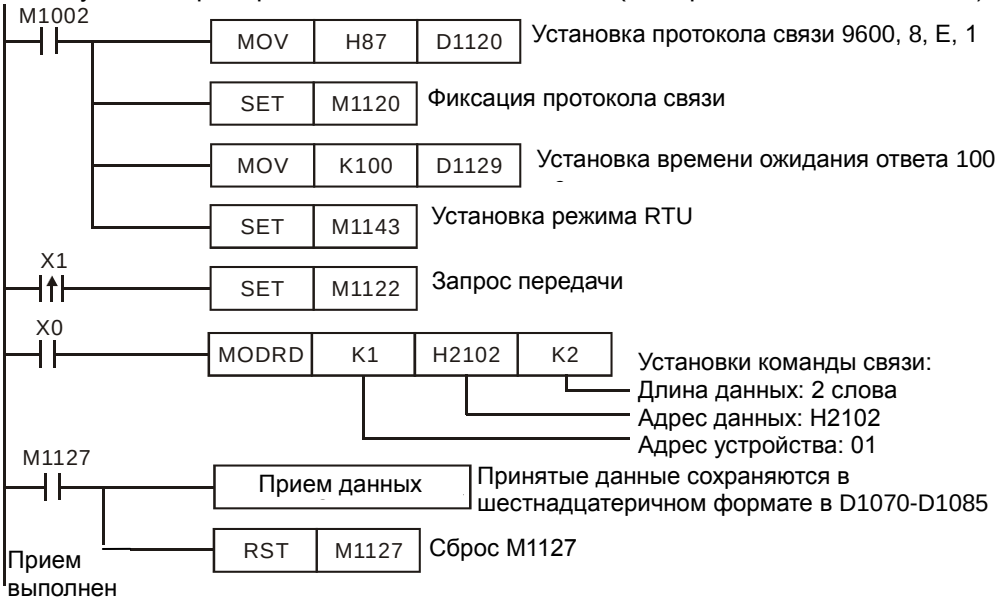
Регистр	Данные		Описание	
D1089 младший байт	‘0’	30 H	ADR 1	Адрес устройства: ADR (1,0)
D1089 старший байт	‘1’	31 H	ADR 0	
D1090 младший байт	‘0’	30 H	CMD 1	Код команды: CMD (1,0)
D1090 старший байт	‘3’	33 H	CMD 0	
D1091 младший байт	2’	32 H	Стартовый адрес данных	
D1091 старший байт	‘1’	31 H		
D1092 младший байт	‘0’	30 H		
D1092 старший байт	‘1’	31 H		
D1093 младший байт	‘0’	30 H	Число данных (слов)	
D1093 старший байт	‘0’	30 H		
D1094 младший байт	‘0’	30 H		
D1094 старший байт	‘6’	36 H		
D1095 младший байт	‘D’	44 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D1095 старший байт	‘4’	34 H	LRC CHK 0	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание	
D1070 младший байт	‘0’	30 H	ADR 1	
D1070 старший байт	‘1’	31 H	ADR 0	
D1071 младший байт	‘0’	30 H	CMD 1	
D1071 старший байт	‘3’	33 H	CMD 0	
D1072 младший байт	‘0’	30 H	Число данных (байт)	
D1072 старший байт	‘C’	43 H		
D1073 младший байт	‘0’	30 H	Содержимое адреса 2101 H	0100 H ПЛК автоматически конвертирует ASCII-код и сохраняет значение в D1050
D1073 старший байт	‘1’	31 H		
D1074 младший байт	‘0’	30 H		
D1074 старший байт	‘0’	30 H		
D1075 младший байт	‘1’	31 H	Содержимое адреса 2102 H	1766 H ПЛК автоматически конвертирует ASCII-код и сохраняет значение в D1051
D1075 старший байт	‘7’	37 H		
D1076 младший байт	‘6’	36 H		
D1076 старший байт	‘6’	36 H		
D1077 младший байт	‘0’	30 H	Содержимое адреса 2103 H	0000 H ПЛК автоматически конвертирует ASCII-код и сохраняет значение в D1052
D1077 старший байт	‘0’	30 H		
D1078 младший байт	‘0’	30 H		
D1078 старший байт	‘0’	30 H		
D1079 младший байт	‘0’	30 H	Содержимое адреса 2104 H	0000 H ПЛК автоматически конвертирует ASCII-код и сохраняет значение в D1053
D1079 старший байт	‘0’	30 H		
D1080 младший байт	‘0’	30 H		
D1080 старший байт	‘0’	30 H		
D1081 младший байт	‘0’	30 H	Содержимое адреса 2105 H	0136 H ПЛК автоматически конвертирует ASCII-код и сохраняет значение в D1054
D1081 старший байт	‘1’	31 H		
D1082 младший байт	‘3’	33 H		
D1082 старший байт	‘6’	36 H		
D1083 младший байт	‘0’	30 H	Содержимое адреса 2106 H	0000 H ПЛК автоматически конвертирует ASCII-код и сохраняет значение в D1055
D1083 старший байт	‘0’	30 H		
D1084 старший байт	‘0’	30 H		
D1084 старший байт	‘0’	30 H		
D1085 младший байт	‘3’	33 H	LRC CHK 1	
D1085 старший байт	‘B’	42 H	LRC CHK 0	

Пример программы 2:

Связь между ПЛК и преобразователем частоты VFD-B (RTU-режим, M1143 включен)



ПЛК → VFD-B , ПЛК передает: 01 03 2102 0002 6F F7
VFD-B → ПЛК, ПЛК принимает: 01 03 04 1770 0000 FE 5C

Регистры передаваемых жанных (передаваемые сообщения)

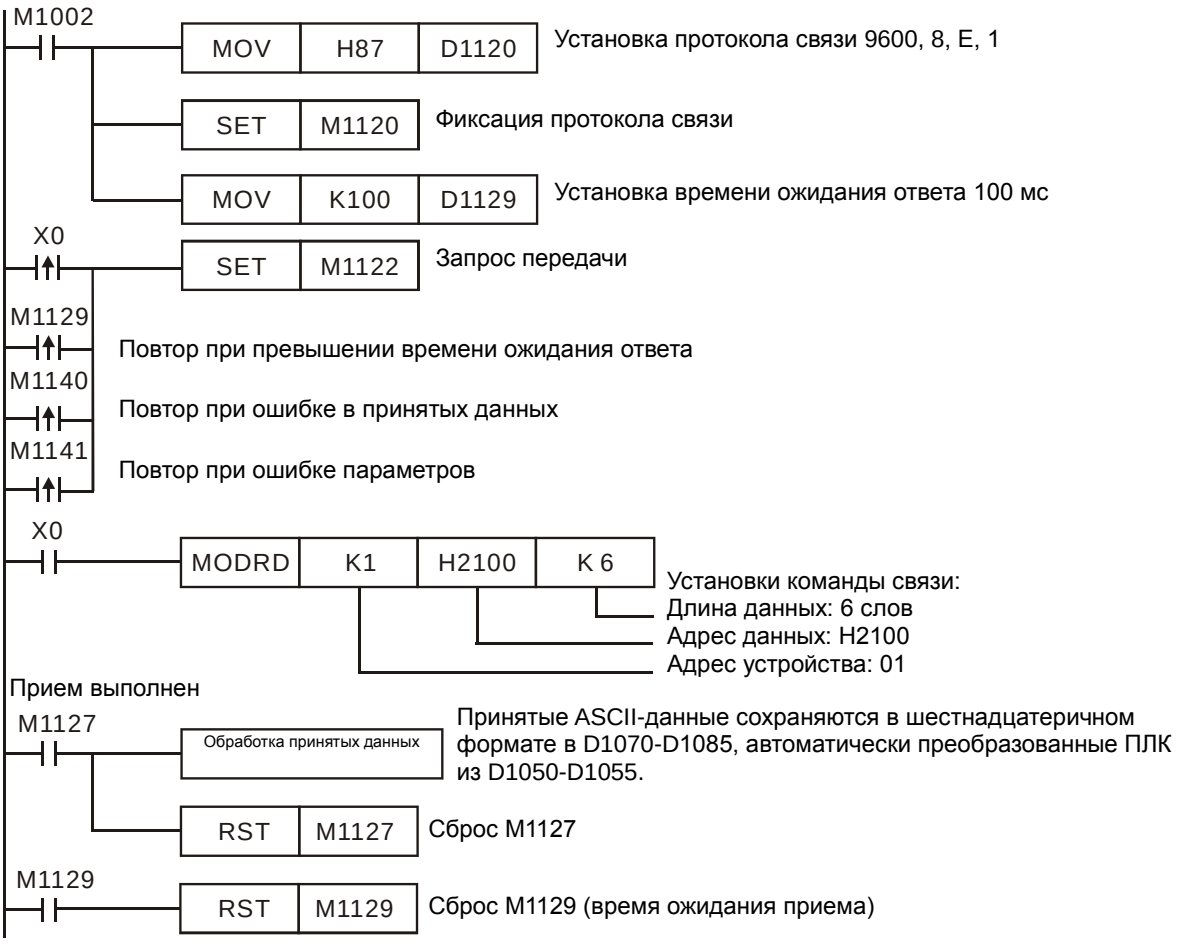
Регистр	Данные	Описание
D1089 младший байт	01 H	Адрес устройства
D1090 младший байт	03 H	Адрес команды
D1091 младший байт	21 H	Стартовый адрес данных
D1092 младший байт	02 H	
D1093 младший байт	00 H	Число данных (слова)
D1094 младший байт	02 H	
D1095 младший байт	6F H	CRC CHK мл.
D1096 младший байт	F7 H	CRC CHK ст.

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные	Описание
D1070 младший байт	01 H	Адрес устройства
D1071 младший байт	03 H	Адрес команды
D1072 младший байт	04 H	Число данных (байт)
D1073 младший байт	17 H	Содержимое регистра с адресом 2102 H
D1074 младший байт	70 H	
D1075 младший байт	00 H	Содержимое регистра с адресом 2103 H
D1076 младший байт	00 H	
D1077 младший байт	FE H	CRC CHK мл.
D1078 младший байт	5C H	CRC CHK ст.

Пример программы 3:

1. При связи между ПЛК и преобразователем частоты VFD-B (ASCII-режим, M1143 выключен) повторная передача при превышении времени ожидания ответа.
2. Когда X0 включен, ПЛК читает данные по адресу H2100 устройства 01(VFD-B) и сохраняет данные в ASCII-формате в регистры D1070 ~ D1085. ПЛК автоматически конвертирует данные и сохраняет их в D1050 ~ D1055.
3. M1129 включается при превышении времени ожидания ответа. Программа включает M1129 и инициирует повторную передачу данных.
4. M1140 включается, если принятые данные содержат ошибку. Программа включает M1140 и инициирует повторную передачу данных.
5. M1141 включается при некорректном адресе данных. Программа включает M1141 и инициирует повторную передачу данных.



API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
101	MODWR	S_1 S_2 n	Запись данных MODBUS				

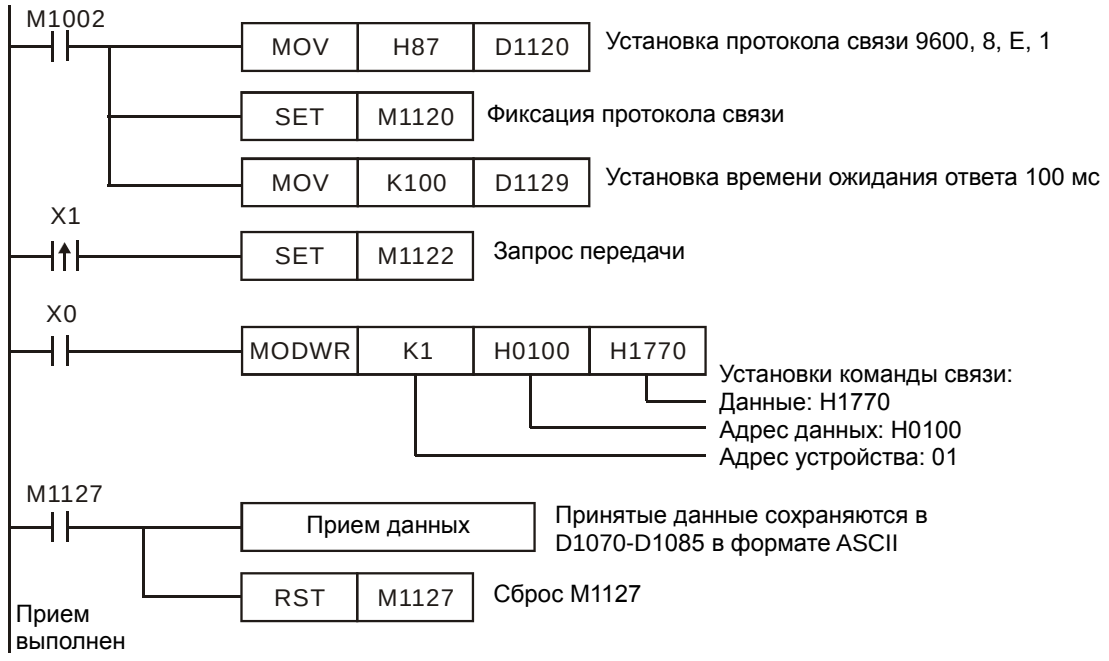
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODWR: 7 шагов
S_2					*	*							*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Адрес устройства (K0~K254) **S_2**: Адрес данных **n**: Данные для записи

Описание:
1. Команда MODWR поддерживает интерфейс COM2 (RS-485).
2. Не существует ограничений на количество использований команды. Тем не менее, только одна команда может выполняться на одном COM-порте.
3. Команда MODWR предназначена для записи данных на внешние устройства, поддерживающие протокол Modbus (ASCII или RTU) по встроенному в устройства Delta VFD (кроме серии VFD-A) интерфейсу RS-485.
4. Если адрес **S_2** установлен неверно, возникнет ошибка, код которой будет записан в регистр D1130 и будет включен флаг ошибки M1141. Например, если 8000H недопустимо для VFD-B, M1141 включится и D1130 = 2.
5. Данные, передаваемые на периферийные устройства, сохраняются в регистры D1070 - D1085. После завершения приема данных ПЛК автоматически проверяет их достоверность, в случае ошибки будет включен флаг M1140.
6. После M1140 или M1141 = 1, данные будут отправлены повторно и, если не будет ошибок, флаги M1140 и M1141 будут сброшены.
7. Если импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF) используются перед командой MODWR, при отправлении запроса флаг M1122 должен быть включен.
8. Описание флагов и специальных регистров см. **Заметки** в описании команды API 80 RS.

Пример программы 1:
Связь ПЛК с преобразователем VFD-B (ASCII-режим, M1143 выключен)



ПЛК → VFD-B, ПЛК передает: “01 06 0100 1770 71”

VFD-B → ПЛК, ПЛК принимает: “01 06 0100 1770 71”

Регистры данных для передачи (переданные сообщения)

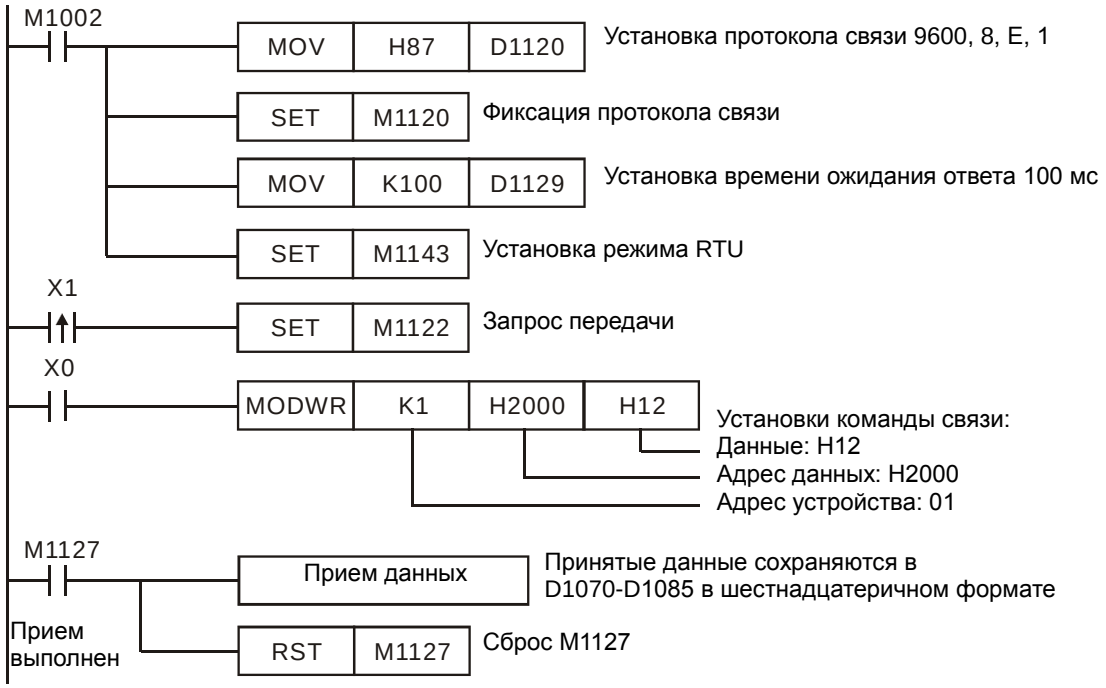
Регистр	Данные		Описание	
D1089 младший байт	‘0’	30 H	ADR 1 ADR 0	Адрес устройства: ADR (1,0)
D1089 старший байт	‘1’	31 H		
D1090 младший байт	‘0’	30 H	CMD 1 CMD 0	Код команды: CMD (1,0)
D1090 старший байт	‘6’	36 H		
D1091 младший байт	‘0’	30 H	Адрес данных	
D1091 старший байт	‘1’	31 H		
D1092 младший байт	‘0’	30 H		
D1092 старший байт	‘0’	30 H		
D1093 младший байт	‘1’	31 H	Содержимое данных	
D1093 старший байт	‘7’	37 H		
D1094 младший байт	‘7’	37 H		
D1094 старший байт	‘0’	30 H		
D1095 младший байт	‘7’	37 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D1095 старший байт	‘1’	31 H	LRC CHK 0	

Регистры данных для приема (принимаемые сообщения)

Регистр	Данные		Описание
D1070 младший байт	‘0’	30 H	ADR 1
D1070 старший байт	‘1’	31 H	ADR 0
D1071 младший байт	‘0’	30 H	CMD 1
D1071 старший байт	‘6’	36 H	CMD 0
D1072 младший байт	‘0’	30 H	Адрес данных
D1072 старший байт	‘1’	31 H	
D1073 младший байт	‘0’	30 H	
D1073 старший байт	‘0’	30 H	
D1074 младший байт	‘1’	31 H	Содержимое данных
D1074 старший байт	‘7’	37 H	
D1075 младший байт	‘7’	37 H	
D1075 старший байт	‘0’	30 H	
D1076 младший байт	‘7’	37 H	LRC CHK 1
D1076 старший байт	‘1’	31 H	LRC CHK 0

Пример программы 2:

Связь ПЛК с преобразователем частоты VFD-B (RTU-режим, M1143 включен)



ПЛК → VFD-B, ПЛК передает: 01 06 2000 0012 02 07

VFD-B → ПЛК, ПЛК принимает: 01 06 2000 0012 02 07

Регистры данных для передачи (передаваемые сообщения)

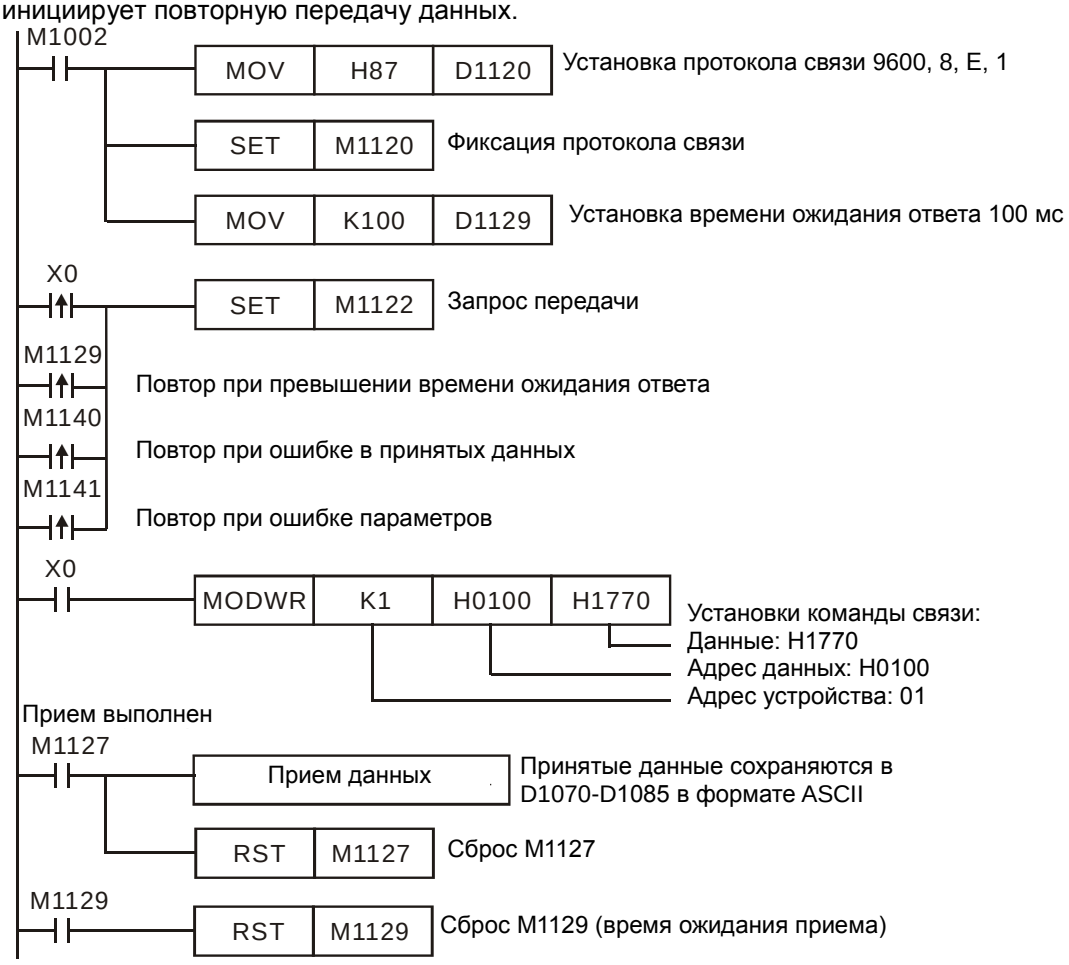
Регистр	Данные	Описание
D1089 младший байт	01 H	Адрес устройства
D1090 младший байт	06 H	Адрес команды
D1091 младший байт	20 H	Адрес данных
D1092 младший байт	00 H	
D1093 младший байт	00 H	Содержимое данных
D1094 младший байт	12 H	
D1095 младший байт	02 H	CRC CHK Мл.
D1096 младший байт	07 H	CRC CHK Ст.

Регистры данных для приема (принимаемые сообщения)

Регистр	Данные	Описание
D1070 младший байт	01 H	Адрес устройства
D1071 младший байт	06 H	Адрес команды
D1072 младший байт	20 H	Адрес данных
D1073 младший байт	00 H	
D1074 младший байт	00 H	Содержимое данных
D1075 младший байт	12 H	
D1076 младший байт	02 H	CRC CHK Мл.
D1077 младший байт	07 H	CRC CHK Ст.

Пример программы 3:

1. При связи между ПЛК и преобразователем частоты VFD-B (ASCII-режим, M1143 выключен) повторная передача при превышении времени ожидания ответа.
2. Когда X0 включен, ПЛК записывает данные H1770 (K6000) в регистр с адресом H0100 в устройстве 01 (VFD-B).
3. M1129 включается при превышении времени ожидания ответа. Программа включает M1129 и инициирует повторную передачу данных.
4. M1140 включается, если принятые данные содержат ошибку. Программа включает M1140 и инициирует повторную передачу данных.
5. M1141 включается при некорректном адресе данных. Программа включает M1141 и инициирует повторную передачу данных.



API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
102	FWD	S1 S2 n	Пуск вперед для VFD				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			FWD: 7 шагов
S_2					*	*							*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

API	Команда	Операнды	Функция	Контроллеры			
103	REV	S_1 S_2 n	Пуск назад для VFD	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			FWD: 7 шагов
S_2					*	*							*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

API	Команда	Операнды	Функция	Контроллеры			
104	STOP	S_1 S_2 n	Стоп для VFD	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			FWD: 7 шагов
S_2					*	*							*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Адрес устройства **S_2**: Заданная частота устройства VFD **n**: Режим работы

Описание:
1. Не существует ограничений на количество использований команд FWD, REV, STOP, но только одна команда может выполняться одновременно на одном COM-порте.
2. Если импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF) выполняются перед командами FWD, REV, STOP, флаги M1122 (COM2) / M1316 (COM3) должны быть включены заранее для корректной работы.
3. Полная информации по флагам и специальным регистрам см. Описание команды RS.

Пример программы: COM2 (RS-485) для связи ПЛК с ПЧ VFD-A.

Повторная передача при превышении времени ожидания и ошибке данных.
M1177 выключен (по умолчанию), поддерживается только Delta VFD-A:

a) Диапазон **S_1**: K0 ~ K31; Диапазон **S_2** для ПЧ VFD-A: K0 ~ K4,000 (0.0 ~ 400.0 Гц).
b) **n** = Режим работы с ПЧ. Диапазон: K1 ~ K2. **n** = 1: команда относится к одному ПЧ. **n** = 2: команда относится ко всем подключенным ПЧ.
c) Ответ ПЧ сохраняется в D1070 ~ D1080. После завершения приема данных, ПЛК автоматически проверяет их корректность. В случае ошибки, включается флаг M1142. При **n** = 2, PLC не будет принимать ответы от ПЧ.



ПЛК ⇨ VFD-A, ПЛК передает: "**C ♥ ☺ 0001 0500**"
VFD-A ⇨ ПЛК, ПЛК принимает: "**C ♥ ♠ 0001 0500**"

Регистры данных для передачи (передаваемые сообщения)

Регистр	Данные		Описание
D1089 младший байт	'C'	43 H	Стартовое слово команды
D1090 младший байт	'♥'	03 H	Контрольная сумма
D1091 младший байт	'☺'	01 H	Подтверждение команды (режим коммуникации)
D1092 младший байт	'0'	30 H	Коммуникационный адрес устройства
D1093 младший байт	'0'	30 H	
D1094 младший байт	'0'	30 H	
D1095 младший байт	'1'	31 H	
D1096 младший байт	'0'	30 H	Рабочая команда
D1097 младший байт	'5'	35 H	
D1098 младший байт	'0'	30 H	
D1099 младший байт	'0'	30 H	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание
D1070 младший байт	'C'	43 H	Стартовое слово команды
D1071 младший байт	'♥'	03 H	Контрольная сумма
D1072 младший байт	'♠'	06 H	Подтверждение ответа (проверка данных обратной связи) (корректно: 06H, ошибка: 07 H)
D1073 младший байт	'0'	30 H	Коммуникационный адрес устройства
D1074 младший байт	'0'	30 H	
D1075 младший байт	'0'	30 H	
D1076 младший байт	'1'	31 H	
D1077 младший байт	'0'	30 H	Рабочая команда
D1078 младший байт	'5'	35 H	
D1079 младший байт	'0'	30 H	
D1080 младший байт	'0'	30 H	

Пример программы: COM2 (RS-485) для связи ПЛК с ПЧ серии VFD-B

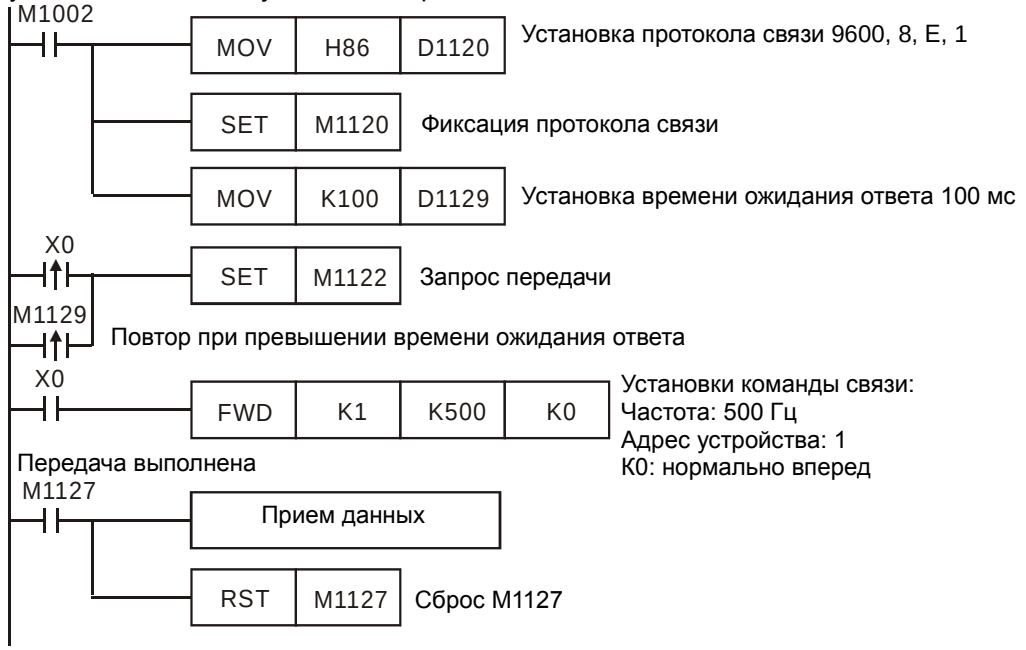
ASCII-режим, M1143 выключен, повторная передача при превышении времени ожидания. M1177 включен, поддерживаются другие модели ПЧ Delta VFD:

a) **S_1 =** Адрес VFD-A. Диапазон **S_1**: K0 ~ K255, когда **S_1** определен как K0, ПЛК будет передавать данные на все подключенные ПЧ VFD.

b) **S_2** = Частота запуска VFD. См. инструкцию по эксплуатации конкретной модели VFD. При выполнении команды STOP операнд **S_2** является резервным.

c) **n =** Режим работы.
- Для команды FWD: **n** = 0 → Режим пуска вперед; **n** = 1 → JOG-режим вперед..
- Для команды REV: **n** = 0 → Режим пуска назад; **n** = 1 → JOG-режим назад.
- Для команды STOP: Операнд **n** является резервным.

d) При выбранном JOG-режиме вперед для команды FWR, заданное значение **S_2** недействительно. Если пользователю необходимо изменить частоту JOG см. руководства по эксплуатации конкретных моделей ПЧ VFD.



ПЛК ⇨ VFD, ПЛК передает: ":**01 10 2000 0002 04 0012 01F4 C2** "
VFD ⇨ ПЛК, ПЛК принимает: ":**01 10 2000 0002 CD** "

Данные для передачи (передаваемые сообщения)

Данные		Описание	
‘0’	30 H	ADR 1 ADR 0	Адрес устройства: ADR (1,0)
‘1’	31 H		
‘1’	31 H	CMD 1 CMD 0	Код команды: CMD (1,0)
‘0’	30 H		
‘2’	32 H	Адрес данных	
‘0’	30 H		
‘0’	30 H		
‘0’	30 H		
‘0’	30 H	Содержимое данных	
‘0’	30 H		
‘0’	30 H		
‘2’	32 H		
‘0’	30 H	Счетчик байт	
‘4’	34 H		
‘0’	30H	Данные 1	H1: пуск вперед
‘0’	30 H		
‘1’	31 H		
‘2’	32 H		
‘0’	30 H	Данные 2	Рабочая частота = K500 Гц H01F4
‘1’	31 H		
‘F’	46 H		
‘4’	34 H		
‘C’	43 H	LRC CHK 1 LRC CHK 0	Ошибка контрольной суммы: LRC CHK (0,1)
‘2’	32 H		

Принимаемые данные (принимаемые сообщения)

Данные		Описание
‘0’	30 H	ADR 1 ADR 0
‘1’	31 H	
‘1’	31 H	CMD 1 CMD 0
‘0’	30 H	
‘2’	32 H	Адрес данных
‘0’	30 H	
‘0’	30 H	
‘0’	30 H	
‘0’	30 H	Число регистров
‘0’	30 H	
‘0’	30 H	
‘2’	32 H	
‘C’	43 H	LRC CHK 1
‘D’	44 H	LRC CHK 0

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/ SE	SX2
105	RDST	(S) (n)	Чтение текущего состояния VFD				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			RDST: 5 шагов
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Адрес устройства **n**: Прочитанное значение состояния

Описание:

1. M1177 выключен (по умолчанию): команда RDST поддерживает порт COM2 (RS-485). Команда поддерживает ПЧ Delta VFD-A. Диапазон S: K0 ~ K31. n: Объект состояния Диапазон n: K0 ~ K3 (n=0 для частоты; n=1 для выходной частоты; n=2 для выходного тока; n=3 для режим работы)
2. Данные обратной связи (ответа) занимают 11 байт (см. инструкцию по эксплуатации VFD-A) и сохраняются в младших байтах D1070 ~ D1080.
3. M1177 включен: команда RDST поддерживает порты COM2 (RS-485), COM3 (RS-485). Команда поддерживает другие модели ПЧ, т.е. VFD-B, VFD-C, VFD-CP.
4. Не существует ограничений на количество использований команды RDST, но только одна команда может выполняться одновременно на одном COM-порте.
5. Если импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF) выполняются перед командой RDST, флаги M1122 (COM2) / M1316 (COM3) должны быть включены заранее для корректной работы.

6. Полная информация по флагам и специальным регистрам см. Описание команды RS.
7. M1177 выключен, поддерживается только ПЧ VFD-A:
   - ”**Q, S, B, Uu, Nn, ABCD**”

Ответ	Описание	Хранение данных
Q	Стартовое слово: 'Q' (51H).	D1070 младший байт
S	Контрольная сумма: 03H.	D0171 младший байт
B	Подтверждение ответа. Корректный: 06H, Ошибка: 07H.	D1072 младший байт
U	Коммуникационный адрес (диапазон: 00~31). Отображается в ASCII-формате.	D1073 младший байт
U		D1074 младший байт
N	Значение состояния (00 ~ 03). В ASCII-формате.	D1075 младший байт
N		D1076 младший байт
A	Данные состояния. Содержимое ”ABCD” различно в зависимости от значения 00~03, установленного в NN. 00 ~ 03 указывает частоту, ток и режим соответственно. См. ниже:	D1077 младший байт
B		D1078 младший байт
C		D1079 младший байт
D		D1080 младший байт

Nn = “00” Заданная частота = ABC.D (Hz)
Nn = “01” Выходная частота = ABC.D (Hz)
Nn = “02” Выходной ток = ABC.D (A)
ПЛК автоматически конвертирует принятые данные в ASCII-коде в шестнадцатеричное значение и сохраняет в D1050. Например, ”ABCD” = “0600”, ПЛК конвертирует в K0600 (0258 H) и сохраняет в специальном регистре D1050.

Nn = “03” Режим работы

‘A’ =

‘0’	Стоп	‘5’	JOG (вперед)
‘1’	Пуск вперед	‘6’	JOG (назад)
‘2’	Стоп	‘7’	JOG (назад)
‘3’	Пуск назад	‘8’	Сбой
‘4’	JOG (вперед),		

ПЛК автоматически конвертирует принятые данные в ASCII-коде в шестнадцатеричное значение и сохраняет в D1050. Например, ”A” = “3”, ПЛК конвертирует A в K3 и сохраняет в D1051.

‘B’ =

b7	b6	b5	b4	Источник заданной частоты
0	0	0	0	Клавиатура
0	0	0	1	Фиксированная скорость 1
0	0	1	0	Фиксированная скорость 2
0	0	1	1	Фиксированная скорость 3
0	1	0	0	Фиксированная скорость 4
0	1	0	1	Фиксированная скорость 5
0	1	1	0	Фиксированная скорость 6
0	1	1	1	Фиксированная скорость 7
1	0	0	0	JOG-частота
1	0	0	1	Аналоговая команда
1	0	1	0	RS-485
1	0	1	1	Команда Up/Down

b3	=	0	Нет торможения	1	Торможение пост. током при останове
b2	=	0	Нет торможения	1	Торможение постоянным током при старте
b1	=	0	Вперед	1	Назад
b0	=	0	Стоп	1	Пуск

ПЛК сохраняет ”B” в M1168 (b0) ~ M1175 (b7).

“CD” =

“00”	No error	“10”	OcA
“01”	oc	“11”	Ocd
“02”	ov	“12”	Ocn
“03”	oH	“13”	GFF
“04”	oL	“14”	Lv
“05”	oL1	“15”	Lv1
“06”	EF	“16”	cF2
“07”	cF1	“17”	bb
“08”	cF3	“18”	oL2
“09”	HPF	“19”	

ПЛК автоматически конвертирует принятые данные в ASCII-коде в шестнадцатеричное значение и сохраняет в D1052. Например, ”CD” = “16”, ПЛК конвертирует CD в K16 и сохраняет в D1052

8. M1177 включен, поддерживаются другие модели ПЧ Delta VFD:
   a) Диапазон $\mathbf{S_1}$: K1 ~ K255
   b) Команда будет читать состояние VFD по адресу параметров 2100H~2104H (см. Инструкцию по эксплуатации конкретной модели VFD) и сохранять данные ответа в D1070~D1074. Однако содержимое D1070~D1074 не будет обновляться при получении ошибки или превышения времени ожидания. Поэтому необходимо заранее проверять состояние флагов.

Пример программы: COM2 (RS-485) для связи ПЛК с ПЧ серии VFD-B

ASCII-режим, M1143 выключен. Повторная отправка при превышении времени ожидания.

1. Чтение состояния VFD по адресу параметров 2100H~2104H и сохранение данных в D1070 ~ D1074.



ПЛК ⇨ VFD-B, ПЛК отправляет: "**:01 03 2100 0005 D6**"
VFD-B ⇨ ПЛК, ПЛК принимает: ":01 03 0A 00C8 7C08 3E00 93AB 0000 2A"

Данные для передачи (переданные сообщения)

Данные		Описание	
'0'	30 H	ADR 1 ADR 0	Адрес устройства: ADR (1,0)
'1'	31 H		
'0'	30 H	CMD 1 CMD 0	Командный код: CMD (1,0)
'3'	33 H		
2'	32 H	Адрес начальных данных	
'1'	31 H		
'0'	30 H		
'0'	30 H		
'0'	30 H	Размер данных (по словам)	
'0'	30 H		
'0'	30 H		
'5'	35 H		
'D'	44 H	LRC CHK 1 LRC CHK 0	Ошибка контрольной суммы: LRC CHK (0,1)
'6'	36 H		

Принятые данные (принятые сообщения)

Данные		Описание	
'0'	30 H	ADR 1 ADR 0	
'1'	31 H		
'0'	30 H	CMD 1 CMD 0	
'3'	33 H		
'0'	30 H	Размер данных (по байтам)	
'A'	41 H		
'0'	30 H	Содержимое адреса 2100 H	ПЛК автоматически конвертирует ASCII-коды и сохраняет результат в D1070 = 00C8 H
'0'	30 H		
'C'	43 H		
'8'	38 H		
'7'	37 H	Содержимое адреса 2101 H	ПЛК автоматически конвертирует ASCII-коды и сохраняет результат в D1071 = 7C08 H
'C'	43 H		
'0'	30 H		
'8'	38 H		
'3'	33 H	Содержимое адреса 2102 H	ПЛК автоматически конвертирует ASCII-коды и сохраняет результат в D1072 = 3E00 H
'E'	45 H		
'0'	30 H		
'0'	30 H		
'9'	39 H	Содержимое адреса 2103H	ПЛК автоматически конвертирует ASCII-коды и сохраняет результат в D1073 = 93AB H
'3'	33 H		
'A'	41 H		
'B'	42 H		
'0'	30 H	Содержимое адреса 2104 H	ПЛК автоматически конвертирует ASCII-коды и сохраняет результат в D1074 = 0000 H
'0'	30 H		
'0'	30 H		
'0'	30 H		
'2'	32 H	LRC CHK 1 LRC CHK 0	
'A'	41 H		

API	Команда	Операнды	Функция	Контроллеры			
106	RSTEF	(S) (n)	Сброс для VFD	ES2/EX2	SS2	SA2/SE	SX2

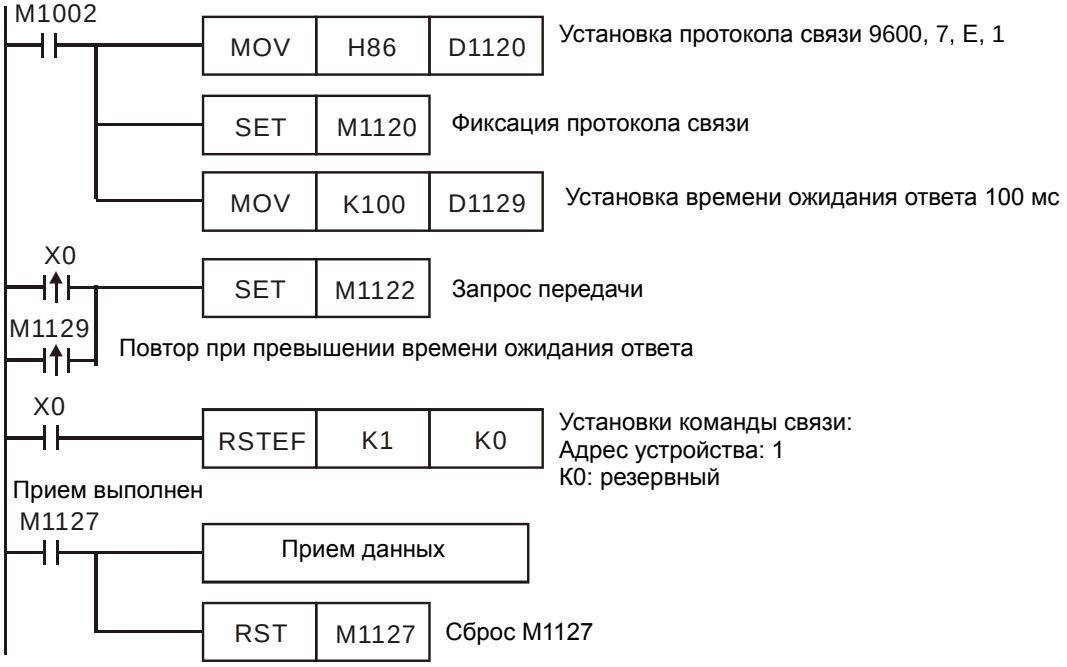
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			RSTEF: 5 шагов
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Адрес устройства **n**: Режим работы

Описание:
1. Когда M1177 выключен (по умолчанию), команда RSTEF поддерживает порт COM2(RS-485) и ПЧ Delta VFD-A. Когда M1177 включен, эта команда поддерживает порты COM2(RS-485), COM3(RS-485) и другие модели ПЧ VFD, т.е. VFD-B, VFD.
2. Не существует ограничений на количество использований команды RSTEF, но только одна команда может выполняться одновременно на одном COM-порте.
3. Если импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF) выполняются перед командой RSTEF, флаги M1122 (COM2) / M1316 (COM3) должны быть включены заранее для корректной работы.
4. Полная информация по флагам и специальным регистрам см. Описание команды RS.
5. M1177 выключен, поддерживается только ПЧ VFD-A, значения операндов:
   a) $\mathbf{S_1}$ = Адрес VFD-A. Диапазон $\mathbf{S_1}$: K0 ~ K31
   b) **n** = Режим работы. Диапазон: K1 ~ K2. **n** = 1: связь с конкретным устройством VFD. **n** = 2: связь со всеми подключенными устройствами VFD.
   c) Команда RSTEF используется для аварийной блокировки ПЧ при возникновении сбоя.
   d) Данные обратной связи (ответ) от периферийных устройств сохраняется в D1070 ~ D1080. Когда **n** = 2, ПЛК не будет принимать ответ от ПЧ.
6. M1177 включен, поддерживаются другие модели Delta: $\mathbf{S_1}$ = Адрес VFD. Диапазон $\mathbf{S_1}$: K0 ~ K255, когда $\mathbf{S_1}$ определен как K0, ПЛК будет передавать данные на все подключенные ПЧ

Пример программы: COM2 (RS-485)
Связь ПЛК с VFD-B (ASCII-режим, M1143 выключен). Повторная передача данных при превышении времени ожидания ответа.



ПЛК ⇨ VFD, ПЛК передает: ":**01 06 2002 0002 D5** "
VFD ⇨ ПЛК, ПЛК принимает: ":**01 06 2002 0002 D5** "

Данные для передачи (переданные сообщения):

Данные		Описание	
‘0’	30 H	ADR 1 ADR 0	Адрес устройства: ADR (1,0)
‘1’	31 H		
‘0’	30 H	CMD 1 CMD 0	Командный код: CMD (1,0)
‘6’	36 H		
‘2’	32 H	Адрес данных	
‘0’	30 H		
‘0’	30 H		
‘2’	32 H		
‘0’	30 H	Содержимое данных	
‘0’	30 H		
‘0’	30 H		
‘2’	32 H		
‘D’	44 H	LRC CHK 1	Ошибка контрольной суммы: LRC CHK (0,1)
‘5’	35 H	LRC CHK 0	

Принятые данные (принятые сообщения)

Данные		Описание
‘0’	30 H	ADR 1
‘1’	31 H	ADR 0
‘0’	30 H	CMD 1
‘6’	36 H	CMD 0
‘2’	32 H	Адрес данных
‘0’	30 H	
‘0’	30 H	
‘2’	32 H	
‘0’	30 H	Содержимое данных
‘0’	30 H	
‘0’	30 H	
‘2’	32 H	
‘D’	44 H	LRC CHK 1
‘5’	35 H	LRC CHK 0

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
107		LRC	P	(S) (n) (D)	Расчет контрольной суммы LRC				

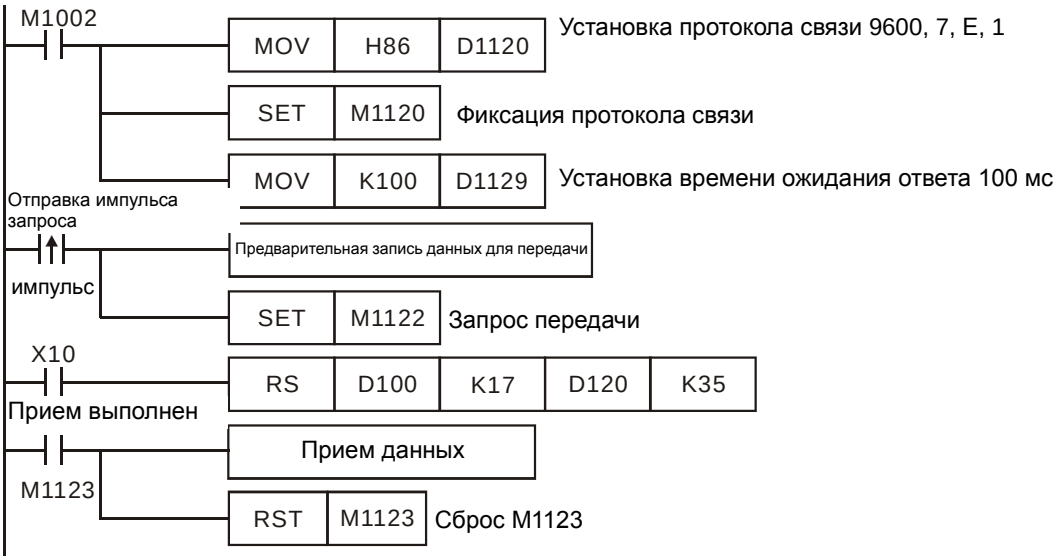
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			LRC, LRCP: 7 шагов
n					*	*							*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Стартовый адрес проверяемых данных в ASCII-режиме **n**: Размер данных для LRC (**n** = K1~K256) **D**: Стартовый адрес хранения результата

Описание:
1. **n** должен быть четным. Если **n** находится за пределами диапазона, будет отображена ошибка, и команда выполняться не будет. При этом включаются M1067 и M1068 и код ошибки H’0E1A записывается в D1067.
2. 16-битный режим: Когда команда LRC выполняется с выключенным M1161, шестнадцатеричные данные, начиная с **S** будут разделены на младшие и старшие биты, команда выполняется для **n** байт. Результат сохраняется в старшие и младшие байты операнда результата **D**.
3. 8-битный режим: Когда команда LRC выполняется с включенным M1161, шестнадцатеричные данные, начиная с **S** будут разделены на младшие и старшие (не используются) биты, команда выполняется для **n** младших байт. Результат сохраняется в младшие байты операнда результата **D** (2 последовательных регистра).
4. Флаг: M1161 переключение 8/16-битный режим

Пример программы:
Связь ПЛК с VFD (ASCII-режим, M1143 выключен), (8-битный режим, M1161 включен), Запись данных, которые будут отправлены в регистры, начиная с D100, заранее для чтения 6 адресов с H0708 на VFD.



ПЛК ⇨ VFD, ПЛК передает: **“: 01 03 07 08 0006 E7 CR LF ”**

Регистры данных для передачи (передаваемые сообщения)

Регистр	Данные		Описание	
D100 младший байт	‘: ’	3A H	STX	
D101 младший байт	‘0’	30 H	ADR 1	Адрес устройства: ADR (1,0)
D102 младший байт	‘1’	31 H	ADR 0	
D103 младший байт	‘0’	30 H	CMD 1	Командный код: CMD (1,0)
D104 младший байт	‘3’	33 H	CMD 0	
D105 младший байт	‘0’	30 H	Стартовый адрес данных	
D106 младший байт	‘7’	37 H		
D107 младший байт	‘0’	30 H		
D108 младший байт	‘8’	38 H		
D109 младший байт	‘0’	30 H	Размер данных (слова)	
D110 младший байт	‘0’	30 H		
D111 младший байт	‘0’	30 H		
D112 младший байт	‘6’	36 H		
D113 младший байт	‘E’	45 H	LRC CHK 0	Ошибка контрольной суммы: LRC CHK (0,1)
D114 младший байт	‘7’	37 H	LRC CHK 1	
D115 младший байт	CR	D H	END	
D116 младший байт	LF	A H		

Контрольную сумму LRC CHK (0, 1) можно рассчитать командой LRC следующим образом (8-битный режим, M1161 включен):



Контрольная сумма LRC: 01 H + 03 H + 07 H + 08 H + 00 H + 06 H = 19 H. 2 дополнение отрицания 19H = E7H. Сохраняем ‘E’(45 H) в младшем байте D113 и ‘7’ (37 H) в младшем байте D114.

Заметки:

Данные коммуникации, ASCII-режим:

STX	‘: ’	Стартовое слово = ‘: ’ (3AH)
Address Hi	‘ 0 ’	Связь:
Address Lo	‘ 1 ’	8-битный адрес, состоящий из 2 ASCII-кодов
Function Hi	‘ 0 ’	Функциональный код:
Function Lo	‘ 3 ’	8-битная функция, состоящая из 2 ASCII-кодов
DATA (n-1)	‘ 2 ’	Содержимое данных:
…….	‘ 1 ’	n × 8-битные данные, состоящие из 2n ASCII-кодов
DATA 0	‘ 0 ’	
	‘ 2 ’	
	‘ 0 ’	
	‘ 0 ’	
	‘ 0 ’	
	‘ 2 ’	
LRC CHK Hi	‘ D ’	Контрольная сумма LRC:
LRC CHK Lo	‘ 7 ’	8-битная контрольная сумма, состоящая из 2 ASCII-кодов
END Hi	CR	Конечное слово:
END Lo	LF	END Ст. = CR (0DH), END Мл.= LF(0AH)

Контрольная сумма LRC: 2 дополнение отрицания суммированного значения до конца данных: 01 H + 03 H + 21 H + 02 H + 00 H + 02 H = 29 H, результат 29H равен D7H.

API	Команда			Операнды	Функция	Контроллеры			
108		CRC	P	S n D	Расчет контрольной суммы CRC	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			CRC, CRCP: 7 шагов
n					*	*							*			
D													*			

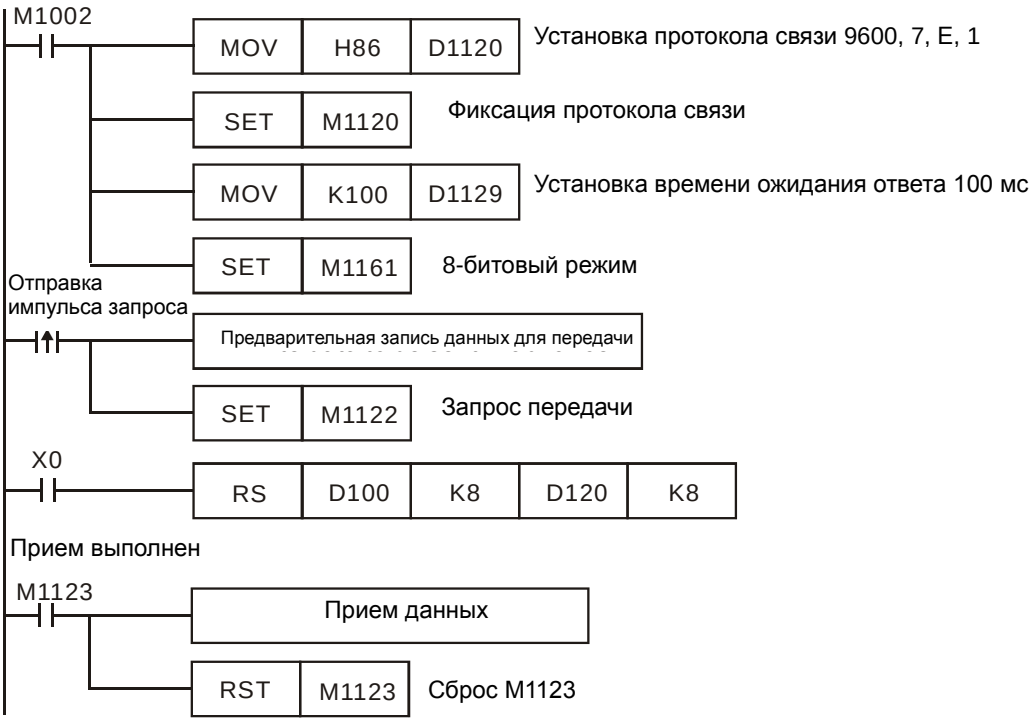
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Стартовый адрес для расчета контрольной суммы в RTU-режиме **n**: Размер данных для команды CRC (**n** = K1~K256) **D**: Стартовый адрес хранения результата

Описание:

1. **n** должен быть четным. Если **n** находится за пределами диапазона, будет отображена ошибка, и команда выполняться не будет. При этом включаются M1067 и M1068 и код ошибки H'0E1A записывается в D1067.
2. 16-битный режим: Когда команда CRC выполняется с выключенным M1161, шестнадцатеричные данные, начиная с **S** будут разделены на младшие и старшие биты, команда выполняется для **n** байт. Результат сохраняется в старшие и младшие байты операнда результата **D**.
3. 8-битный режим: Когда команда CRC выполняется с включенным M1161, шестнадцатеричные данные, начиная с **S** будут разделены на младшие и старшие (не используются) биты, команда выполняется для **n** младших байт. Результат сохраняется в младшие байты операнда результата **D** (2 последовательных регистра).
4. Флаг: M1161 переключение 8/16-битный режим

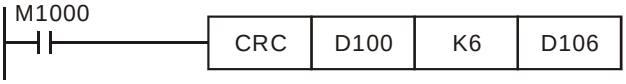
Пример программы: Связь ПЛК с ПЧ (RTU-режим, M1143 = 1), (8-битный режим, M1161 = 1)



ПЛК ⇨ VFD, ПЛК передает: **01 06 0706 1770 66 AB**

Регистры данных для передачи (передаваемые сообщения)

Регистр	Данные	Описание
D100 младший байт	01 H	Адрес
D101 младший байт	06 H	Функция
D102 младший байт	07 H	Адрес данных
D103 младший байт	06 H	
D104 младший байт	17 H	Содержимое данных
D105 младший байт	70 H	
D106 младший байт	66 H	CRC CHK 0
D107 младший байт	AB H	CRC CHK 1



Контрольная сумма CRC CHK (0,1) может быть рассчитана командой CRC (8-битный режим, M1161 включен).

Контрольная сумма CRC: 66 H сохраняется в младшем байте D106 и AB H в младшем байте D107.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
110	D	ECMP	P	S_1 S_2 D	Сравнение чисел с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DECMP, DECMPP: 13 шагов
S_2					*	*							*			
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

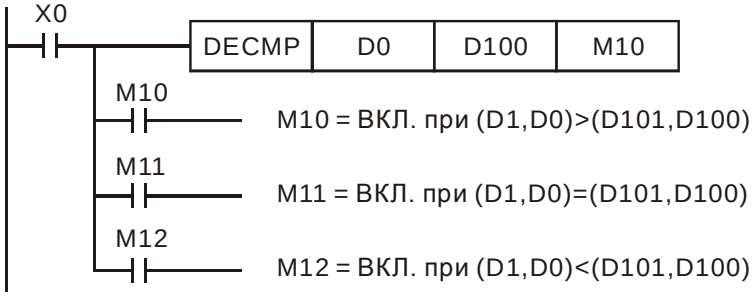
Операнды:

S_1: 1-е сравниваемое значение **S_2**: 2-е сравниваемое значение **D**: Результат сравнения, 3 последовательных адреса

Описание:

1. Данные **S_1** сравниваются с данными **S_2** и результат ($>$, $=$, $<$) отображается в трех последовательных операндах **D**.
2. Если в качестве исходного значения **S_1** или **S_2** выступает константа K или H, значение автоматически конвертируется в двоичное с плавающей запятой.

Пример программы:



1. Если указано устройство M10, M10~M12 используются автоматически.
2. Когда X0 включен, один из M10~M12 включается. Когда X0 выключен, команда DECMP не выполняется, M10~M12 сохраняет свое состояние, предыдущее выключению X0.
3. Можно комбинировать M10~M12 для отображения результата в виде ≤, ≥, ≠.
4. Команды RST или ZRST могут быть применены для сброса результата сравнения.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
111	D	EZCP	P	S_1 S_2 S D	Зонное сравнение чисел с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEZCP, DEZCPP: 17 шагов
S_2					*	*							*			
S					*	*							*			
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Нижняя граница зоны сравнения **S_2**: Верхняя граница зоны сравнения
S: Сравнительное значение **D**: Результат (3 последовательных операнда)

Описание:

1. Данные **S** сравниваются с данными диапазона **S_1** ~ **S_2** и результат (**>** , **=** , **<**) отображается в 3-х последовательных операндах **D**.
2. Если исходный операнд **S_1** или **S_2** имеет значение константы K или H, константа автоматически конвертируется в двоичное число с плавающей запятой.

3. Операнд S_1 должен быть меньше операнда S_2. Если $S_1 > S_2$, команда определяет S_2 как 1[e] сравниваемое значение и проводит операцию в нормальном режиме.

Пример программы:



1. При назначенном устройстве M10, M10~M12 используются автоматически.
2. Когда X0 включен, один из M10~M12 включается. Когда X0 выключен, команда DEZCP не выполняется, M10~M12 сохраняет свое состояние, предыдущее выключению X0.
3. Команды RST или ZRST можно применять для сброса результата сравнения.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
112	D	MOVR	P	(S) (D)	Перемещение числа с плавающей запятой				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S																DMOVR, DMOVRP: 9 шагов
D								*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат

Описание:
1. Непосредственный ввод значения с плавающей запятой в **S**.
2. После выполнения команды значение **S** будет перемещено в **D**.

Пример программы:



При выключенном X0 значения D10 и D11 не меняются. Когда X0 включен, происходит перемещение F1.200E+0 (вводимое число F1.2 и экспоненциальное значение F1.200E+0 отображено на релейно-контактной схеме. Пользователь может установить формат отображаемых данных с помощью функции View) в D10 и D11.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
114		MUL16 MUL32	P	(S_1) (S_2) (D)	16-битное умножение 32-битное умножение				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		MUL16, MUL16P: 7 шагов MUL32, MUL32P: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*		
D								*	*	*	*	*	*	*		

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Множимое значение S_2: Множитель **D**: Результат умножения

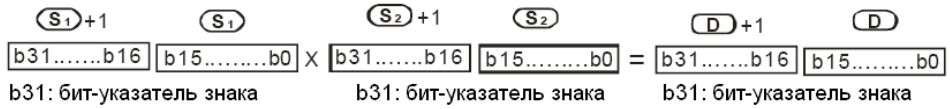
Описание:
1. MUL16 и MUL16P являются 16-битными командами. MUL32 и MUL32P – 32-битными.
2. Двоичное значение из **S1** перемножается с двоичным значением из **S2**, результат сохраняется в **D**. Операция осуществляется по обычным алгебраическим правилам.
3. Если бит – указатель знака, имеет значение 0, то результат положительный. Если бит – указатель знака, имеет значение 1, результат отрицательный.
4. Команды поддерживаются ПЛК серий DVP-ES2/EX2 v. 3.22, DVP-SS2 v. 3.20, DVP-SA2/SX2 v. 2.66, DVP-SE v.1.60 (и выше).

5. 16-битное двоичное умножение:



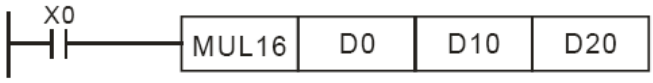
16-битное значение x 16-битное значение = 16-битное значение
Если **D** в битовом формате, K1~K4 сохраняют 16-битный результат. **D** занимает только 16 бит.

6. 32-битное двоичное умножение:



32-битное значение x 32-битное значение = 32-битное значение
Если **D** в битовом формате, K1~K8 сохраняют 32-битный результат. **D** занимает только 32 бит.

Пример программы 1:



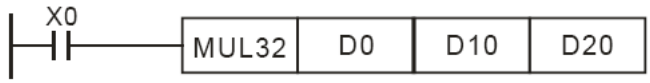
16-битное значение x 16-битное значение = 16-битное значение

⇨ D0 × D10 = D20

⇨ D0=K100, D10=K200, D20=K20,000

16-битное значение K100 из D0 перемножается с 16-битным значением K200 из D10, результат сохраняется в D20. Если бит 15 в D20 = 0, результат в D20 имеет положительное значение. Если бит 15 в D20 = 1, то отрицательное.

Пример программы 2:



32-битное значение x 32-битное значение = 32-битное значение

⇨ (D1,D0) ×(D11,D10) = (D21,D20)

⇨ (D1,D0)=K10,000, (D11,D10)=K20,000, (D21, D20)=K200,000,000

32-битное значение K10000 из (D1, D0) перемножается с 32-битным значением K20000 из (D11, D10), результат сохраняется в (D21, D20). Если бит 31 в (D21, D20) равен 0, результат в (D21, D20) имеет положительное значение. Если бит 31 в (D21, D20) равен 1, то отрицательное.

Примечание:

1. Если результат 16-битного умножения не является 16-битным и больше максимального 16-битного значения (K32767) или меньше минимального 16-битного значения (K-32767), сохраняются младшие 16 бит результата и включается флаг переноса M1022.
2. Если результат 32-битного умножения не является 32-битным и больше макс. 32-битного значения (K2147483647) или меньше мин. 32-битного значения (K-2147483647), сохраняются младшие 32 бит результата и включается флаг переноса M1022.
3. Если пользователю необходим полный результат 16-битного умножения (32-битное значение) или 32-битного умножения (64-битное значение), необходимо использовать команды API22 DMUL/DMULP. См. описание этих команд.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
115		DIV16 DIV32	P	S1 S2 D	16-битное деление 32-битное деление				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*		MUL16, MUL16P: 7 шагов
S_2					*	*	*	*	*	*	*	*	*	*		MUL32, MUL32P: 13 шагов
D								*	*	*	*	*	*	*		

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Делимое **S_2**: Делитель **D**: Результат (Частное)

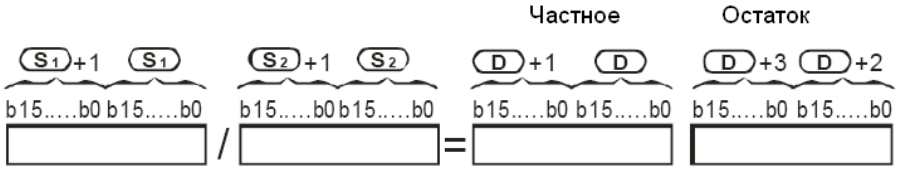
Описание:

1. DIV16 и DIV16P являются 16-битными командами. DIV32 и DIV32P – 32-битными.
2. Двоичное значение из **S1** делится на двоичное значение из **S2**, результат сохраняется в **D**. Операция осуществляется по обычным алгебраическим правилам. Следует следить за знаком S1,S2 и D при выполнении 16- и 32-битных команд.
3. Команда не будет выполнена, если делитель равен 0. Флаги M1067 и M1068 будут включены, а в D1067 будет записан код ошибки H0E19 (в шестнадцатеричном коде).
4. Команды поддерживаются ПЛК серий DVP-ES2/EX2 v. 3.22, DVP-SS2 v. 3.20, DVP-SA2/SX2 v. 2.66, DVP-SE v.1.60 (и выше).
5. 16-битное двоичное деление:



Если **D** в битовом формате, K1~K4 сохраняют 16-битный результат. **D** занимает только 16 бит.

6. 32-битное двоичное деление:



Если **D** в битовом формате, K1~K8 сохраняют 32-битный результат. **D** занимает только 32 бит.

Пример программы 1:



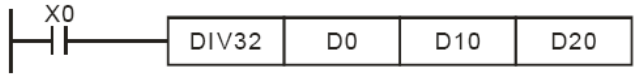
D0/D10=D20

⇨ K103/K5=K20, остаток K3

⇨ D20=K20 (остаток не учтен)

При включенном X0 делимое K103 из D0 делится на делитель K5 из D10, результат сохраняется в D20. Знак результата показывает бит-указатель знака (левый бит b15 из D20):

Пример программы 2:



(D1, D0)/(D11, D10)=(D21, D20)

⇨ K81,000/K40,000=K2, остаток K1000

⇨ (D21, D20)=K2 (остаток не учтен)

При включенном X0 делимое K81000 из (D1, D0) делится на делитель K40000 из (D11, D10), результат сохраняется в (D21, D20). Знак результата показывает бит-указатель знака (левый бит b31 из (D21, D20)):

Примечание:
Если пользователю необходим полный результат 16-битного деления (32-битное значение) или 32-битного деления (64-битное значение), необходимо использовать команды API22 DIV/DIVP. См. описание этих команд.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
116	D	RAD	P	S D	Перевод градусов в радианы				

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S					*	*							*		
D													*		

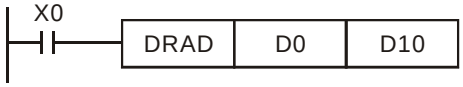
Шаги программы: DRAD, DRADP: 9 steps

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

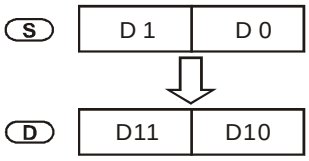
Операнды:
S: Источник данных (градусы) **D**: Результат преобразования (радианы)

Описание:
1. Преобразование осуществляется по формуле: Рад = град × (π/180)
2. Флаги: Флаг нуля M1020, флаг заимствования M1021, флаг переноса M1022.
   Если абсолютное значение результата превышает макс. значение числа с плавающей запятой, включается флаг переноса M1022.
   Если абсолютное значение результата меньше мин. значения числа с плавающей запятой, включается флаг заимствования M1021.
   Если результат равен 0, включается флаг нуля M1020.

Пример программы:



Когда X0 вкл., происходит преобразование значения в градусах двоичного числа с плавающей запятой из (D1, D0) в радианы и результат сохраняется в том же формате в (D11, D10).



Градусы, двоичное значение с плавающей запятой

Радианы (градусы x π/180), двоичное значение с плавающей запятой

API	Команда			Операнды	Функция	Контроллеры			
117	D	DEG	P	S D	Перевод радианов в градусы	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DDEG, DDEGP: 9 шагов
D													*			

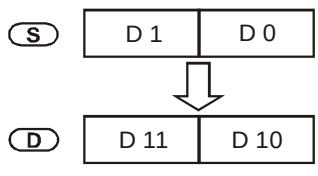
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S: Источник данных (радианы) **D**: Результат (градусы)

Описание:
1. Преобразование осуществляется по формуле: Град = Рад × (180/π)
2. Флаги: Флаг нуля M1020, флаг заимствования M1021, флаг переноса M1022.
   Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022.
   Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021.
   Если результат равен 0, включается флаг нуля M1020.

Пример программы:
Когда X0 включен, происходит преобразование значения в радианах двоичного числа с плавающей запятой из (D1, D0) в градусы и результат сохраняется в том же формате в (D11, D10).





Радианы, двоичное значение с плавающей запятой

Градусы (радианы x 180/π), двоичное значение с плавающей запятой

API	Команда			Операнды	Функция
118	D	EBCD	P	(S) (D)	Перевод двоичного с плавающей запятой в десятичное с плавающей запятой

Контроллеры			
ES2/EX2	SS2	SA2/SE	SX2

Тип Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S													*		
D													*		

Шаги программы: DEBCD, DEBCDP: 9 steps

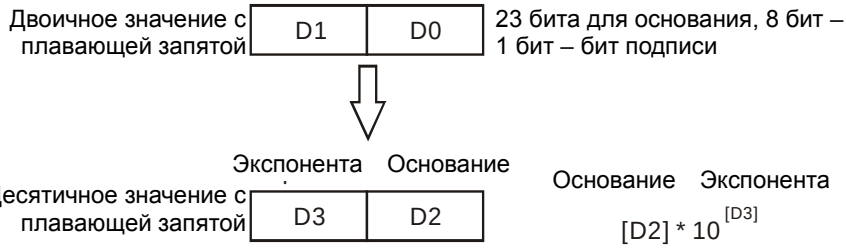
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат

Описание:
1. Команда преобразует двоичное число с плавающей запятой из **S** в десятичное число с плавающей запятой и сохраняет результат в **D**.
2. Флаги: Флаг нуля M1020, флаг заимствования M1021, флаг переноса M1022.
   Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022.
   Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021.
   Если результат равен 0, включается флаг нуля M1020.

Пример программы:
При включенном X0 двоичное значение с плавающей запятой из D1, D0 преобразуется в десятичное значение с плавающей запятой, и результат сохраняется в D3, D2.





API	Команда			Операнды	Функция
119	D	EBIN	P	(S) (D)	Перевод десятичного с плавающей запятой в двоичное с плавающей запятой

Контроллеры			
ES2/EX2	SS2	SA2/SE	SX2

Тип Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S													*		
D													*		

Шаги программы: DEBIN, DEBINP: 9 шагов

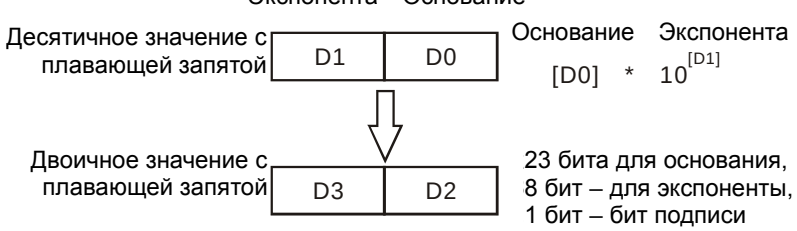
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат

Описание:
1. Команда преобразует десятичное число с плавающей запятой из **S** в двоичное число с плавающей запятой и сохраняет результат в **D**.
2. Например, **S** = 1234, **S** +1 = 3. Десятичное значение с плавающей запятой: 1.234×10^6
3. **D** должно быть в двоичном формате с плавающей запятой. **S** и **S** +1 представляют реальное число (мантиссу) и экспоненту соответственно.
4. Диапазон мантиссы: -9,999 ~ +9,999. Диапазон экспоненты: - 41 ~ +35. Если результат преобразования равен 0, включается флаг нуля M1020.

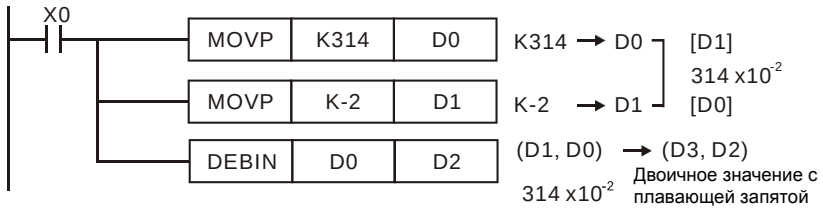


Пример программы 1:
При включенном X0, десятичное значение с плавающей запятой из D1, D0 преобразуется в двоичное значение с плавающей запятой, и результат сохраняется в D3, D2.



Пример программы 2:

1. Команда FLT (API 49) преобразует двоичное целое число в двоичное с плавающей запятой перед выполнением операции преобразования в десятичное с плавающей запятой. Значение результата преобразования может быть двоичным целым - применение команды DEBIN преобразует десятичное с плавающей запятой в двоичное целое.
2. При включенном X0, происходит перемещение K314 в D0 и K-2 в D1 для получения десятичного значения с плавающей запятой ($3.14 = 314 \times 10^{-2}$).



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
120	D	EADD	P	S_1 S_2 D	Сложение чисел с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEADD, DEADDP: 13 шагов
S_2					*	*							*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Слагаемое **S_2**: Слагаемое **D**: Результат сложения

Описание:

1. **S_1** + **S_2** = **D**. Числа с плавающей запятой в **S_1** и **S_2** складываются, и результат сохраняется в **D**.
2. Если значения операндов **S_1** или **S_2** являются константами K или H, то перед операцией сложения они автоматически преобразуются в двоичные числа с плавающей запятой.
3. **S_1** и **S_2** могут находиться в одном и том же регистре. В этом случае, если команда задана в режиме «непрерывного выполнения» (в основном, команда DEADDP) и включен контакт управляемого устройства, регистр будет прибавляться только один раз в каждом цикле.
4. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса):
   Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022.
   Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021.
   Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



При включенном X0, складываются двоичные значения с плавающей запятой из (D1, D0) и (D3, D2), результат сохраняется в (D11, D10).

Пример программы 2:



При включенном X2, складываются двоичное значение с плавающей запятой (D11, D10) с K1234 (автоматически преобразованное в двоичное значение с плавающей запятой) и результат сохраняется в (D21, D20).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
121	D	ESUB	P	S_1 S_2 D	Вычитание чисел с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DESUB, DESUBP: 13 шагов
S_2					*	*							*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходное значение **S_2**: Вычитаемое значение **D**: Результат вычитания

Описание:

1. **S_1** – **S_2** = **D**. Число с плавающей запятой в **S_2** вычитается из числа с плавающей запятой в **S_1** и результат сохраняется в **D**.
2. Если значения операндов **S_1** или **S_2** являются константами K или H, то перед операцией вычитания они автоматически преобразуются в двоичные числа с плавающей запятой.
3. **S_1** и **S_2** могут находиться в одном и том же регистре. В этом случае, если команда задана в режиме «непрерывного выполнения» (в основном, команда DESUB) и включен контакт управляемого устройства, регистр будет отниматься один раз в каждом цикле.
4. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса):
   Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022.
   Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021.
   Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



При включенном X0, двоичное значение с плавающей запятой из (D3, D2) вычитается из двоичного значения с плавающей запятой (D1, D0) и результат сохраняется в (D11, D10).

Пример программы 2:



При включенном X2, K1234 (автоматически преобразованное в двоичное с плавающей запятой) вычитается из числа в (D1, D0) и результат созраняется в (D11, D10).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
122	D	EMUL	P	S_1 S_2 D	Умножение чисел с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEMUL, DEMULP: 13 шагов
S_2					*	*							*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Множимое значение **S_2**: Множитель **D**: Результат умножения

Описание:

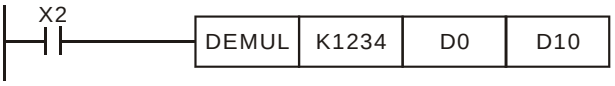
1. $\mathbf{S_1} \times \mathbf{S_2} = \mathbf{D}$. Число с плавающей запятой из $\mathbf{S_1}$ перемножается с числом с плавающей запятой из $\mathbf{S_2}$ и результат сохраняется в **D**. Формат множителей – двоичный с плавающей запятой.
2. Если значения операндов $\mathbf{S_1}$ или $\mathbf{S_2}$ являются константами K или H, то перед операцией умножения они автоматически преобразуются в двоичные числа с плавающей запятой.
3. $\mathbf{S_1}$ и $\mathbf{S_2}$ могут находиться в одном и том же регистре. В этом случае, если команда задана в режиме «непрерывного выполнения» (в основном, команда DEMUL) и включен контакт управляемого устройства, регистр будет умножаться один раз в каждом цикле.
4. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса): Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022. Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



При включенном X1, двоичное число с плавающей запятой (D1, D0) перемножается с числом (D11, D10) и результат сохраняется в (D21, D20).

Пример программы 2:



При включенном X2, K1234 (автоматически преобразованное в двоичное с плавающей запятой) перемножается с числом (D1, D0) и результат сохраняется в (D11, D10).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
123	D	EDIV	P	S_1 S_2 D	Деление чисел с плавающей запятой				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DEADD, DEADDP: 13 шагов
S_2					*	*							*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

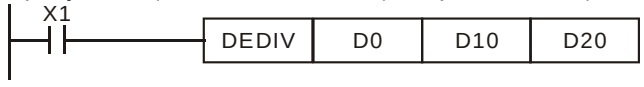
$\mathbf{S_1}$: Делимое $\mathbf{S_2}$: Делитель **D**: Результат (частное и остаток)

Описание:

1. $\mathbf{S_1} \div \mathbf{S_2} = \mathbf{D}$. Число с плавающей запятой $\mathbf{S_1}$ делится на число с плавающей запятой в $\mathbf{S_2}$ и результат сохраняется в **D**. Числа – в двоичном формате.
2. Если значения операндов $\mathbf{S_1}$ или $\mathbf{S_2}$ являются константами K или H, то перед операцией деления они автоматически преобразуются в двоичные числа с плавающей запятой.
3. Если $\mathbf{S_2}$ = 0, возникает ошибка и команда не выполняется.
4. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса): Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022. Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

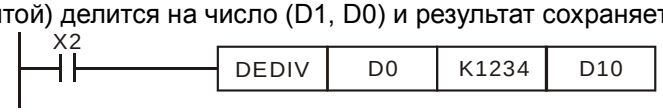
Пример программы 1:

При включенном X1, двоичное число с плавающей запятой (D1, D0) делится на число (D11, D10) и результат (частное и остаток) сохраняется в (D21, D20).



Пример программы 2:

При включенном X2, K1234 (автоматически преобразованное в двоичное с плавающей запятой) делится на число (D1, D0) и результат сохраняется в (D11, D10).



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
124	D	EXP	P	(S) (D)	Вычисление экспоненты в формате с плавающей запятой				

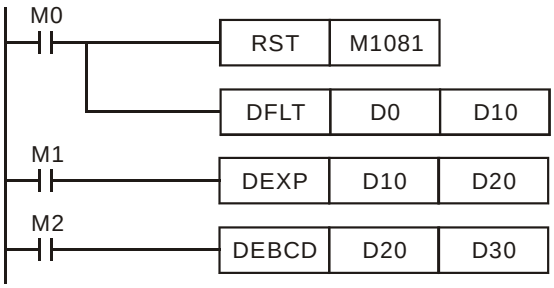
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DEXP, DEXPP: 9 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Экспонента **D**: Результат

Описание:
1. Находится экспонента числа, заданного в S, т.е. **D** = e ^{S}, основание e =2.71828, **S** является значением экспоненты.
2. EXP [**S +1, S**] = [**D** +1, **D**]
3. Операнд **S** может иметь положительное или отрицательное значение. Регистр **D** - в 32-битном формате. Значение **S** перед выполнением команды должно быть преобразовано в двоичное с плавающей запятой.
4. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса): Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022. Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы:



1. Когда M0 включен, число в (D1, D0) преобразуется в двоичное с плавающей запятой и сохраняется в (D11, D10).
2. Когда M1 включен, находится экспонента числа в (D11, D10). Результат записывается в (D21, D20) в двоичном формате с плавающей запятой.
3. Когда M2 включен, двоичное число в (D21, D20) преобразуется в десятичное с плавающей запятой и сохраняется в (D31, D30). (D31 указывает степень 10 для D30).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
125	D	LN	P	(S) (D)	Вычисление натурального логарифма числа с плавающей запятой				

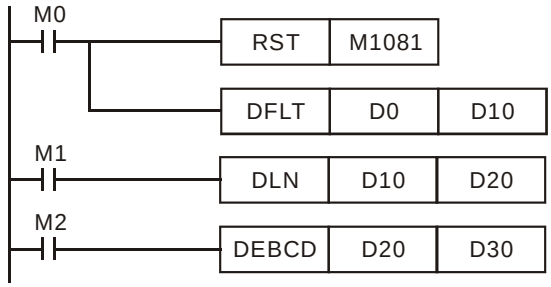
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DLN, DLNP: 9 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **D**: Результат

Описание:
1. Вычисление натурального логарифма (LN) операнда **S**: e^{D} = **S,** LN[**S** +1, **S**]=[**D** +1, **D**]
2. Операнд **S** может иметь только положительное значение. Регистр **D** - в 32-битном формате. Значение **S** перед выполнением команды должно быть преобразовано в двоичное с плавающей запятой.
3. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса): Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022. Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы:



1. Когда M0 включен, число в (D1, D0) преобразуется в двоичное с плавающей запятой и сохраняется в (D11, D10).
2. Когда M1 включен, находится натуральный логарифм числа в (D11, D10). Результат записывается в (D21, D20) в двоичном формате с плавающей запятой.
3. Когда M2 включен, двоичное число в (D21, D20) преобразуется в десятичное с плавающей запятой и сохраняется в (D31, D30). (D31 указывает степень 10 для D30).

API	Команда			Операнды	Функция
126	D	LOG	P	S_1 S_2 D	Вычисление логарифма в формате с плавающей запятой

Контроллеры			
ES2/EX2	SS2	SA2/ SE	SX2

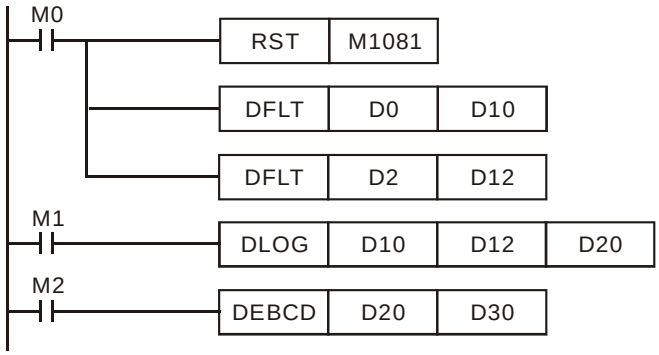
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DLOG, DLOGP: 13 шагов
S_2					*	*							*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:
S_1: Основание S_2: Число **D**: Результат

Описание:
1. Вычисление логарифма с основанием S_1 числа S_2 и сохранение результата в **D**.
2. Операнд **S** может иметь только положительное значение. Регистр **D** - в 32-битном формате. Значение **S** перед выполнением команды должно быть преобразовано в двоичное с плавающей запятой.
3. Вычисление логарифма: $S_1^D = S_2$, **D** = ? → $Log_{S1}^{S_2} = D$
   Пример: примем $S_1 = 5$, $S_2 = 125$, $S_1^D = S_2$, **D** = ? → $5^D = 125$ → **D** = $Log_{S1}^{S_2} = log_5^{125} = 3$.
4. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса): Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022. Если абсолютное значение результата меньше минимального значения числа с плавающей запятой, включается флаг заимствования M1021.Если результат равен 0, включается флаг нуля M1020.

Пример программы:
1. Когда M0 включен, значения (D1, D0) и (D3, D2) преобразуются в двоичные с плавающей запятой и сохраняются соответственно в (D11, D10) и (D13, D12).
2. Когда M1 включен, вычисляется логарифм с основанием (D11, D10) числа (D13, D12). Результат сохраняется в регистрах (D21, D20) в двоичном формате с плавающей запятой.
3. Когда M2 включен, двоичное число в (D21, D20) преобразуется в десятичное с плавающей запятой и сохраняется в (D31, D30). (D31 указывает степень 10 для D30).



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
127	D	ESQR	P	(S) (D)	Квадратный корень в формате с плавающей запятой				

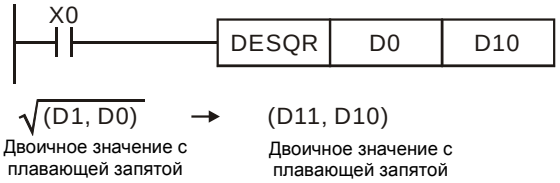
Тип Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DESQR, DESQRP: 9 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат операции

Описание:
1. Эта команда вычисляет квадратный корень из числа с плавающей запятой в **S** и сохраняет результат в **D**. Исходные данные и результат - в двоичном формате с плавающей запятой.
2. Если операнд **S** является константой K или H, то перед выполнением команды он автоматически преобразуется в двоичное число с плавающей запятой.
3. Если результат **D** равен 0, включается флаг нуля M1020.
4. **S** может иметь только положительное значение. Попытка выполнения операции с отрицательным значением приводит к ошибке, и команда выполняться не будет. Включаются флаги M1067 и M1068 и код ошибки "0E1B" будет записан в D1067.
5. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса).

Пример программы 1:



Когда X0 включен, вычисляется квадратный корень двоичного значения с плавающей запятой из (D1, D0) и сохраняется в (D11, D10).

Пример программы 2:



Когда X2 включен, квадратный корень из K1234 (автоматически преобразованного в двоичное с плавающей запятой) сохраняется в (D11, D10).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
128	D	POW	P	(S_1) (S_2) (D)	Возведение в степень числа с плавающей запятой				

Тип Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPOW, DPOWP: 13 шагов
S_2					*	*							*			
D													*			

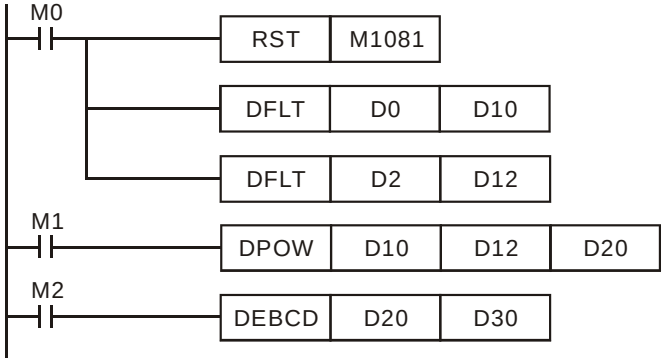
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Основание S_2: Экспонента **D**: Результат операции

Описание:
1. Возведение в степень S_2 числа из S_1 и сохранение результата в **D**. S_1^{S2} = D
   POW [S_1+1, S_1]^[S_2+1, S_2] = **D**
   Операнд **S** может иметь только положительное значение. Регистр **D** - в 32-битном формате. Значения S_1 и S_2 перед выполнением команды должны быть преобразованы в двоичные с плавающей запятой. Например: Приняв S_1 = 5, S_2 = 3, тогда **D** = 5^3 =125
2. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса):
   Если абсолютное значение результата превышает максимальное значение числа с плавающей запятой, включается флаг переноса M1022.
   Если абсолютное значение результата меньше минимального значения числа с

плавающей запятой, включается флаг заимствования M1021.
Если результат равен 0, включается флаг нуля M1020.

Пример программы:



1. Когда M0 = включен, значения из (D1, D0) и(D3, D2) преобразовываются в двоичные с плавающей запятой, и результат сохраняется в регистрах (D11, D10) и (D13, D12) соответственно.
2. Когда M1 = включен, число (D11, D10) возводится в степень (D13, D12). Результат сохраняется в (D21, D20) в двоичном формате с плавающей запятой.
3. Когда M2 = ON, преобразуется значение из (D21, D20) в десятичное с плавающей запятой и результат сохраняется в (D31, D30). (D31 показывает степень 10 для D30).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
129	D	INT	P	(S) (D)	Преобразование числа с плавающей запятой в целое				

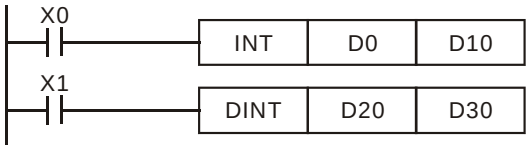
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			INT, INTP: 5 шагов DINT, DINTP: 9 шагов
D											*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат операции

Описание:
1. Двоичное с плавающей запятой значение из **S** преобразуется в двоичное целое и результат сохраняется в **D**. Десятичные знаки в результате операции отбрасываются.
2. Эта команда противоположна команде API 49 (FLT).
3. Флаги - M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса):
   Если результат преобразования равен 0, включается флаг нуля M1020.
   Если есть отброшенные десятичные знаки, включается флаг заимствования M1021.
   Если результат превышает допустимый диапазон, включается флаг переноса M1022.
   16-битная команда: -32,768 ~ 32,767
   32-битная команда: -2,147,483,648 ~ 2,147,483,647

Пример программы:



1. Когда X0 включен, двоичные значения с плавающей запятой из (D1, D0) преобразуются в двоичные целые и сохраняются в D10. Десятичные отбрасываются.
2. Когда X1 включен, двоичные значения с плавающей запятой из (D21, D20) преобразуются в двоичные целые и сохраняются в (D31, D30). Десятичные результата отбрасываются.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
130	D	SIN	P	(S) (D)	Вычисление синуса				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DSIN, DSINP: 9 шагов
D													*			

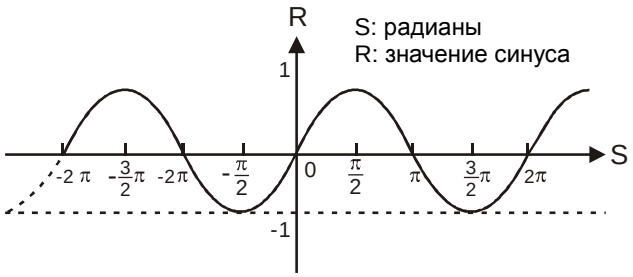
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S: Источник данных (0° < **S** < 360°) **D**: Результат операции

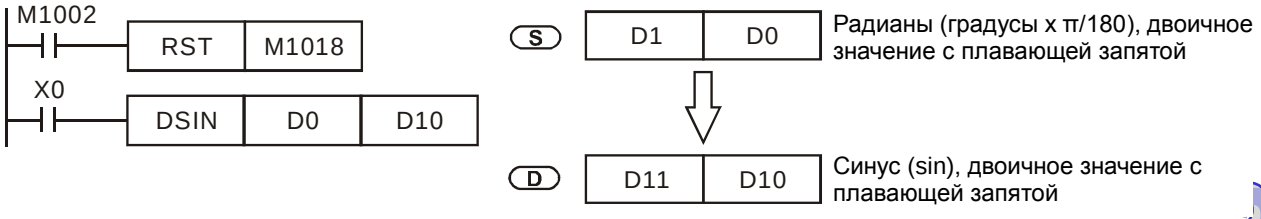
Описание:

1. Команда SIN вычисляет синус **S** и сохраняет результат в **D**.
2. Значение **S** может быть установлено в радианах или градусах флагом M1018.
   M1018 выключен – радианы.
   RAD = град ×π/180.
   M1018 включен – градусы.
   Диапазон: 0° ＜ град ＜ 360°.
3. Если результат в **D** равен 0, включается флаг нуля M1020.

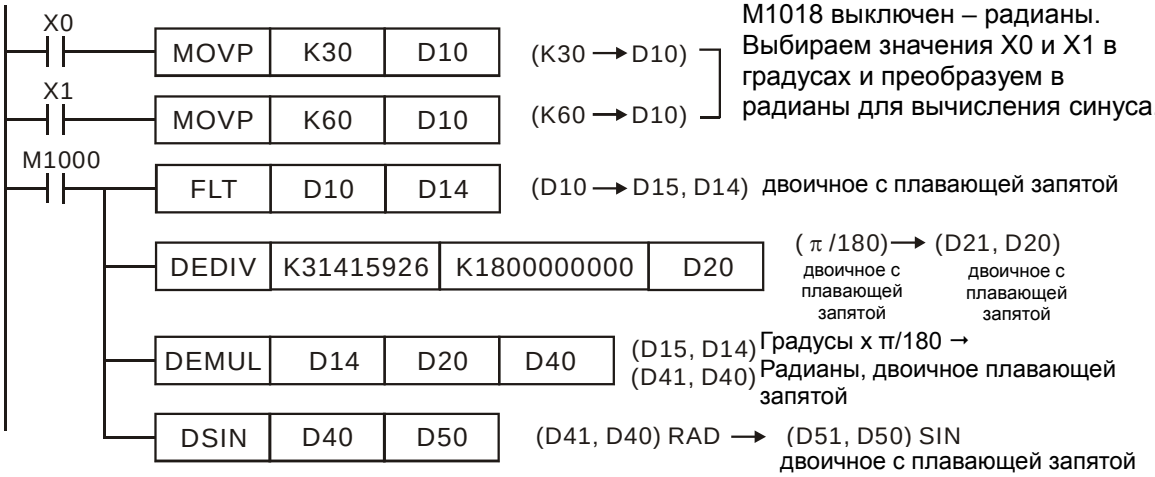


Пример программы 1:

M1018 выключен – радианы. Когда X0 включен, команда DSIN вычисляет синус двоичного числа с плавающей запятой из (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.

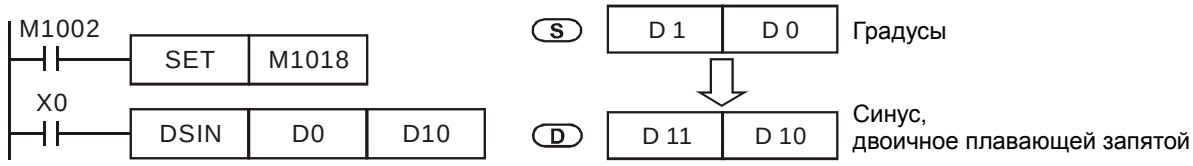


Пример программы 2:



Пример программы 3:

M1018 включен – градусы. Когда X0 включен, команда DSIN вычисляет синус (0° ＜ град ＜ 360°) в (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.



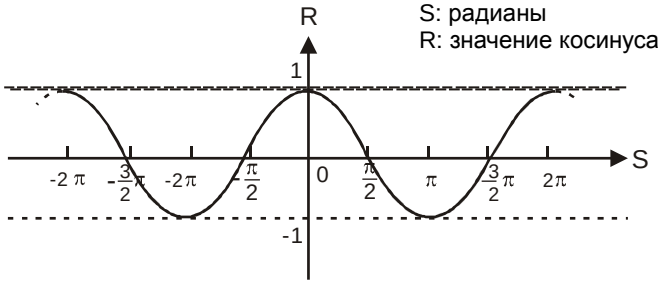
API	Команда			Операнды	Функция	Контроллеры			
131	D	COS	P	(S) (D)	Вычисление косинуса	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DCOS, DCOSP: 9 шагов
D													*			

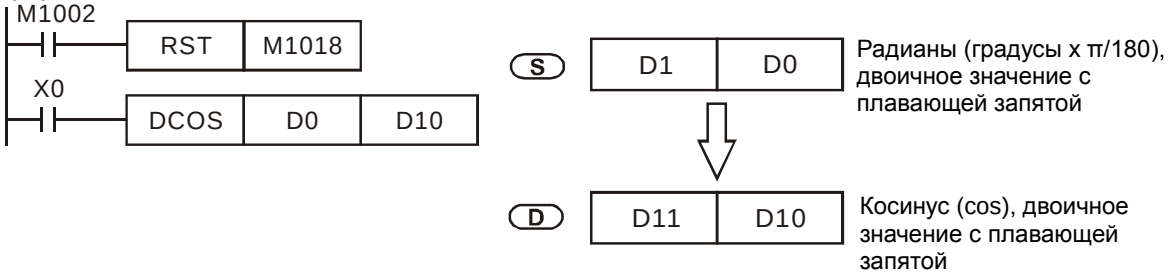
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных (0° < **S** < 360°) **D**: Результат операции

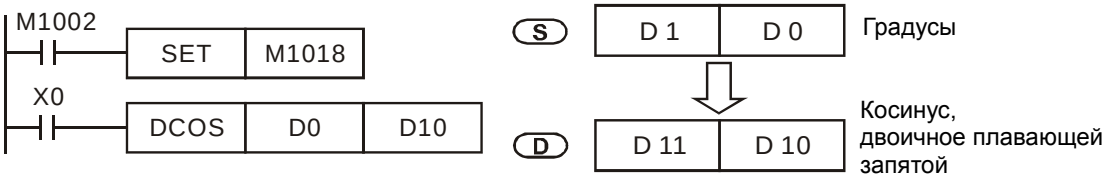
Описание:
1. Команда COS вычисляет косинус числа из **S** и сохраняет результат в **D**.
2. Значение в **S** выбирается флагом M1018 в радианах или градусах.
3. M1018 выключен – радианы. Рад. = град. ×π/180.
4. M1018 включен – градусы. Диапазон: 0° < град. < 360°.
5. Флаг: M1018 (Флаг переключения град/рад).
6. Если результат в **D** равен 0, включается флаг нуля M1020.



Пример программы 1:
M1018 выключен - радианы. Когда X0 включен, команда DCOS вычисляет косинус двоичного числа с плавающей запятой из (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.



Пример программы 2:
M1018 вкл. – градусы. Когда X0 включен, команда DCOS вычисляет косинус (0° < град < 360°) в (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.



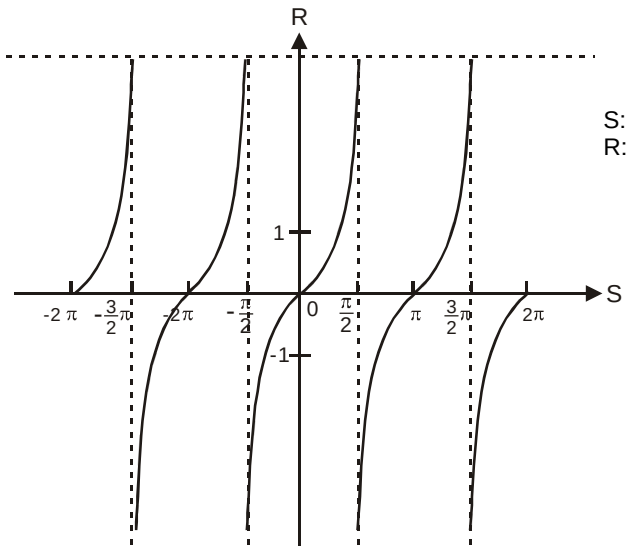
API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
132	D	TAN	P	(S) (D)	Вычисление тангенса				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DTAN, DTANP: 9 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

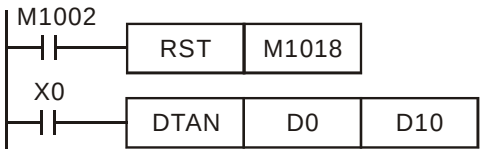
Операнды:
S: Источник данных (0° < **S** < 360°) **D**: Результат операции

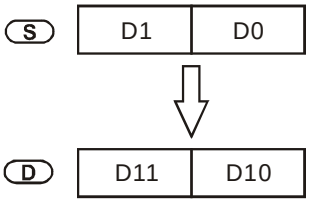
Описание:
1. Команда TAN вычисляет тангенс значения в **S** и сохраняет результат в **D**.
2. Значение в **S** устанавливается в радианах или градусах флагом M1018.
   M1018 выключен – радианы. RAD = град ×π/180.
   M1018 включен – градусы. Диапазон: 0° < град < 360°.
3. Если результат в **D** равен 0, включается флаг нуля M1020.



S: радианы
R: значение тангенса

Пример программы 1:
M1018 выключен - радианы. Когда X0 включен, команда DTAN вычисляет тангенс двоичного числа с плавающей запятой из (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.

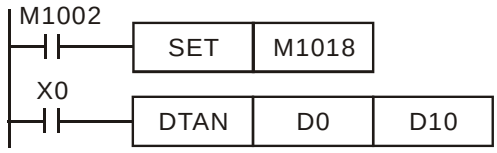


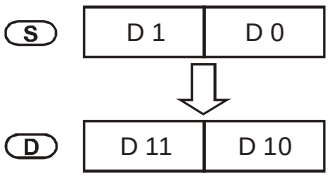


Радианы (градусы x π/180), двоичное значение с плавающей запятой

Тангенс (tan), двоичное значение с плавающей запятой

Пример программы 2:
M1018 включен – градусы. Когда X0 включен, команда DTAN вычисляет тангенс (0°＜град＜360°) в (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.





Градусы

Тангенс (tan), двоичное значение с плавающей запятой

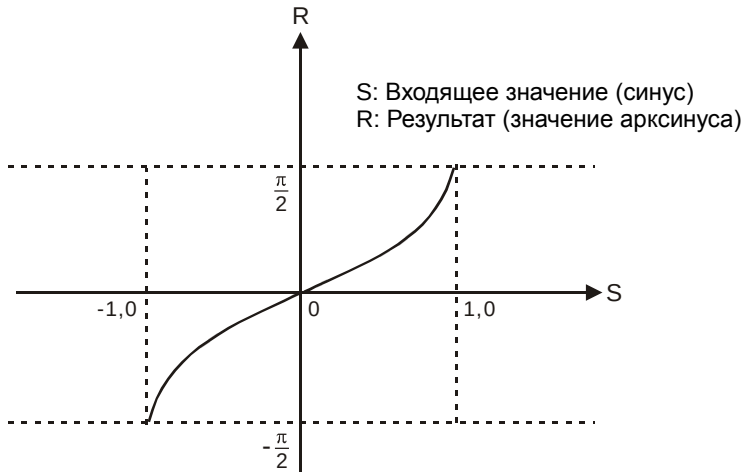
API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
133	D	ASIN	P	(S) (D)	Вычисление арксинуса				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DASIN, DASINP: 9 шагов
D													*			

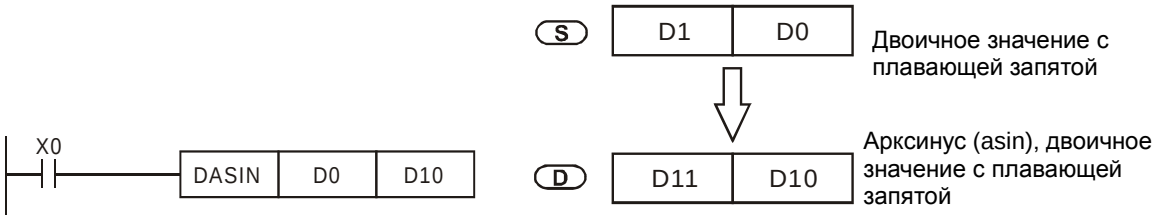
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных (двоичное с плавающей запятой) **D**: Результат операции

Описание:
1. Команда ASIN вычисляет арксинус значения в **S** и сохраняет результат в **D.**
2. Значение ASIN = SIN^{-1}
3. Если результат в **D** равен 0, включается флаг нуля M1020.
4. Десятичное значение SIN, заданное в **S** должно быть -1.0 ~ +1.0. Если значение выходит за пределы этого диапазона, включаются флаги M1067 и M1068 и команда прекращает выполнение.



Пример программы:
Когда X0 включен, команда DASIN вычисляет арксинус в двоичном формате с плавающей запятой из (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.



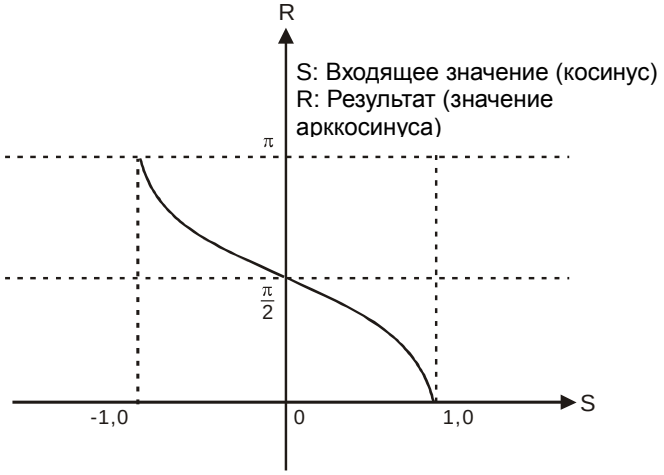
API	Команда			Операнды	Функция	Контроллеры			
134	D	ACOS	P	S D	Вычисление арккосинуса	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DACOS, DACOSP: 9 шагов
D													*			

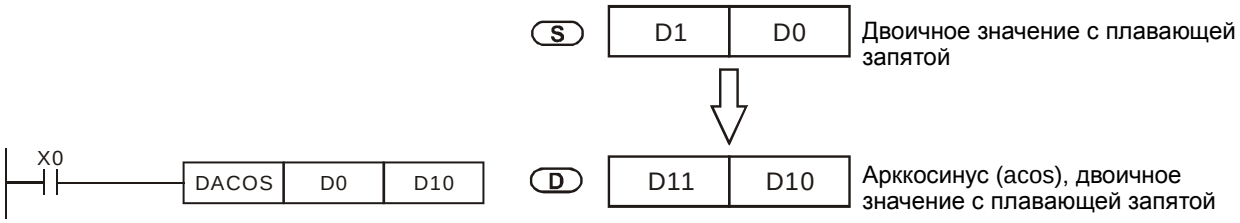
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных (двоичное с плавающей запятой) **D**: Результат операции

Описание:
1. Команда ACOS вычисляет арккосинус значения в **S** и сохраняет результат в **D.**
2. Значение ACOS = COS^{-1}
3. Если результат в **D** равен 0, включается флаг нуля M1020.
4. Десятичное значение COS, заданное в **S** должно быть -1.0 ~ +1.0. Если значение выходит за пределы этого диапазона, включаются флаги M1067 и M1068 и команда прекращает выполнение.



Пример программы:
Когда X0 включен, команда DACOS вычисляет арккосинус в двоичном формате с плавающей запятой из (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.



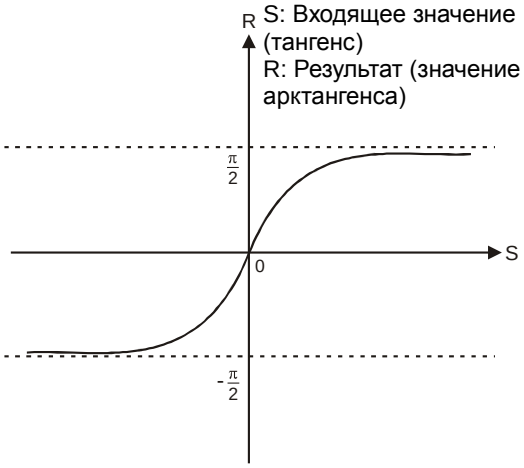
API	Команда			Операнды	Функция	Контроллеры			
135	D	ATAN	P	(S) (D)	Вычисление арктангенса	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DATAN, DATANP: 9 шагов
D													*			

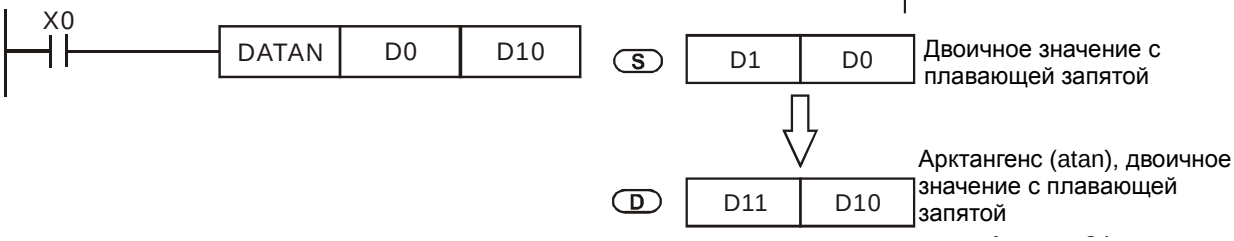
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных (двоичное с плавающей запятой) **D**: Результат операции

Описание:
1. Команда ATAN вычисляет арктангенс значения в **S** и сохраняет результат в **D.**
2. Значение ATAN = TAN^{-1}
3. Если результат в **D** равен 0, включается флаг нуля M1020.



Пример программы:
Когда X0 включен, команда DATAN вычисляет арктангенс в двоичном формате с плавающей запятой из (D1, D0) и сохраняет результат в (D11, D10) в двоичном формате с плавающей запятой.



API	Команда			Операнды	Функция
136	D	SINH	P	(S) (D)	Вычисление гиперболического синуса

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			5 шагов для 16-битной команды
D													*			

Операнды:
S: Источник данных (двоичное с плавающей запятой) **D**: Результат операции

Описание: Команда SINH вычисляет гиперболический синус =$(e^{S}-e^{-S})/2$ и сохраняет результат в **D.**

Пример:
При включении X0 в качестве исходного значения принимается двоичное число с плавающей запятой (D1, D0). Рассчитывается значение гиперболического синуса и результат сохраняется в (D11, D10) в виде двоичного числа с плавающей запятой.



API	Команда			Операнды	Функция	
137	D	COSH	P	S D	Вычисление гиперболического косинуса	

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			5 шагов для 16-битной команды
D													*			

Операнды:
S: Источник данных (двоичное с плавающей запятой) **D**: Результат операции

Описание: Команда COSH вычисляет гиперболический синус $=(e^{S}+e^{-S})/2$ и сохраняет результат в **D.**

Пример:



При включении X0 в качестве исходного значения принимается двоичное число с плавающей запятой (D1, D0). Рассчитывается значение гиперболического косинуса и результат сохраняется в (D11, D10) в виде двоичного числа с плавающей запятой.

API	Команда			Операнды	Функция	
138	D	TANH	P	S D	Вычисление гиперболического тангенса	

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			5 шагов для 16-битной команды
D													*			

Операнды:
S: Источник данных (двоичное с плавающей запятой) **D**: Результат операции

Описание: Команда TANH вычисляет гиперболический тангенс $=(e^{S}-e^{-S})/(e^{S}+e^{-S})$ и сохраняет результат в **D.**

Пример:

При включении X0 в качестве исходного значения принимается двоичное число с плавающей запятой (D1, D0). Рассчитывается значение гиперболического тангенса и результат сохраняется в (D11, D10) в виде двоичного числа с плавающей запятой.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
143		DELAY	P	S	Задержка выполнения				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DELAY, DELAYP: 3 шага

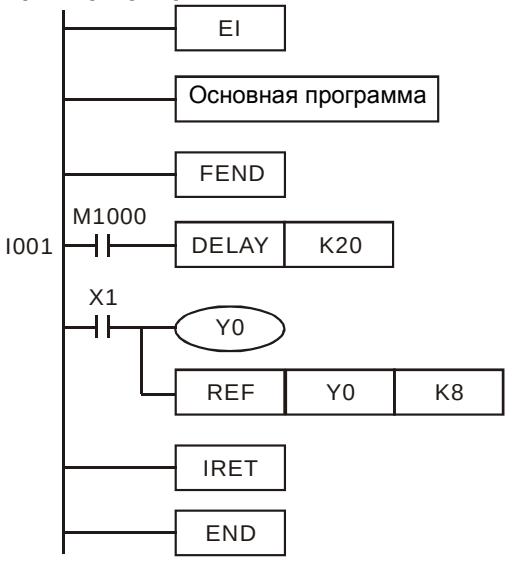
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Время задержки, шаг: 0.1 мс (K1~K1000)

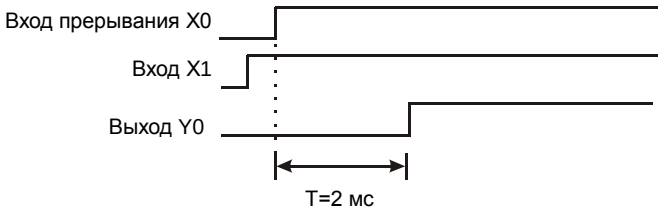
Описание: (Команда для DVP-ES2/EX2 версии 3.00 и выше, DVP-SS2 версии 2.80 выше, DVP-SA2 версии 2.40, DVP-SX2 версии 2.20, DVP-SE версии 1.20).

1. Шаг времени задержки устанавливается с помощью флага M1148. При выключенном M1148 – 100 мкс, при включенном – 5 мкс. После выполнения флаг M1148 будет сброшен.
2. При выполнении команды DELAY каждый цикл выполнения программы будет отложен на время задержки, устанавленного пользователем.

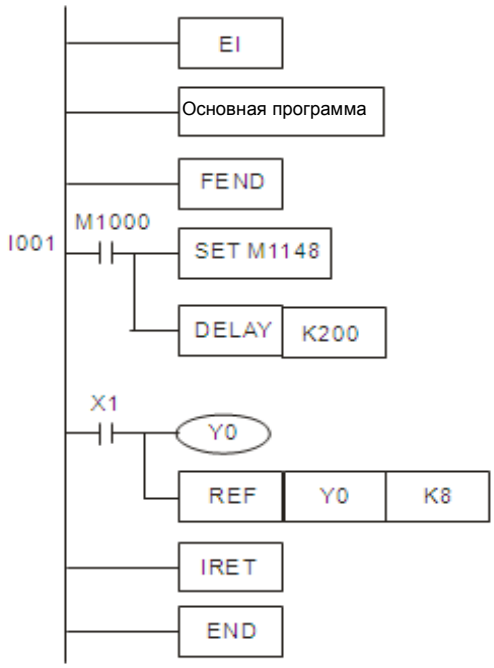
Пример программы:

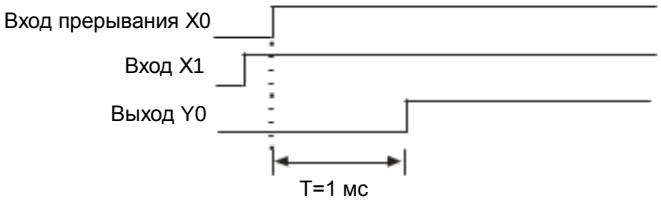


Когда вход прерывания X0 включается, подпрограмма прерывания выполняется первой на время задержки, а выполнение основной программы (X1 включен, Y0 включен) будет отложено на 2 мс.



Пример программы: (Команда для DVP-ES2/EX2 версии 3.20, DVP-SS2 версии 3.00, DVP-SA2 версии 2.60, DVP-SX2 версии 2.40, DVP-SE версии 1.40).





Когда вход прерывания X0 включается, подпрограмма прерывания выполняется первой на время задержки, а выполнение основной программы (X1 включен, Y0 включен) будет отложено на 1 мс.

Заметки:
1. Время задержки настраивается пользователем. Оно может быть увеличено при работе с высокоскоростным счетчиком и высокоскоростными командами импульсного выхода.
2. Время задержки команды DELAY может быть больше из-за задержки транзисторов или реле, определенных в качестве внешних выходов.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
144	GPWM	(S1) (S2) (D)	Общая команда генерации импульсов ШИМ				

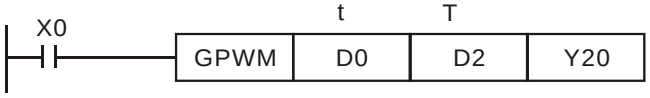
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			GPWM: 7 шагов
S_2													*			
D		*	*	*												

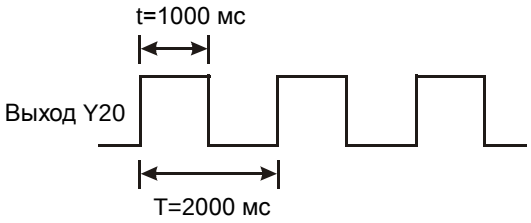
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Ширина выходного импульса **S_2**: Период выходного импульса (занимает 3 регистра)
D: Импульсный выход

Описание:
1. Когда команда GPWM выполняется, на импульсном выходе **D** генерируются импульсы шириной **S_1** с периодом **S_2**.
2. Диапазон **S_1**: t = 0~32,767 мс.; Диапазон **S_2**: T = 1~32,767 мс, **$S_1 \le S_2$**.
3. **S_2** +1 и **S_2** +2 системные параметры, не используйте их в программе.
4. **D:** импульсный выход: Y, M и S.
5. Когда **$S_1 \le 0$**, импульсы не генерируются. При **$S_1 \ge S_2$**, выход будет включен постоянно.
6. **S_1** и **S_2** могут быть изменены во время выполнения команды GPWM.

Пример программы:
Принимаем D0 = K1000, D2 = K2000. Когда X0 включен, Y20 выдает импульсы (см. диаграмму). При выключенном X0 выход Y20 выключен.





Заметки:
Команда работает один цикл программы, поэтому максимальная ошибка не выйдет за рамки одного цикла работы ПЛК. **S_1**, **S_2** и (**S_2** - **S_1**) должны быть больше цикла работы ПЛК, иначе произойдет сбой работы команды GPWM. Применение этой команды в подпрограмме приведет к некорректным выходным значениям.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
145	FTC	(S1) (S2) (S3) (D)	Регулировка температуры с нечеткой логикой				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			FTC: 7 шагов
S_2					*	*							*			
S_3													*			
D													*			

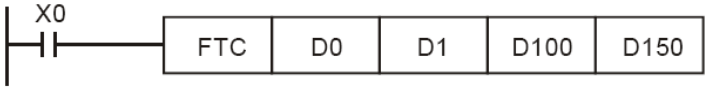
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Заданное значение (SV) **S_2**: Текущее значение (PV)
S_3: Параметр (время дискретизации) **D**: Выходное значение (MV)

Описание:

1. Диапазон **S1**: 1 ~ 5000 (т.е. 0.1°C ~ 500°C). Ед. изм.: 0.1°. Если (**S3** +1) является K0, диапазон: 0.1°C ~ 500°C.
2. Настройки параметра **S3** +1: бит0 = 0 ->°C; бит1 = 0 ->°F; бит1 = 0 -> функция фильтрации отключена; бит1 = 1 -> функция фильтрации включена; бит2 ~ бит5 -> 4 способа нагрева; бит6 ~ бит15 -> зарезервированы. См. примечание.
3. **D** находится в диапазоне 0 ~ время дискретизации × 100. При использовании данной команды пользователь должен настроить другие команды в зависимости от типа нагревателя. Например, команда FTC применяется вместе с командой GPWM для управления выходными импульсами. “Время дискретизации × 100” – цикл выходного импульса GPWM; MV является шириной импульса GPWM. См. пример программы 1.
4. Нет ограничений в количестве применяющихся команд FTC, но многократное использование одного и того операнда может привести к ошибке.
5. Команда поддерживается ПЛК серий DVP-ES2/EX2 v. 3.22, DVP-SS2 v. 3.20, DVP-SA2/SX2 v. 2.66, DVP-SE v.1.60 (и выше).

Пример программы:



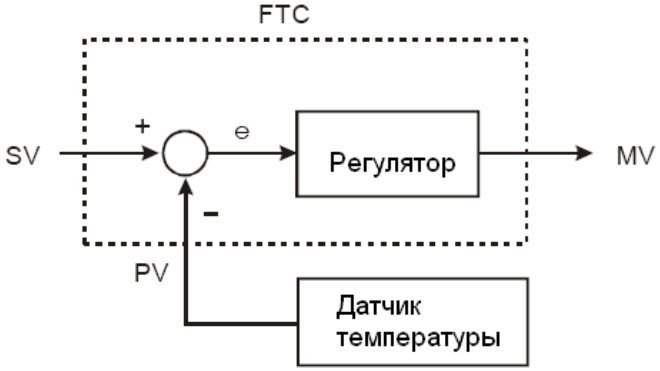
При включении X0 команда выполняется и результат сохраняется в D150. При выключенном X0 команда не выполняется и данные остаются без изменений:

Примечание:

1. Настройки **S3**:

Операнд	Функция	Диапазон	Описание
S_3	Время дискретизации (Ts) (ед.изм.: 100 мс)	1 ~ 200 (ед.изм.:100 мс)	Временной интервал между расчетами ПИД-регулирования и обновлением параметров MV. Если TS = 0, команда PID выполняться не будет. Если TS меньше времени 1 цикла программы, команда PID будет выполняться в каждом цикле программы, т.е минимальное значение TS должно быть больше времени цикла программы.
S_3 + 1	b0: ед. изм. температуры b1: функция фильтрации b2 ~ b5: способы нагрева b6 ~ b15: зарезервированы	b0 =0: °C b0 =1: °F	Если значение превышает верхний предел, то применяется верхнее предельное значение
		b1=0: без функции фильтрации b1=1: с функцией фильтрации	Без функции фильтрации PV = Текущее значение. При включенной функции фильтрации, PV = (текущее значение + предыдущее PV)/2
		b2=1	Высокоинерционный (медленный) нагревательный элемент
		B3=1	Стандартный нагревательный элемент
		B4=1	Быстрый нагревательный элемент
		B5=1	Очень быстрый нагревательный элемент
S_3 + 2… S_3 + 6	Системные параметры, не используются		

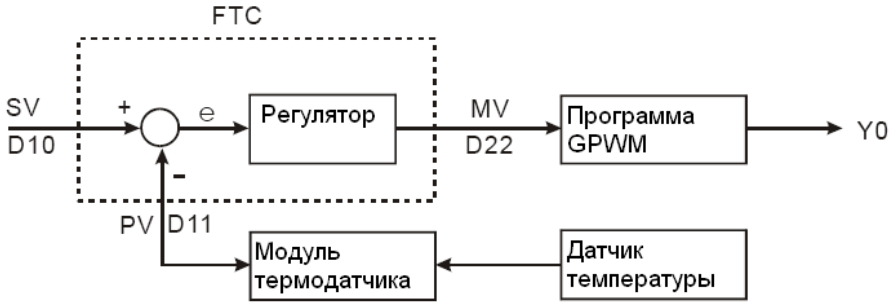
2. Схема управления:

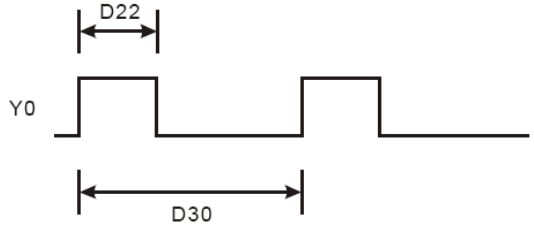


3. Рекомендации:

- Рекомендуется задать время дискретизации в 2 раза больше времени опроса датчика температуры для улучшения процесса регулирования.
- Бит2 ~ бит5 **S**з+1 служат для управления скоростью регулирования. Если эти параметры не установлены, команда FTC автоматически работает в режиме “Стандартный нагревательный элемент”. Если процесс регулирования – достижение значения SV - слишком медленный, выберите режим “ Высокоинерционный (медленный) нагревательный элемент” для повышения скорости регулирования. Если скорость регулирования слишком велика или происходят значительные колебания температуры, необходимо выбрать режим “Быстрый нагревательный элемент” для уменьшения скорости.
- Если несколько бит2 ~ бит5 **S**з+1 = 1, то команда FTC выберет руководствоваться первым из них. Параметр можно менять в процессе регулирования.

Пример программы 1:
Схема регулирования:

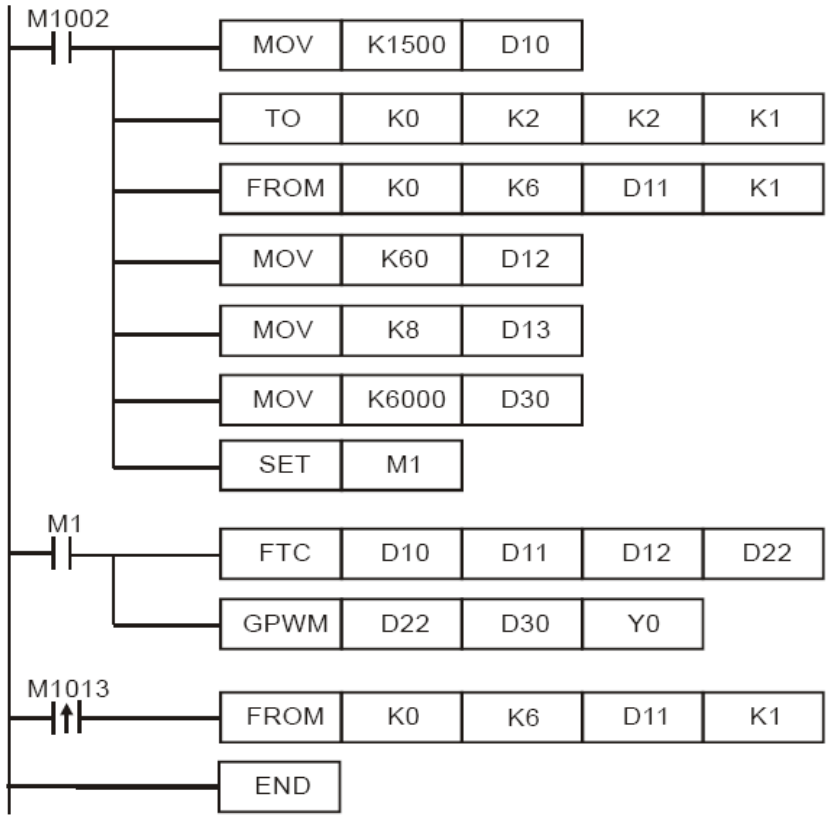




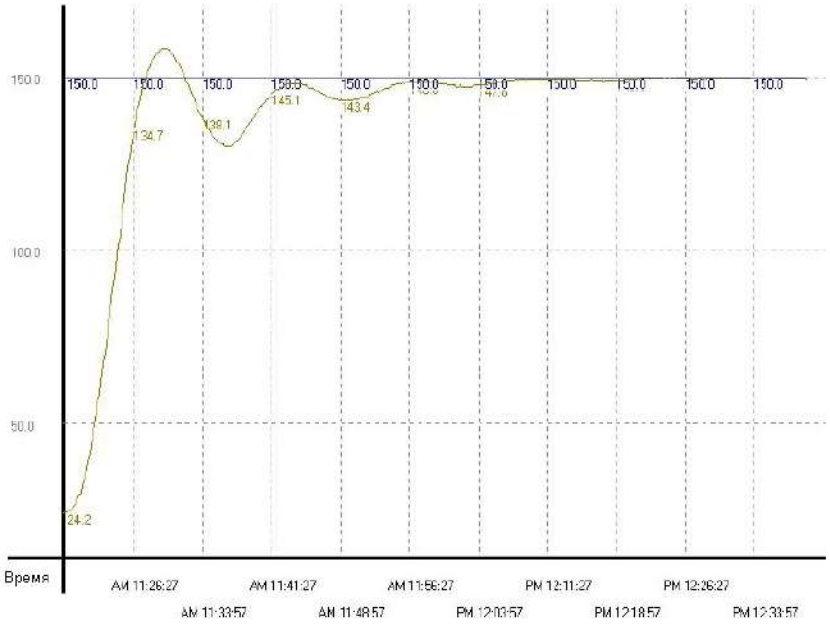
Выход D22 (MV) для команды является входом D22 команды GPWM, регулирующей длину импульсов. D30 является фиксированным временем цикла импульсов. Временные диаграммы для Y0:

Для примера примем: D10 = K1,500 (заданная температура), D12 = K60 (время дискретизации: 6 сек.), D13 = K8 (бит3=1), D30 = K6,000 (=D12*100)

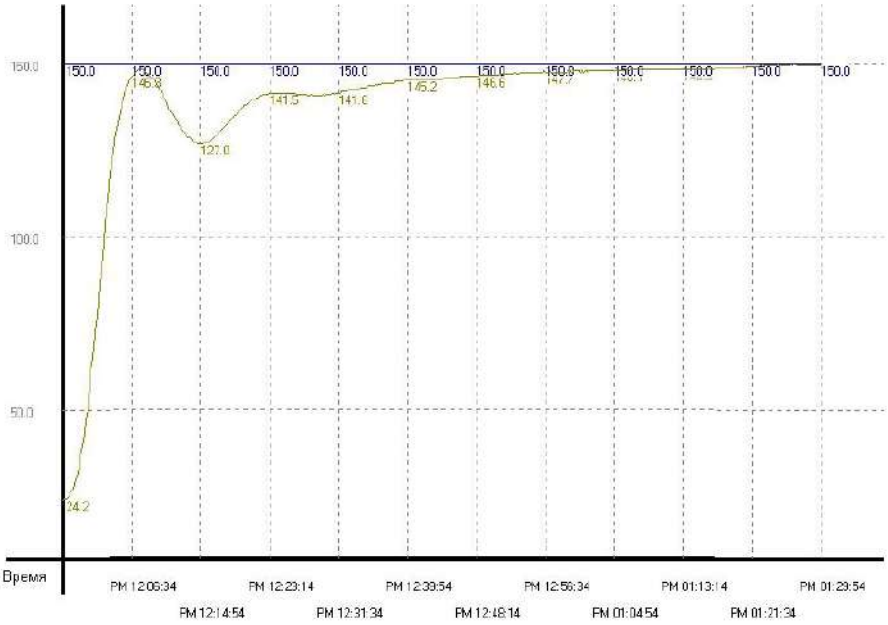
Пример программы регулирования:



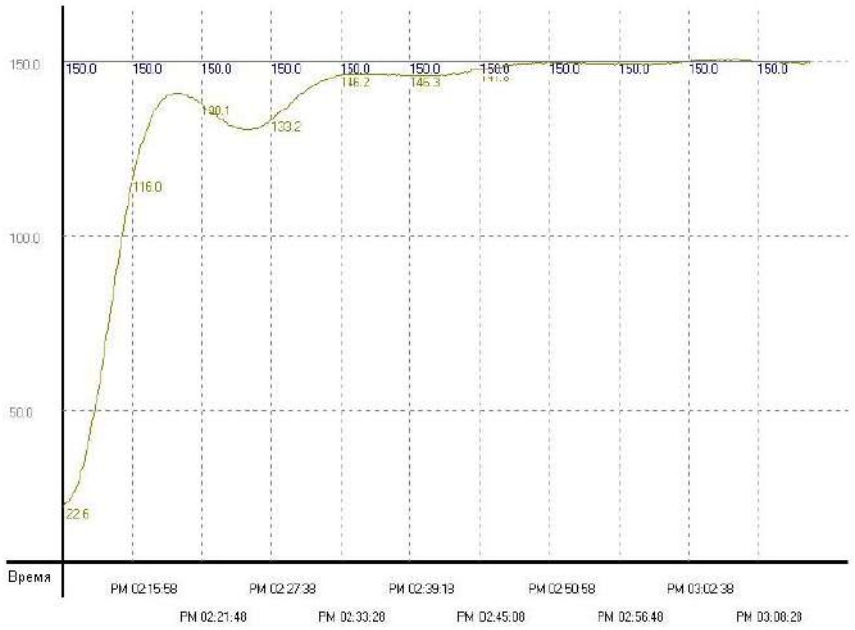
Эксперимент 1: Следим за нагревом печи, которая рассчитана на нагрев до 250°C. На графике показаны заданная (150°C) и текущая температура. По графику видно, что температура достигает заданного значения через 48 мин. С погрешностью ±1°C с максимальным превышением заданной температуры примерно на 10°C.



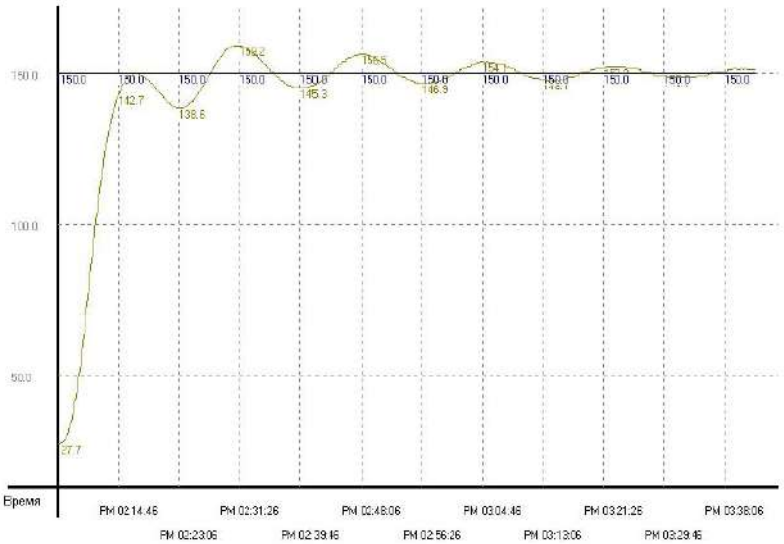
Пример программы 2:
Т.к. на предыдущем графике видно, что присутсвует перерегулирование, изменим режим на “Быстрый нагревательный элемент” (D13 = K16). Результат показан на графике ниже.
По графику видно, что, хотя температура в процессе нагрева не превышала на этот раз заданную, но достижение заданной температуры с погрешностью ±1°C заняло 1 ч 15 мин, т.е., выбран правильный режим, но время дискретизации слишком велико, что привело к увеличенному времени нагрева.



Пример программы 3:
Для ускорения достижения заданной температуры изменим время дискретизации на 4 сек. и период импульсов ШИМ (D12 = K40, D30 = K4,000). Как видно из графика, удалось снизить время достижения заданной температуры до 37 минут.
Следовательно, изменение времени дискретизации позволяет уменьшить время регулирования.



Пример программы 4:
Попробуем для ускорения процесса нагрева изменить время дискретизации на 2 сек и период импульсов ШИМ (D12 = K20, D30 = K2,000). Результат на графике ниже.
Хотя время еще более сократилось, на графике видны значительные колебания, как раз и обусловленные малым временем дискретизации, т.е. система становится слишком чувствительной.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
147	D	SWAP	P	(S)	Перестановка байтов в регистре				

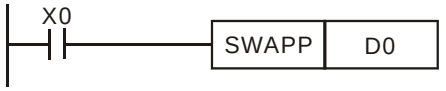
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*	*	*	SWAP, SWAPP: 3 шага DSWAP, DSWAPP: 5 шагов

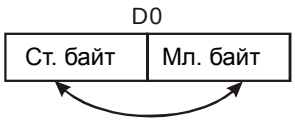
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Операнд, в котором происходит перестановка байтов.

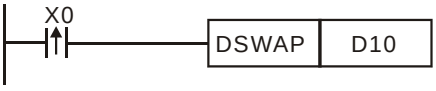
Описание:
1. Для 16-битной команды, старший байт и младший байт в регистре меняются местами.
2. Для 32-битной команды, перестановка байтов проводится для двух регистров отдельно.
3. Команда может выполняться вместе с импульсными командами (SWAPP, DSWAPP).
4. Если операнд D имеет значение F, возможно выполнение только 16-битных команд.

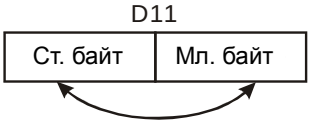
Пример программы 1:
Когда X0 включен, старший и младший байты D0 меняются местами.

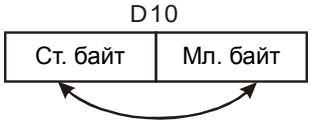




Пример программы 2:
Когда X0 включен, старший и младший байт D11 меняются местами, так же, как и стрший и младший байты D10.







API	Команда			Операнды	Функция	Контроллеры				
						ES2/EX2	SS2	SA2	SX2	SE
148		MEMR	P	(m) (D) (n)	Чтение данных из файлового регистра					

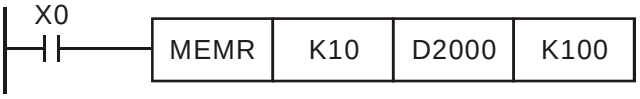
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
m					*	*							*			MEMR: 7 шагов 32-битная команда и DVP-SS2 не поддерживаются
D													*			
n					*	*							*			

ИМПУЛЬС					16-бит					32-бит				
ES2/EX2	SS2	SA2	SX2	SE	ES2/EX2	SS2	SA2	SX2	SE	ES2/EX2	SS2	SA2	SX2	SE

Операнды:
m: Начальный адрес файлового регистра для считывания данные (Диапазон: K0 - K4999)
D: Начальный адрес регистрахранения считанных данных (диапазон: D2000 - D9999)
n: Размер данных (Диапазон K1 - K5000)

Описание:
1. Копирование данных из файловых регистров в регистры общего назначения.
2. Существует 5 000 16-битных файловых регистров. Номер регистра в диапазоне K0 - K4999.
3. Если значения **m**, **D**, или **n** находятся вне пределов диапазона, возникает ошибка, команда не выполняется, включаются M1067 и M1068 и в D1067 записывается код ошибки H0E1A.
4. Если нет данных, записанных в файловый регистр, то значение по умолчанию, которое будет из него считываться равно -1.
5. Поддерживаются DVP-ES2/EX2/SS2 вер. 2.80 и выше, DVP-SA2/SX2 вер. 2.40 и выше. Не поддерживает DVP-ES2-C.
6. Файловые регистры не поддерживают M1101. Если пользователю необходимо прочитать данные из файлового регистра во время работы ПЛК, можно использовать LD M1002 и MEMR для чтения данных.

Пример программы:



1. Применение команды MEMR для чтения данных из 100 файловых регистров, начиная с десятого (K10), для получения данных в регистрах, начиная с D2000.
2. Когда X0 включен, команда выполняется. Когда X0 выключается, выполнение команды прекращается, и несчитанные регистры останутся не измененными.

API	Команда		Операнды	Функция	Контроллеры				
					ES2/EX2	SS2	SA2	SX2	SE
149	MEMW	P	(S) (m) (n)	Запись данных в файловый регистр					

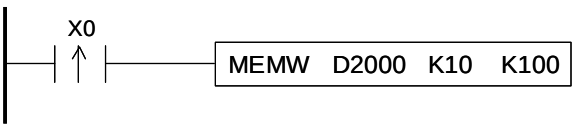
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S													*			MEMW: 7 шагов 32-битная команда и DVP-SS2 не поддерживаются
m					*	*							*			
n					*	*							*			

ИМПУЛЬС					16-бит					32-бит				
ES2/EX2	SS2	SA2	SX2	SE	ES2/EX2	SS2	SA2	SX2	SE	ES2/EX2	SS2	SA2	SX2	SE

Операнды:
S: Источник данных (Диапазон D2000 - D9999) **m**: Файловый регистр, в который записываются данные (Диапазон K0 - K4999) **n**: Размер данных (Диапазон K1 – K100)

Описание:
1. Копирование данных из регистров общего назначения в файловые регистры.
2. Существует 5 000 16-битных файловых регистров. Номер регистра в диапазоне K0 - K4999.
3. Если значения **m**, **D**, или **n** находятся вне пределов диапазона, возникает ошибка, команда не выполняется, включаются M1067 и M1068 и в D1067 записывается код ошибки H0E1A.
4. Т.к. в большинстве случаев файловые регистры используют флэш-память, в файловый регистр может быть записано 100 слов, и данные могут быть записаны в файловый регистр однократно только при включении управляющего контакта. Примечание: данные могут быть записаны в файловый регистр до 100,000 раз. Обратите на это внимание.
5. Поддерживаются DVP-ES2/EX2/SS2 вер. 2.80 и выше, DVP-SA2/SX2 вер. 2.40 и выше. Не поддерживает DVP-ES2-C.

Пример программы:



1. Применение команды MEMW для записи 100 регистров данных, начиная с D2000, в файловые регистры, начиная с десятого (K10).
2. Когда X0 включается, команда выполняется однократно.

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
150	MODRW	(S1) (S2) (S3) (S) (n)	Чтение/запись данных MODBUS				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			MODRW: 11 шагов
S_2					*	*							*			
S_3					*	*							*			
S													*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Адрес устройства (K1~K254) **S_2**: Функциональный код команды: K2(H02), K3(H03), K4(H04), K5(H05), K6(H06), K15(H0F), K16(H10), K23(H17) **S_3**, **S, n**: Назначение определяется применяемым функциональным кодом.

Описание:

1. Команда MODRW поддерживает порты COM1 (RS-232), COM2 (RS-485), COM3 (RS-485). COM3 – только для моделей DVP-ES2/EX2/SA2/SE. DVP-ES2-C – не поддерживается.
2. **S_1**: Адрес доступного устройства. Диапазон: K1~K254. Адреса кодов функций – K2, K3, K4, K23. K0 – некорректный адрес кода функции.
3. **S_2**: Функциональный код (в будущем возможно появление новых функциональных кодов)

Функциональный код	Описание	Поддерживаемые модели
H02	Чтение нескольких битовых устройств	Все модели
H03	Чтение нескольких слов	Все модели
H04	Чтение нескольких словных устройств	ES2/EX2 V2.6, SS2 V2.4, SA2/SX2 V2.0, SE V1.0 и выше
H05	Запись одиночного битного устройства	Все модели
H06	Запись одиночного слова	Все модели
H0F	Запись нескольких битовых устройств	Все модели
H10	Запись нескольких слов	Все модели
H17	Чтение/запись нескольких словных устройств	ES2/EX2 V3.2, SS2 V3.0, SA2 V2.6, SX2 V2.4 и выше

4. **S_3**: Адрес данных. Если адрес устройства назначен неправильно, будет возникать ошибка. ПЛК сохранит код ошибки и будет включен соответствующий флаг ошибки. Если функциональный код K23, то S3 может быть только регистром данных. При этом S3 является регистром, из которого данные считываются, S3+1 – регистром, в который данные записываются.

   Регистры и флаги, показывающие ошибку по портам: (для детальной информации см. **Заметки** описания команды API 80 RS)

ПЛК COM-порт	COM1	COM2	COM3
Флаг ошибки	M1315	M1141	M1319
Код ошибки	D1250	D1130	D1253

   Например, если 8000H является недопустимым для ПЛК, ошибка отображается соответствующими регистрами и флагами. Для COM2, включен M1141 и D1130 = 2; для COM1, включен M1315 и D1250 = 3, для COM3, включен M1319 и D1253 = 3.
5. **S**: Регистры для хранения считанных/записанных данных. Когда COM2 отправляет код функции чтения (K2/K3/K4), регистры из S непосредственно получают строки данных и преобразованные данные сохраняются в D1296~D1311. См. примеры программ 1 и 3. Когда COM1 или COM3 отправляет код функции чтения (K2/K3), регистры сохраняют преобразованные данные непосредственно. См. примеры программ 2 и 4. Если функциональный код K23, то S3 может быть только регистром данных. При этом S3 является регистром, из которого данные считываются, S3+1 – регистром, в который данные записываются. Если функциональный код для COM2 является K23, то полученные и преобразованные данные не сохраняются непосредственно в D1296 ~ D1311. См. примеры программ 13 и 14.
6. **n**: Длина получаемых данных.
   - Когда **S_2** (функциональный код MODBUS) имеет значение H05, которое обозначает состояние (ВКЛ/ВЫКЛ) ПЛК, **n** = 0 показывает ВКЛ и **n** = 1 показывает ВЫКЛ.
   - Когда **S_2** имеет значение H02, H03, H04, H0F, H10, H17, которые обозначают длину принимаемых данных, доступный диапазон будет **K1~Km**, где значение **m** определяется режимом коммуникации и COM-портов, см. таблицу ниже. (H02/H0F, ед. изм.: бит, H03/H04/H10/H17, ед. изм.: слово)

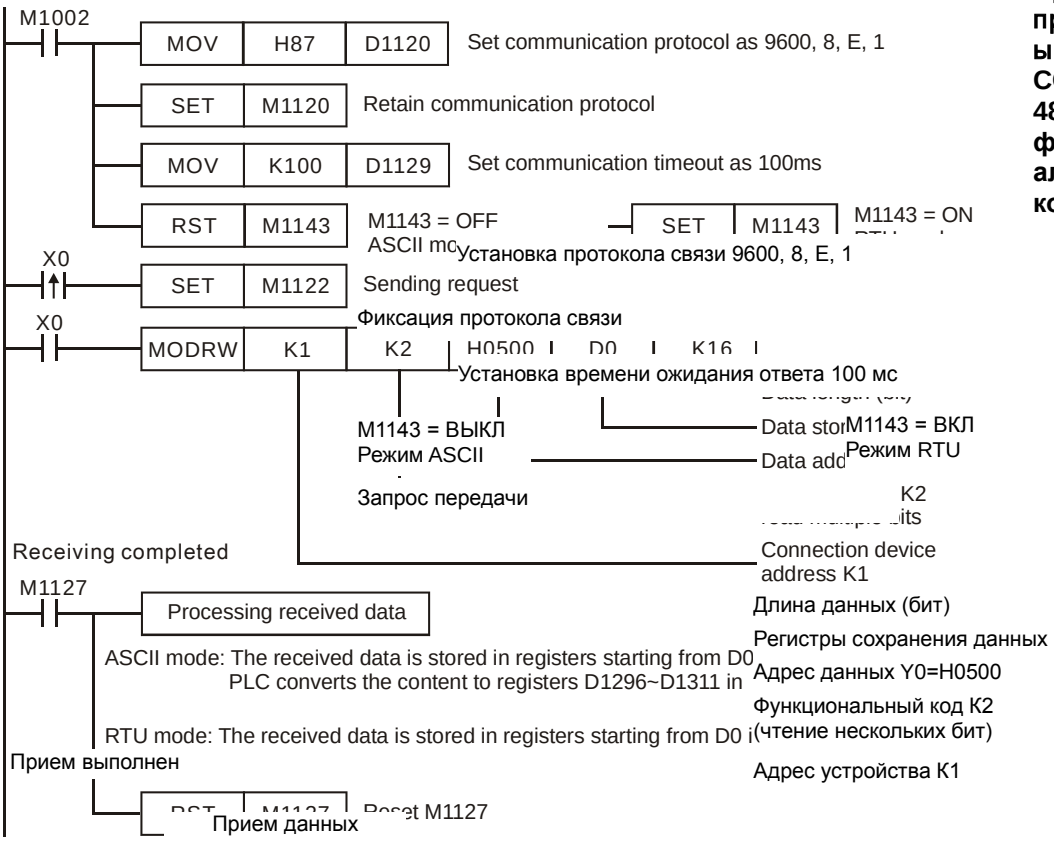
COM. режим	COM	H02	H03/H04	H0F	H10	H17
RTU	COM1	K 64	K 16	K 64	K 16	K 16
	COM2	K 64	K 16	K 64	K 16	K 16
	COM3	K 64	K 16	K 64	K 16	K 16
ASCII	COM1	K 64	K 16	K 64	K 16	K 16
	COM2	K 64	K 8	K 64	K 8	K 16
	COM3	K 64	K 16	K 64	K 16	K 16

7. Функции **S_3**, **S** и **n** в зависимости от применяемого функционального кода:

Функциональный код	S_3	S	n
H02	Адрес, с которого данные считываются	Регистр хранения считанных данных	Длина считываемых данных
H03			
H04			
H05	Адрес, на который данные записываются	Нет	Состояние записанного значения
H06		Регистр хранения записываемых данных	Нет
H0F		Регистр хранения записываемых данных	Длина записываемых данных
H10		Регистр хранения записываемых данных	Длина записываемых данных
H17	**S_3:** Адрес, с которого данные считываются **S_{3+1}:** Адрес, на который данные записываются	**S:** Регистр хранения считанных данных **S+1**: Регистр хранения записываемых данных	**n:** Длина считываемых данных **n+1:** Длина записываемых данных

8. Не существует ограничений на количество использований команды. Тем не менее, только одна команда может выполняться на одном COM-порту.
9. Нельзя использовать в качестве условия выполнения MODRW-инструкции (функциональный код H02, H03) импульсные команды (LDP, ANDP, ORP или LDF, ANDF, ORF). Иначе данные будут приняты не корректно.
10. Если импульсные команды LDP, ANDP, ORP, LDF, ANDF, ORF используются до команды MODWR, флаг запроса M1122(COM2) / M1312(COM1) / M1316(COM3) должен быть обязательно выключен.
11. Команда MODRW определяет COM-порты в соответствии с запросом коммуникации. Определение COM-портов осуществляется в соответствии с порядком: COM1→COM3→COM2. Поэтому необходимо задавать команду MODRW каждый раз сразу после отправки запроса во избежание ошибок расположения принятых данных.
12. Значения флагов и специальных регистров, см. **Заметки** API 80 команды RS.

Пример программы 1: COM2(RS-485), функциональный код H02



Режим ASCII: принятые данные сохраняются в регистрах, начиная с D0 в ASCII-формате, и ПЛК преобразует содержимое регистров D1296-D1311 в шестнадцатеричный формат автоматически

Режим RTU: принятые данные сохраняются в регистрах, начиная с D0 в шестнадцатеричном

1. Функциональный код K2 (H02): чтение всех битов, может быть прочитано до 64 бит.
2. ПЛК1 связан с ПЛК2: (M1143 выключен, ASCII-режим), (M1143 включен, RTU-режим)
3. В ASCII- или RTU-режиме, когда COM2 ПЛК пересылает данные, они сохраняются в D1256~D1295. Полученные данные сохраняются в регистрах, начиная с **S,** и автоматически преобразованные в 16-ричный формат в D1296~D1311.

Рассмотрим связь между ПЛК1 (COM2) и ПЛК2 (COM1) для примера: таблицы поясняют статус, когда ПЛК1 читает значения Y0~Y17 из ПЛК2.

ASCII-режим (M1143 выключен):

Когда X0 включен, ПЛК1⇨ ПЛК2, ПЛК1 посылает: "**01 02 0500 0010 E8**"
ПЛК2 ⇨ПЛК1, ПЛК1 принимает: **"01 02 02 3412 B5**"

Регистры переданных данных (переданные сообщения)

Регистр	Данные		Описание	
D1256 младший байт	'0'	30 H	ADR 1	Адрес устройства: ADR (1,0)
D1256 старший байт	'1'	31 H	ADR 0	
D1257 младший байт	'0'	30 H	CMD 1	Код команды: CMD (1,0)
D1257 старший байт	'2'	32 H	CMD 0	
D1258 младший байт	'0'	30 H	Y0 = H0500 Начальный адрес данных	
D1258 старший байт	'5'	35 H		
D1259 младший байт	'0'	30 H		
D1259 старший байт	'0'	30 H		
D1260 младший байт	'0'	30 H	Число данных (битов)	
D1260 старший байт	'0'	30 H		
D1261 младший байт	'1'	31 H		
D1261 старший байт	'0'	30 H		
D1262 младший байт	'E'	45 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D1262 старший байт	'8'	38 H	LRC CHK 0	

Регистры принятых данных (принятые сообщения)

Регистр	Данные		Описание	
D0 младший байт	'0'	30 H	ADR 1	
D0 старший байт	'1'	31 H	ADR 0	
D1 младший байт	'0'	30 H	CMD 1	
D1 старший байт	'2'	33 H	CMD 0	
D2 младший байт	'0'	30 H	Число данных (битов)	
D2 старший байт	'2'	32 H		
D3 младший байт	'3'	33 H	Содержимое данных 0500H~ 0515H	1234 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1296
D3 старший байт	'4'	34 H		
D4 младший байт	'1'	31H		
D4 старший байт	'2'	32H		
D5 младший байт	'B'	52H	LRC CHK 1	
D5 старший байт	'5'	35 H	LRC CHK 0	

Анализ чтения состояния ПЛК2 (Y0~Y17): 1234H

Устройство	Состояние	Устройство	Состояние	Устройство	Состояние	Устройство	Состояние
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

RTU-режим (M1143 включен):

Когда X0 включен, ПЛК1⇨ ПЛК2, ПЛК1 передает: "**01 02 0500 0010 79 0A**"
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: "**01 02 02 34 12 2F 75**"

Регистры переданных данных (переданные сообщения)

Регистр	Данные	Описание
D1256 младший байт	01 H	Адрес
D1257 младший байт	02 H	Функция
D1258 младший байт	05 H	Y0 = H0500 Начальный адрес данных
D1259 младший байт	00 H	
D1260 младший байт	00 H	Число данных (слов)
D1261 младший байт	10 H	
D1262 младший байт	79 H	CRC CHK младш.
D1263 младший байт	0A H	CRC CHK старш.

Регистры принятых данных (принятые сообщения)

Регистр	Данные	Описание
D0	1234 H	ПЛК сохраняет значение 1234H в D1296
D1 младший байт	02 H	Функция
D2 младший байт	02 H	Число данных (байтов)
D3 младший байт	34 H	Содержимое данных H0500~H0515
D4 младший байт	12 H	
D5 младший байт	2F H	CRC CHK младш.
D6 младший байт	75 H	CRC CHK старш.

Анализ чтения состояния ПЛК2 (Y0~Y17): 1234H

Устройство	Состояние	Устройство	Состояние	Устройство	Состояние	Устройство	Состояние
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

Пример программы 2: COM1(RS-232) / COM3(RS-485), функциональный код H02

Функциональный код K2 (H02): чтение всех битов, может быть прочитано до 64 бит.

1. ПЛК1 связан с ПЛК2: (M1143 выключен, ASCII-режим), (M1143 включен, RTU-режим)
2. В обоих режимах - ASCII и RTU, ПЛК COM1/COM3 сохраняет только принятые данные в регистрах, начиная с **S**, и не будет сохранять отправленные данные. Сохраненные данные могут быть преобразованы и перенесены с помощью команды DTM для последующего использования.
3. Рассмотрим связь между ПЛК1 (COM2) и ПЛК2 (COM1) для примера: таблицы поясняют статус, когда ПЛК1 читает значения Y0~Y17 из ПЛК2.

   Если ПЛК1 применяет COM1 для связи, см. программу ниже (изменения):

   1. D1109→D1036: протокол связи
   2. M1136→M1138: сохранение параметров связи.
   3. D1252→D1249: установка задержки получения данных
   4. M1320→M1139: ASCII/RTU – выбор режима
   5. M1316→M1312: запрос передачи данных
   6. M1318→M1314: флаги завершения приема



ASCII-режим (COM3: M1320 выключен, COM1: M1139 выключен):
Когда X0 включен, ПЛК1⇨ ПЛК2, ПЛК1 передает: "**01 02 0500 0010 E8**"
ПЛК2 ⇨PLC1, ПЛК1 принимает: "**01 02 02 3412 B5**"

ПЛК1: регистр принимаемых данных D0

Регистр	Данные	Описание
D0	1234H	ПЛК преобразует ASCII данные из адресов 0500H~0515H и сохраняет преобразованные данные автоматически.

Анализ чтения состояния ПЛК2 (Y0~Y17): 1234H

Устройство	Состояние	Устройство	Состояние	Устройство	Состояние	Устройство	Состояние
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

RTU-режим (COM3: M1320 включен, COM1: M1139 включен):
Когда X0 включен, ПЛК1⇨ ПЛК2, ПЛК1 передает: "**01 02 0500 0010 79 0A**"
ПЛК2 ⇨PLC1, ПЛК1 принимает: "**01 02 02 34 12 2F 75**"

Регистры принимаемых данных:

Регистр	Данные	Описание
D0	1234 H	ПЛК преобразует ASCII данные из адресов 0500H~0515H и сохраняет преобразованные данные автоматически

Анализ чтения состояния ПЛК2 (Y0~Y17): 1234H

Устройство	Состояние	Устройство	Состояние	Устройство	Состояние	Устройство	Состояние
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

4. Флаги и регистры COM1 / COM2 / COM3, работающих в мастер-режиме:

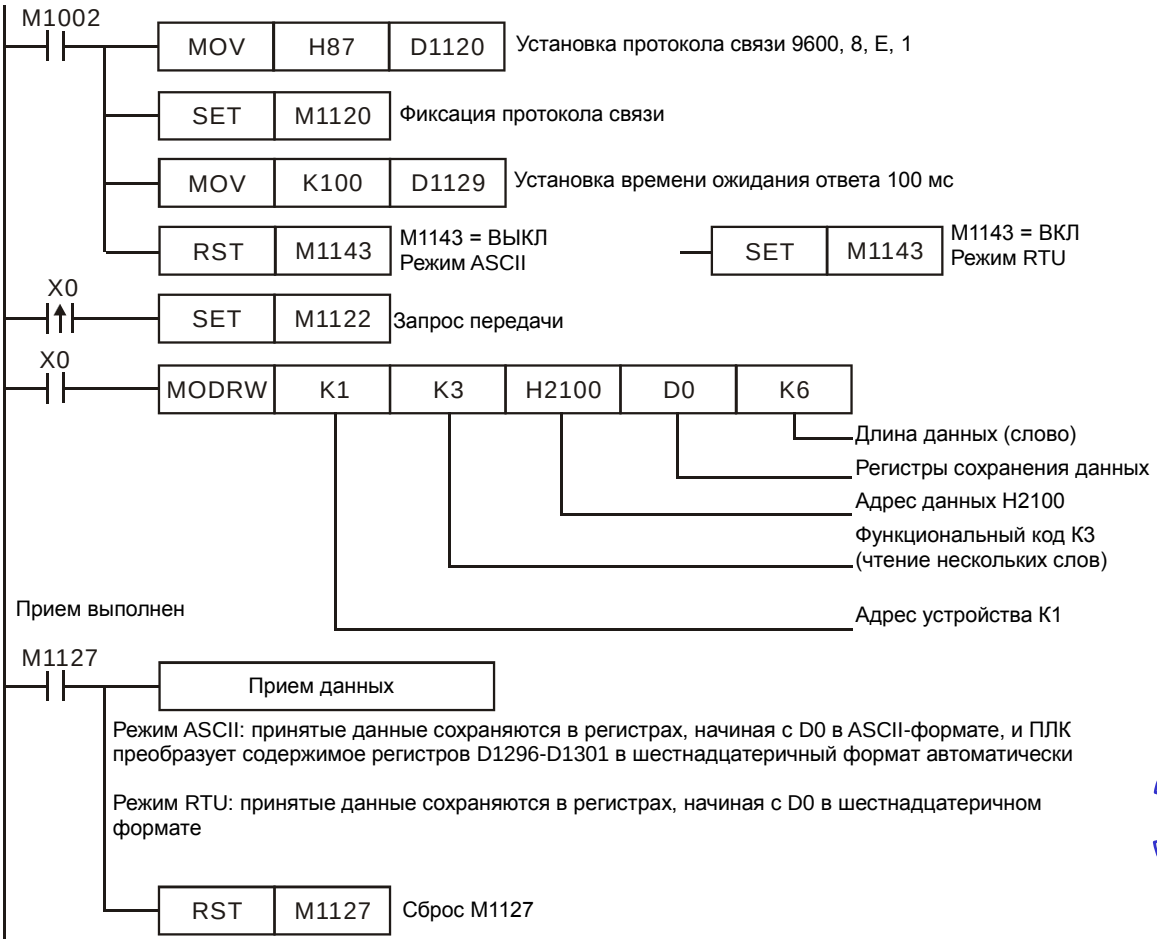
	COM2	COM1	COM3	Функция
Установка связи	M1120	M1138	M1136	Сохранение параметров связи.
	M1143	M1139	M1320	ASCII/RTU выбор режима
	D1120	D1036	D1109	Протокол связи
	D1121	D1121	D1255	Коммуникационный адрес ПЛК
Запрос передачи данных	M1122	M1312	M1316	Запрос передачи данных
	D1129	D1249	D1252	Установка времени задержки передачи (мс)
Получение данных завершено	M1127	M1314	M1318	Завершение приема данных
Ошибки	-	M1315	M1319	Ошибка получения данных
	-	D1250	D1253	Код ошибки связи
	M1129	-	-	Задержка приема
	M1140	-	-	Ошибка принятых данных
	M1141	-	-	Параметрическая ошибка. Код сохраняется в D1130
	D1130	-	-	Код ошибки коммуникации Modbus

Пример программы 3: COM2 (RS-485), Функциональный код **H03**

1. Функциональный код K3 (H03): чтение всех слов. Может быть прочитано до 16 слов. Для COM2 в ASCII-режиме – до 8 слов.
2. Для ASCII или RTU режимов, ПЛК (COM2) сохраняет передаваемые данные в D1256~D1295, конвертирует принятые данные в регистрах, начиная с **S**, и сохраняет преобразованные 16-битные данные в D1296 ~ D1311.

Рассмотрим связь между ПЛК (COM2) и VFD-B: таблицы показывают статус, когда ПЛК считывает состояние VFD-B. (M1143 выключен, ASCII-режим) (M1143 включен, RTU-режим)

ASCII-режим (M1143 выключен):
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: "01 03 2100 0006 D5"
VFD-B ⇨ ПЛК, ПЛК принимает: "01 03 0C 0100 1766 0000 0000 0136 0000 3B"



Регистры переданных данных (переданные сообщения)

Регистр	Данные		Описание	
D1256 младший байт	'0'	30 H	ADR 1	Адрес VFD-B: ADR (1,0)
D1256 старший байт	'1'	31 H	ADR 0	
D1257 младший байт	'0'	30 H	CMD 1	Код команды: CMD (1,0)
D1257 старший байт	'3'	33 H	CMD 0	
D1258 младший байт	'2'	32 H	Адрес данных	
D1258 старший байт	'1'	31 H		
D1259 младший байт	'0'	30 H		
D1259 старший байт	'0'	30 H		
D1260 младший байт	'0'	30 H	Число данных (слов)	
D1260 старший байт	'0'	30 H		
D1261 младший байт	'0'	30 H		
D1261 старший байт	'6'	36 H		
D1262 младший байт	'D'	44 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D1262 старший байт	'5'	35 H	LRC CHK 0	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание	
D0 младший байт	'0'	30 H	ADR 1	
D0 старший байт	'1'	31 H	ADR 0	
D1 младший байт	'0'	30 H	CMD 1	
D1 старший байт	'3'	33 H	CMD 0	
D2 младший байт	'0'	30 H	Число данных (байтов)	
D2 старший байт	'C'	43 H		
D3 младший байт	'0'	30 H	Содержимое данных H2100	0100 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1296
D3 старший байт	'1'	31 H		
D4 младший байт	'0'	30 H		
D4 старший байт	'0'	30 H		
D5 младший байт	'1'	31 H	Содержимое данных H2101	1766 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1297
D5 старший байт	'7'	37 H		
D6 младший байт	'6'	36 H		
D6 старший байт	'6'	36 H		
D7 младший байт	'0'	30 H	Содержимое данных H2102	0000 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1298
D7 старший байт	'0'	30 H		
D8 младший байт	'0'	30 H		
D8 старший байт	'0'	30 H		

Регистр	Данные		Описание	
D9 младший байт	‘0’	30 H	Содержимое данных H2103	0000 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1299
D9 старший байт	‘0’	30 H		
D10 младший байт	‘0’	30 H		
D10 старший байт	‘0’	30 H		
D11 младший байт	‘0’	30 H	Содержимое данных H2104	0136 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1300
D11 старший байт	‘1’	31 H		
D12 младший байт	‘3’	33 H		
D12 старший байт	‘6’	36 H		
D13 младший байт	‘0’	30 H	Содержимое данных H2105	0000 H ПЛК автоматически преобразует ASCII-коды и сохраняет результат в D1301
D13 старший байт	‘0’	30 H		
D14 младший байт	‘0’	30 H		
D14 старший байт	‘0’	30 H		
D15 младший байт	‘3’	33 H	LRC CHK 1	
D15 старший байт	‘B’	42 H	LRC CHK 0	

RTU-режим (M1143 включен):

Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: **” 01 03 2100 0006 CF F4”**

VFD-B ⇨ ПЛК, ПЛК принимает: **“01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5”**

Регистры переданных данных (переданные сообщения)

Регистр	Данные	Описание
D1256 младший байт	01 H	Адрес
D1257 младший байт	03 H	Функция
D1258 младший байт	21 H	Адрес данных
D1259 младший байт	00 H	
D1260 младший байт	00 H	Число данных (слов)
D1261 младший байт	06 H	
D1262 младший байт	CF H	CRC CHK младш.
D1263 младший байт	F4 H	CRC CHK старш.

Регистры принимаемых данных (принимаемые сообщения)

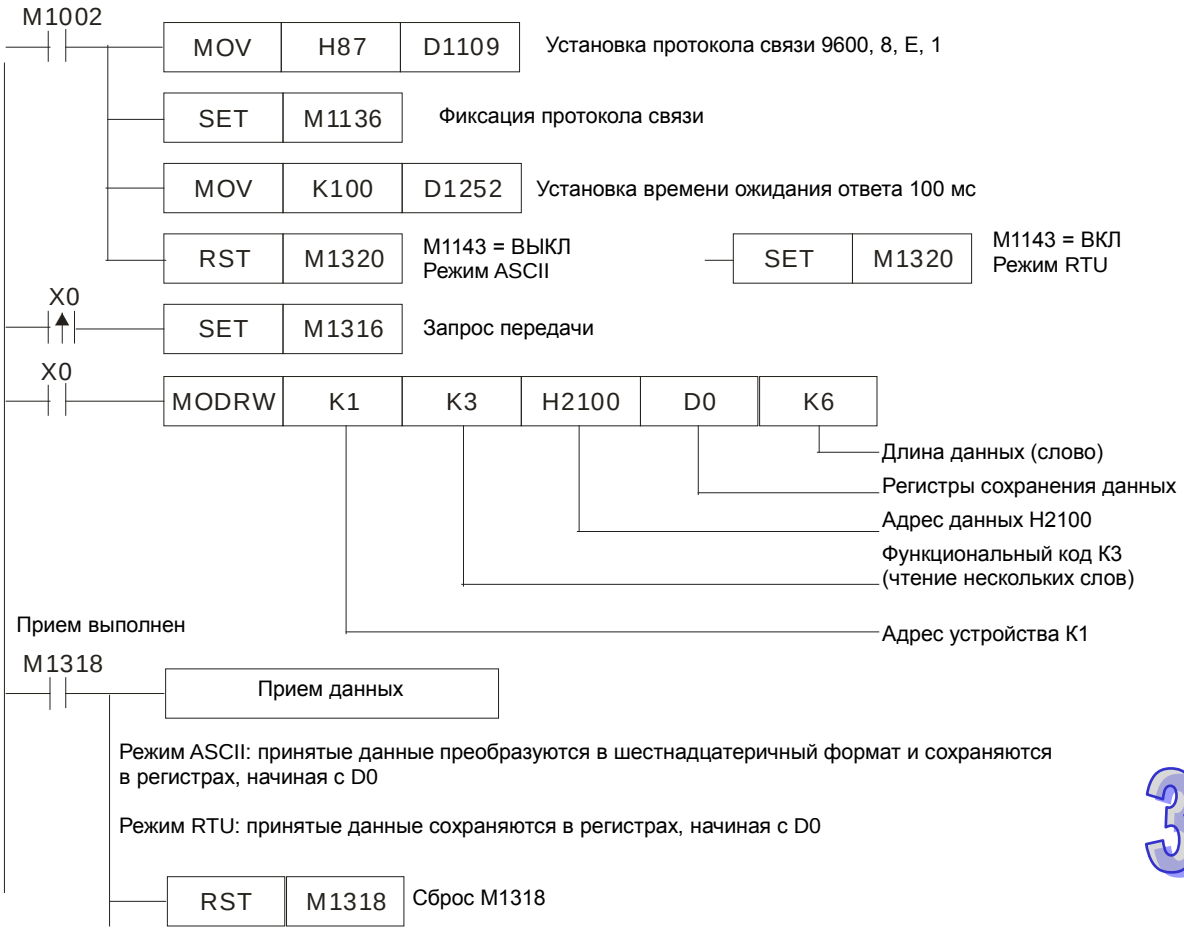
Регистр	Данные	Описание	
D0 младший байт	01 H	Адрес	
D1 младший байт	03 H	Функция	
D2 младший байт	0C H	Число данных (байтов)	
D3 младший байт	00 H	Содержимое данных H2100	0000 H ПЛК COM2 автоматически сохраняет значение в D1296
D4 младший байт	00 H		
D5 младший байт	05 H	Содержимое данных H2101	0503 H ПЛК COM2 автоматически сохраняет значение в D1297
D6 младший байт	03 H		
D7 младший байт	0B H	Содержимое данных H2102	0BB8 H ПЛК COM2 автоматически сохраняет значение в D1298
D8 младший байт	B8 H		
D9 младший байт	0B H	Содержимое данных H2103	0BB8 H ПЛК COM2 автоматически сохраняет значение в D1299
D10 младший байт	B8 H		
D11 младший байт	00 H	Содержимое данных H2104	0000 H ПЛК COM2 автоматически сохраняет значение в D1300
D12 младший байт	00 H		
D13 младший байт	01 H	Содержимое данных H2105	012D H ПЛК COM2 автоматически сохраняет значение в D1301
D14 младший байт	2D H		
D15 младший байт	8E H	CRC CHK младш.	
D16 младший байт	C5 H	CRC CHK старш.	

Пример программы 4: COM1(RS-232) / COM3(RS-485), функциональный код H03

1. Функциональный код K3 (H03): чтение всех слов. Может быть прочитано до 16 слов. Для COM2 в ASCII-режиме – до 8 слов.
2. ПЛК (COM1 / COM3) сохраняет принятые данные в регистрах, начиная с **S**, Сохраненные данные могут быть преобразованы и перенесены с помощью команды DTM для последующего использования.
3. Рассмотрим связь между ПЛК и VFD-B для примера: таблицы поясняют статус, когда ПЛК читает состояние VFD-B (M1320 выключен, ASCII-режим), (M1320 включен, RTU-режим).

Если ПЛК1 применяет COM1 для связи, см. программу ниже (изменения):

1. D1109→D1036: протокол связи
2. M1136→M1138: сохранение параметров связи.
3. D1252→D1249: установка задержки получения данных
4. M1320→M1139: ASCII/RTU – выбор режима
5. M1316→M1312: запрос передачи данных
6. M1318→M1314: флаги завершения приема



ASCII mode (COM3: M1320 = OFF, COM1: M1139 = OFF):
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: "**01 03 2100 0006 D5**"
VFD-B ⇨ ПЛК, ПЛК принимает: "**01 03 0C 0100 1766 0000 0000 0136 0000 3B**"

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные	Описание
D0	0100 H	ПЛК преобразует ASCII-коды в 2100 H и сохраняет преобразованные данные автоматически
D1	1766 H	ПЛК преобразует ASCII-коды в 2101 H и сохраняет преобразованные данные автоматически
D2	0000 H	ПЛК преобразует ASCII-коды в 2102 H и сохраняет преобразованные данные автоматически
D3	0000 H	ПЛК преобразует ASCII-коды в 2103 H и сохраняет преобразованные данные автоматически
D4	0136 H	ПЛК преобразует ASCII-коды в 2104 H и сохраняет преобразованные данные автоматически
D5	0000 H	ПЛК преобразует ASCII-коды в 2105 H и сохраняет преобразованные данные автоматически

RTU-режим (COM3: M1320 включен COM1: M1139 включен):
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: " **01 03 2100 0006 CF F4**"
VFD-B ⇨ ПЛК, ПЛК принимает: "**01 03 0C 0000 0503 0BB8 0BB8 0000 012D 8E C5**"

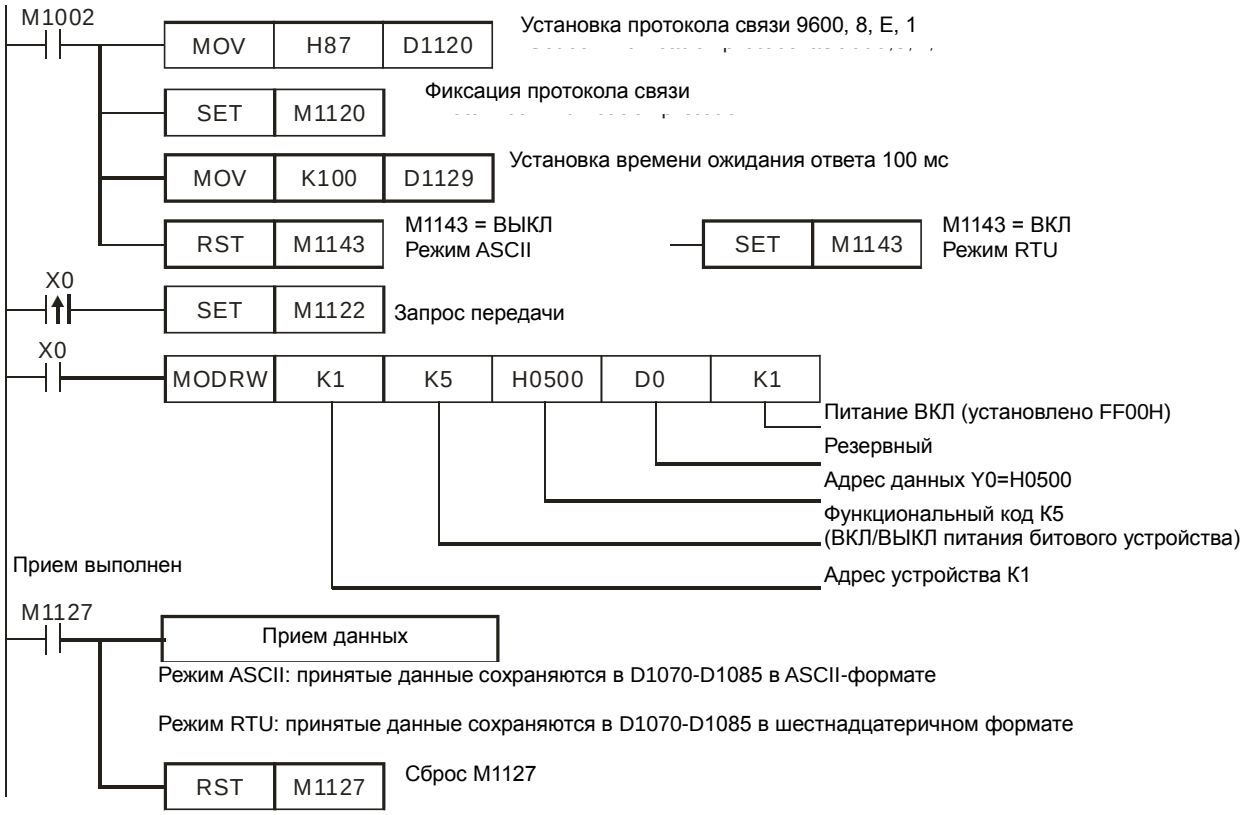
Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные	Описание
D0	0000 H	ПЛК преобразует ASCII-коды в 2100 H и сохраняет преобразованные данные автоматически
D1	0503 H	ПЛК преобразует ASCII-коды в 2101 H и сохраняет преобразованные данные автоматически
D2	0BB8 H	ПЛК преобразует ASCII-коды в 2102 H и сохраняет преобразованные данные автоматически
D3	0BB8 H	ПЛК преобразует ASCII-коды в 2103 H и сохраняет преобразованные данные автоматически
D4	0136 H	ПЛК преобразует ASCII-коды в 2104 H и сохраняет преобразованные данные автоматически
D5	012D H	ПЛК преобразует ASCII-коды в 2105 H и сохраняет преобразованные данные автоматически

Пример программы 5: COM2(RS-485), функциональный код H05

Функциональный код K5(H05): ВКЛ/ВЫКЛ битового устройства

1. ПЛК1 связан с ПЛК2: (M1143 выключен, ASCII-режим), (M1143 включен, RTU-режим)
2. **n** = 1 показывает ВКЛ (установка FF00H) и **n** = 0 показывает ВЫКЛ (установка 0000H)
3. Для режимов ASCII или RTU: ПЛК (COM2) сохраняет передаваемые данные в D1256~D1295 и принимаемые данные в D1070~D1085
4. Рассмотрим связь между ПЛК1 (COM2) и ПЛК2 (COM1), таблица показывает состояния, когда ПЛК1 ВКЛЮЧАЕТ ПЛК2 Y0.



ASCII-режим (M1143 выключен):

Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: "**01 05 0500 FF00 6F**"

ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: "**01 05 0500 FF00 6F**"

Регистры переданных данных (переданные сообщения)

Регистр	Данные		Описание	
D1256 младший байт	'0'	30 H	ADR 1	Адрес устройства: ADR (1,0)
D1256 старший байт	'1'	31 H	ADR 0	
D1257 младший байт	'0'	30 H	CMD 1	CMD (1,0) Код команды
D1257 старший байт	'5'	35H	CMD 0	
D1258 младший байт	'0'	30 H	Адрес данных	
D1258 старший байт	'5'	35 H		
D1259 младший байт	'0'	30 H		
D1259 старший байт	'0'	30 H		
D1260 младший байт	'F'	46 H	Старший байт ВКЛ/ВЫКЛ	
D1260 старший байт	'F'	46 H		
D1261 младший байт	'0'	30H	Младший байт ВКЛ/ВЫКЛ	
D1261 старший байт	'0'	30 H		
D1262 младший байт	'6'	36 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D1262 старший байт	'F'	46 H	LRC CHK 0	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание
D1070 младший байт	'0'	30 H	ADR 1
D1070 старший байт	'1'	31 H	ADR 0
D1071 младший байт	'0'	30 H	CMD 1
D1071 старший байт	'5'	35H	CMD 0

D1072 младший байт	‘0’	30 H	Адрес данных
D1072 старший байт	‘5’	35 H	
D1073 младший байт	‘0’	30 H	
D1073 старший байт	‘0’	30 H	
D1074 младший байт	‘F’	46 H	Старший байт ВКЛ/ВЫКЛ
D1074 старший байт	‘F’	46 H	
D1075 младший байт	‘0’	30H	Младший байт ВКЛ/ВЫКЛ
D1075 старший байт	‘0’	30 H	
D1076 младший байт	‘6’	36 H	LRC CHK 1
D1076 старший байт	‘F’	46 H	LRC CHK 0

RTU-режим (M1143 включен)
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: “**01 05 0500 FF00 8C F6**”
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: “**01 05 0500 FF00 8C F6**”

Регистры переданных данных (переданные сообщения)

Регистр	Данные	Описание
D1256 младший байт	01 H	Адрес
D1257 младший байт	05 H	Функция
D1258 младший байт	05 H	Адрес данных
D1259 младший байт	00 H	
D1260 младший байт	FF H	Содержимое данных (ВКЛ = FF00H)
D1261 младший байт	00 H	
D1262 младший байт	8C H	CRC CHK младш.
D1263 младший байт	F6 H	CRC CHK старш.

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные	Описание
D1070 младший байт	01 H	Адрес
D1071 младший байт	05 H	Функция
D1072 младший байт	05 H	Адрес данных
D1073 младший байт	00 H	
D1074 младший байт	FF H	Содержимое данных (ВКЛ = FF00H)
D1075 младший байт	00 H	
D1076 младший байт	8C H	CRC CHK младш.
D1077 младший байт	F6 H	CRC CHK старш.

Пример программы 6: COM1(RS-232) / COM3(RS-485), функциональный код H05

Функциональный код K5(H05): ВКЛ/ВЫКЛ битового устройства

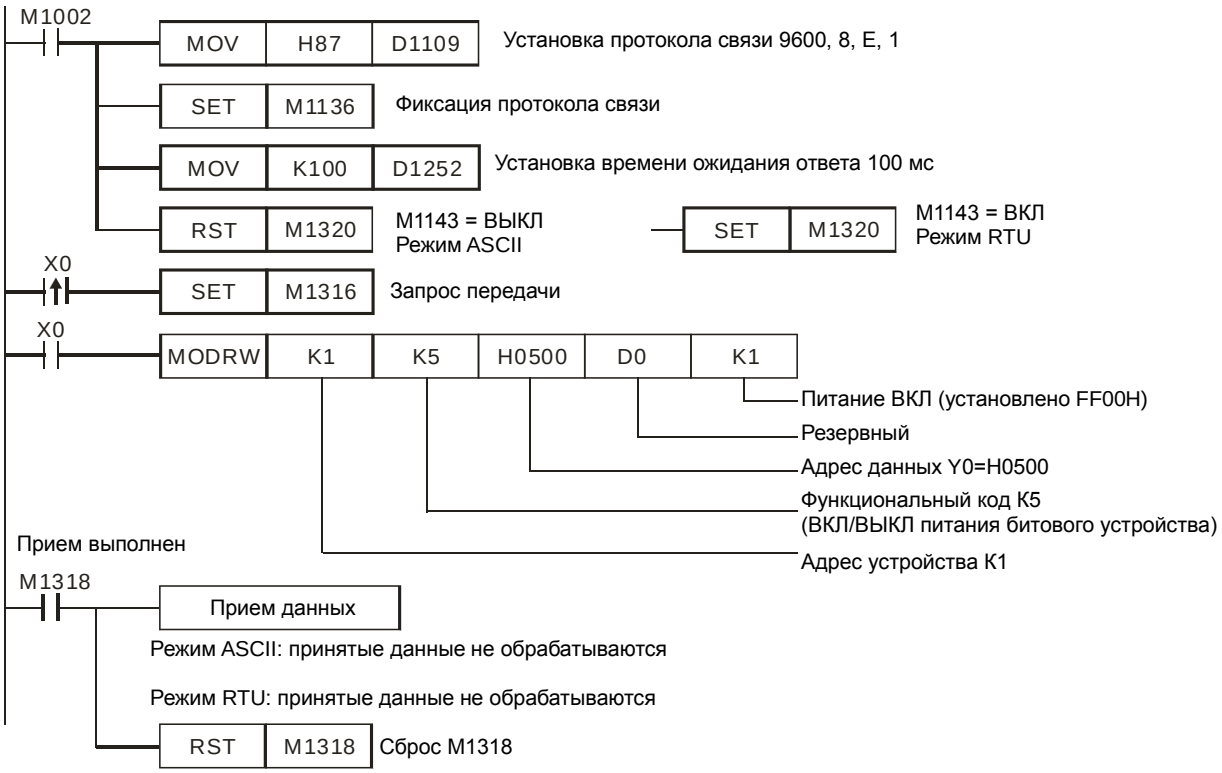
1. ПЛК1 связан с ПЛК2: (M1143 выключен, ASCII-режим), (M1143 включен, RTU-режим)
2. **n** = 1 показывает ВКЛ (установка FF00H) и **n** = 0 показывает ВЫКЛ (установка 0000H)
3. ПЛК (COM1/COM3) не будет обрабатывать полученные данные.
4. Рассмотрим связь между ПЛК1 (COM2) и ПЛК2 (COM1), таблица показывает состояния, когда ПЛК1 ВКЛЮЧАЕТ ПЛК2 Y0-Y17.

   Если ПЛК1 применяет COM1 для связи, см. программу ниже (изменения):

   1. D1109→D1036: протокол связи
   2. M1136→M1138: сохранение параметров связи.
   3. D1252→D1249: установка задержки получения данных
   4. M1320→M1139: ASCII/RTU – выбор режима
   5. M1316→M1312: запрос передачи данных
   6. M1318→M1314: флаги завершения приема

ASCII-режим (COM3: M1320 выключен, COM1: M1139 выключен):
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: “**01 05 0500 FF00 6F**”
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: “**01 05 0500 FF00 6F**”
(Полученные данные не обрабатываются)

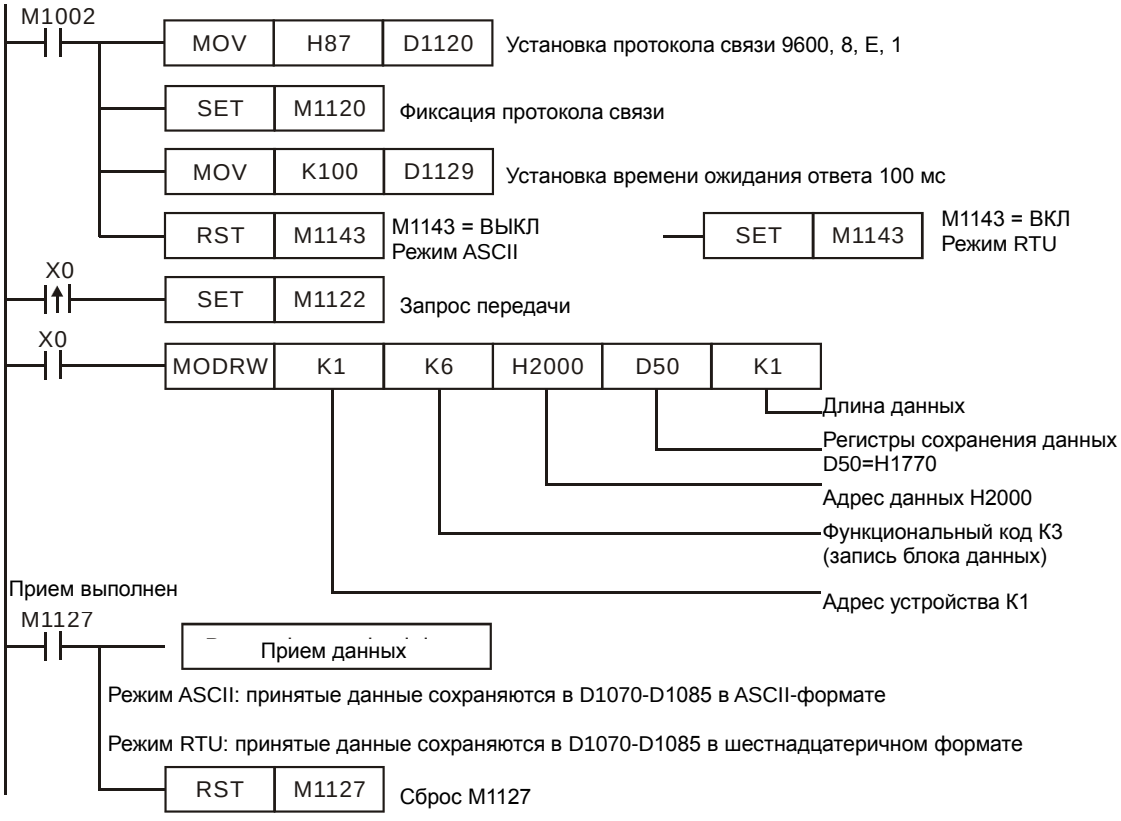
RTU-режим (COM3: M1320 включен, COM1: M1139 включен):
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: “**01 05 0500 FF00 8C F6**”
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: “**01 05 0500 FF00 8C F6**”
(Полученные данные не обрабатываются)



Пример программы 7: COM2(RS-485), функциональный код H06

Функциональный код K6 (H06): чтение одного слова.

1. Установленные значения должны быть записаны в VFD-B в регистры, назначенные для операнда **S**.
2. Для режимов ASCII или RTU: ПЛК (COM2) сохраняет передаваемые данные в D1256~D1295, а принимаемые в D1070~D1085.
3. Рассмотрим связь между ПЛК и VFD-B для примера: таблицы поясняют статус, когда ПЛК читает состояние VFD-B (M1320 выключен, ASCII-режим), (M1320 включен, RTU-режим).



ASCII-режим (M1143 выключен)
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: **“01 06 2000 1770 52”**
VFD-B ⇨ ПЛК, ПЛК принимает: **“01 06 2000 1770 52”**

Регистры передаваемых данных (передаваемые сообщения)

Регистр	Данные		Описание	
D1256 младший байт	‘0’	30 H	ADR 1 ADR 0	Адрес устройства VFD-B: ADR (1,0)
D1256 старший байт	‘1’	31 H		
D1257 младший байт	‘0’	30 H	CMD 1 CMD 0	Командный код: CMD (1,0)
D1257 старший байт	‘6’	36 H		
D1258 младший байт	‘2’	32 H	Адрес данных	
D1258 старший байт	‘0’	30 H		
D1259 младший байт	‘0’	30 H		
D1259 старший байт	‘0’	30 H		
D1260 младший байт	‘1’	31 H	Содержимое данных	H1770 = K6000. Содержимое регистра D50
D1260 старший байт	‘7’	37 H		
D1261 младший байт	‘7’	37 H		
D1261 старший байт	‘0’	30 H		
D1262 младший байт	‘5’	35 H	LRC CHK 1 LRC CHK 0	Контрольная сумма: LRC CHK (0,1)
D1262 старший байт	‘2’	32 H		

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание
D1070 младший байт	‘0’	30 H	ADR 1 ADR 0
D1070 старший байт	‘1’	31 H	
D1071 младший байт	‘0’	30 H	CMD 1 CMD 0
D1071 старший байт	‘6’	36 H	
D1072 младший байт	‘2’	32 H	Адрес данных
D1072 старший байт	‘0’	30 H	
D1073 младший байт	‘0’	30 H	
D1073 старший байт	‘0’	30 H	
D1074 младший байт	‘1’	31 H	Содержимое данных
D1074 старший байт	‘7’	37 H	
D1075 младший байт	‘7’	37 H	
D1075 старший байт	‘0’	30 H	
D1076 младший байт	‘6’	36 H	LRC CHK 1 LRC CHK 0
D1076 старший байт	‘5’	35 H	

RTU-режим (M1143 включен)
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: **“01 06 2000 1770 8C 1E”**
VFD-B → ПЛК, ПЛК принимает: “**01 06 2000 1770 8C 1E**”

Регистры передаваемых данных (передаваемые сообщения)

Регистр	Данные	Описание	
D1256 младший байт	01 H	Адрес	
D1257 младший байт	06 H	Функция	
D1258 младший байт	20 H	Адрес данных	
D1259 младший байт	00 H		
D1260 младший байт	17 H	Содержимое данных	H1770 = K6000. Содержимое регистра D50
D1261 младший байт	70 H		
D1262 младший байт	8C H	CRC CHK младш.	
D1263 младший байт	1E H	CRC CHK старш.	

Регистры принимаемых данных (принимаемые сообщения)

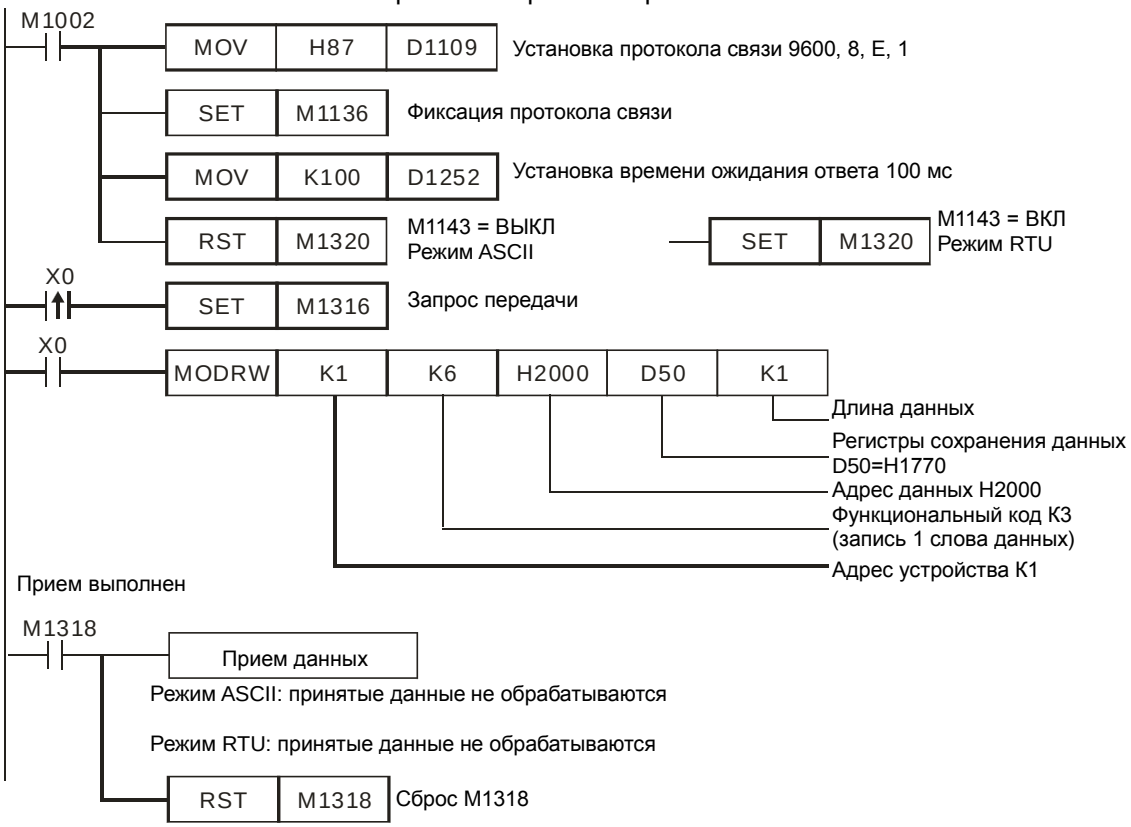
Регистр	Данные	Описание
D1070 младший байт	01 H	Адрес
D1071 младший байт	06 H	Функция
D1072 младший байт	20 H	Адрес данных
D1073 младший байт	00 H	
D1074 младший байт	17 H	Содержимое данных
D1075 младший байт	70 H	
D1076 младший байт	8C H	CRC CHK младш.
D1077 младший байт	1E H	CRC CHK старш.

Пример программы **8: COM1 (RS-232) / COM3 (RS-485), функциональный код H06**

1. Функциональный код K6 (H06): чтение одного слова.
2. Установленные значения должны быть записаны в VFD-B в регистры, назначенные для операнда **S**.
3. ПЛК COM1/COM3 не обрабатывает принимаемые данные.
4. Рассмотрим связь между ПЛК (COM3) и VFD-B для примера: таблицы поясняют статус, когда ПЛК (COM3) читает одно слово из VFD-B (M1320 выключен, ASCII-режим), (M1320 включен, RTU-режим).

Если ПЛК1 применяет COM1 для связи, см. программу ниже (изменения):

1. D1109→D1036: протокол связи
2. M1136→M1138: сохранение параметров связи.
3. D1252→D1249: установка задержки получения данных
4. M1320→M1139: ASCII/RTU – выбор режима
5. M1316→M1312: запрос передачи данных
6. M1318→M1314: флаги завершения приема



ASCII-режим (COM3: M1320 выключен, COM1: M1139 выключен):
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: **“01 06 2000 1770 52”**
VFD-B ⇨ ПЛК, ПЛК принимает: **“01 06 2000 1770 52”**
(Принимаемые данные не обрабатываются)

RTU-режим (COM3: M1320 включен, COM1: M1139 включен)
Когда X0 включен, ПЛК ⇨ VFD-B, ПЛК передает: **“01 06 2000 1770 8C 1E”**
VFD-B ⇨ ПЛК, ПЛК принимает: “**01 06 2000 1770 8C 1E**”
(Принимаемые данные не обрабатываются)

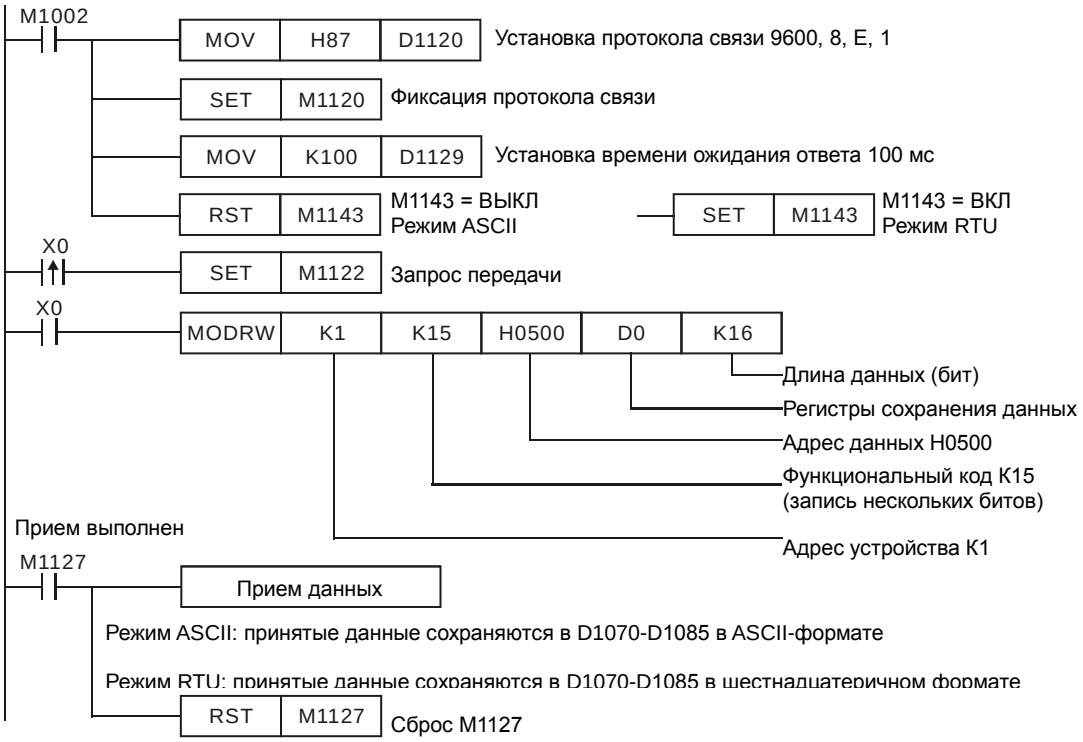
Пример программы 9: COM2 (RS-485), функциональный код H0F

Функциональный код K15 (H0F): чтение нескольких битов. Может быть прочитано до 64 битов.

1. ПЛК1 соединяется с ПЛК2: (M1143 выключен, ASCII-режим), (M1143 включен, RTU-режим)
2. Для режимов ASCII или RTU, ПЛК (COM2) сохраняет передаваемые данные в D1256~D1295, D1070~D1085.
3. Рассмотрим связь между ПЛК1 (COM2) и ПЛК2 (COM1), таблица показывает состояния, когда ПЛК1 ВКЛЮЧАЕТ ПЛК2 Y0-Y17.
   Установленное значение: K4Y0=1234H

Устройство	Состояние	Устройство	Состояние	Устройство	Состояние	Устройство	Состояние
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

ASCII-режим (M1143 выключен)
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: “ **01 0F 0500 0010 02 3412 93** ”
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: “ **01 0F 0500 0010 DB** ”



Регистры передаваемых данных (передаваемые сообщения)

Регистр	Данные		Описание	
D1256 младший байт	‘0’	30 H	ADR 1 ADR 0	Адрес устройства: ADR (1,0)
D1256 старший байт	‘1’	31 H		
D1257 младший байт	‘0’	30 H	CMD 1 CMD 0	Командный код: CMD (1,0)
D1257 старший байт	‘F’	46 H		
D1258 младший байт	‘0’	30 H	Адрес данных	
D1258 старший байт	‘5’	35 H		
D1259 младший байт	‘0’	30 H		
D1259 старший байт	‘0’	30 H		
D1260 младший байт	‘0’	30 H	Размер данных (биты)	
D1260 старший байт	‘0’	30 H		
D1261 младший байт	‘1’	31H		
D1261 старший байт	‘0’	30 H		
D1262 младший байт	‘0’	30 H	Количество байт	
D1262 старший байт	‘2’	32 H		
D1263 младший байт	‘3’	33 H	Содержимое данных	1234H Содержимое регистра D0
D1263 старший байт	‘4’	46 H		
D1264 младший байт	‘1’	33 H		
D1264 старший байт	‘2’	46 H		
D1265младший байт	‘9’	39 H	LRC CHK 1	Контрольная сумма: LRC CHK (0,1)
D1265 старший байт	‘3’	33 H	LRC CHK 0	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание
D1070 младший байт	‘0’	30 H	ADR 1 ADR 0
D1070 старший байт	‘1’	31 H	
D1071 младший байт	‘0’	31 H	CMD 1 CMD 0
D1071 старший байт	‘F’	46 H	
D1072 младший байт	‘0’	30 H	Адрес данных
D1072 старший байт	‘5’	35 H	
D1073 младший байт	‘0’	30 H	
D1073 старший байт	‘0’	30 H	
D1074младший байт	‘0’	30 H	Размер данных (биты)
D1074 старший байт	‘0’	30 H	
D1075 младший байт	‘1’	31 H	
D1075 старший байт	‘0’	30 H	
D1076 младший байт	‘D’	44 H	LRC CHK 1 LRC CHK 0
D1076 старший байт	‘B’	42 H	

RTU mode (M1143 = ON)
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: "**01 0F 0500 0010 02 34 12 21 ED**"
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: "**01 0F 0500 0010 54 CB**"

Регистры передаваемых данных (передаваемые сообщения)

Регистр	Данные	Описание	
D1256 младший байт	01 H	Адрес	
D125 младший байт	0F H	Функция	
D125 младший байт	05 H	Адрес данных	
D125 младший байт	00 H		
D126 младший байт	00 H	Размер данных (биты)	
D126 младший байт	10 H		
D126 младший байт	02 H	Количество байт	
D126 младший байт	34 H	Содержимое данных 1	Содержимое D0: H34
D126 младший байт	12 H	Содержимое данных 2	Содержимое D1: H12
D126 младший байт	21 H	CRC CHK младш.	
D126 младший байт	ED H	CRC CHK старш.	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные	Описание
D107 младший байт	01 H	Адрес
D107 младший байт	0F H	Функция
D107 младший байт	05 H	Адрес данных
D107 младший байт	00 H	
D107 младший байт	00 H	Размер данных (биты)
D107 младший байт	10H	
D107 младший байт	54H	CRC CHK младш.
D107 младший байт	CB H	CRC CHK старш.

Пример программы **10: COM1 (RS-232) / COM3 (RS-485),**
функциональный код **H0F**

Функциональный код K15 (H0F): чтение нескольких битов. Может быть прочитано до 64 битов.

1. ПЛК1 соединяется с ПЛК2: (M1143 выключен, ASCII-режим), (M1143 включен, RTU-режим)
2. PLC COM1/COM3 не обрабатывает принимаемые данные.
3. Рассмотрим связь между ПЛК1 (COM3) и ПЛК2 (COM1), таблица показывает состояния, когда ПЛК1 ВКЛЮЧАЕТ ПЛК2 Y0-Y17.

Установленное значение: K4Y0=1234H

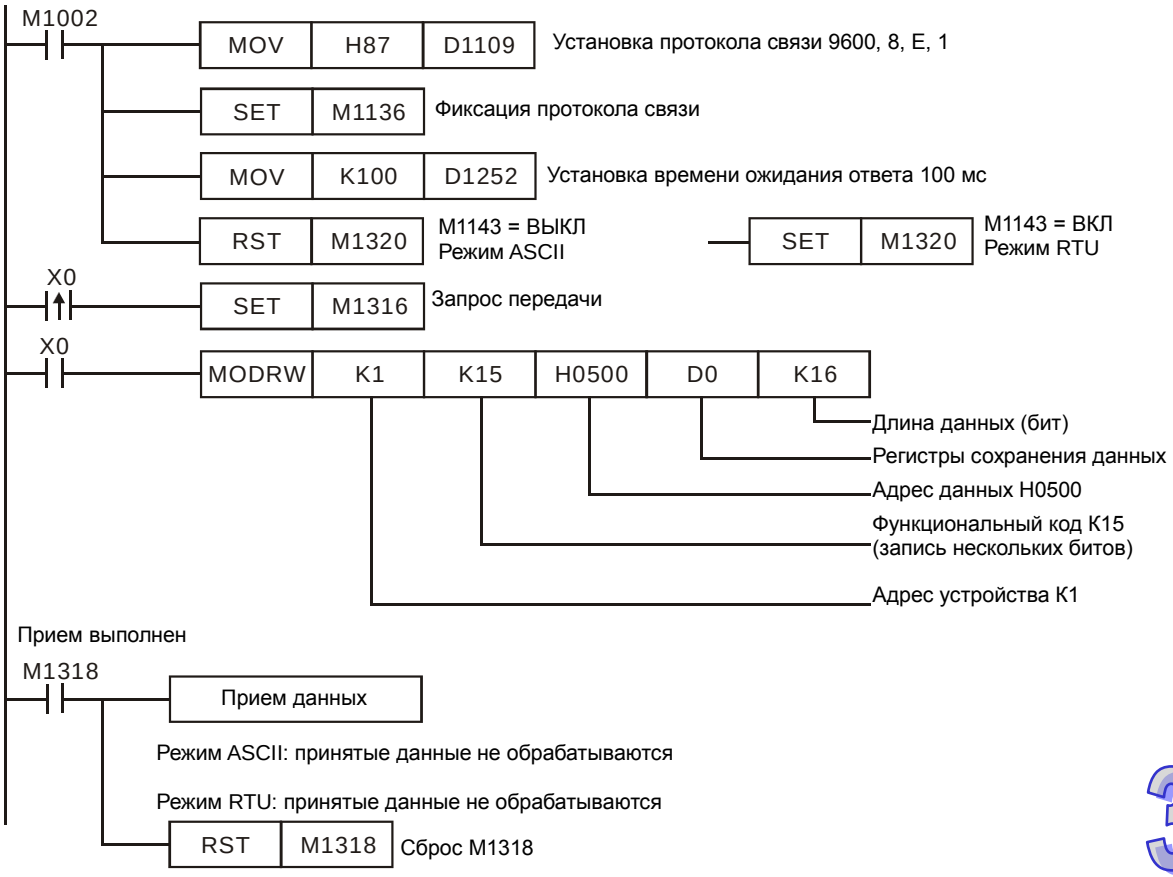
Устройство	Состояние	Устройство	Состояние	Устройство	Состояние	Устройство	Состояние
Y0	OFF	Y1	OFF	Y2	ON	Y3	OFF
Y4	ON	Y5	ON	Y6	OFF	Y7	OFF
Y10	OFF	Y11	ON	Y12	OFF	Y13	OFF
Y14	ON	Y15	OFF	Y16	OFF	Y17	OFF

Если ПЛК1 применяет COM1 для связи, см. программу ниже (изменения):

1. D1109→D1036: протокол связи
2. M1136→M1138: сохранение параметров связи.
3. D1252→D1249: установка задержки получения данных
4. M1320→M1139: ASCII/RTU – выбор режима
5. M1316→M1312: запрос передачи данных
6. M1318→M1314: флаги завершения приема

ASCII-режим (COM3: M1320 выключен, COM1: M1139 выключен):
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: " **01 0F 0500 0010 02 3412 93** "
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: " **01 0F 0500 0010 DB** "
(Принимаемые данные не обрабатываются)

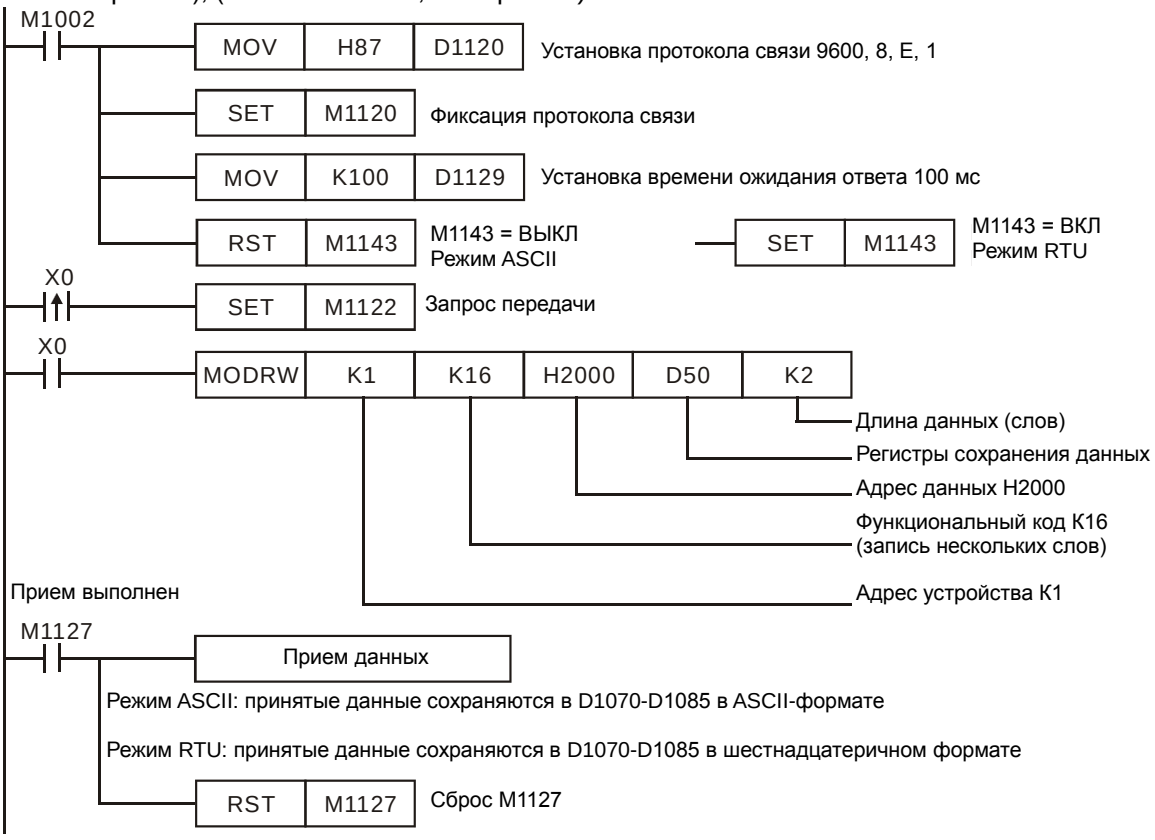
RTU-режим (COM3: M1320 включен, COM1: M1139 включен):
Когда X0 включен, ПЛК1 ⇨ ПЛК2, ПЛК1 передает: "**01 0F 0500 0010 02 34 12 21 ED**"
ПЛК2 ⇨ ПЛК1, ПЛК1 принимает: "**01 0F 0500 0010 54 CB**" ,
(Принимаемые данные не обрабатываются)



Пример программы 11: COM2 (RS-485), функциональный код H10

Функциональный код K16 (H10): чтение нескольких слов. Может быть прочитано до 16 слов.

1. Для ПЛК (COM2) в ASCII-режиме: только 8 слов.
2. Для режимов ASCII или RTU, ПЛК (COM2) сохраняет передаваемые данные в D1256~D1295, а принимаемые - в D1070~D1085.
3. Рассмотрим связь между ПЛК (COM2) и VFD-B для примера: таблицы поясняют статус, когда ПЛК (COM2) читает несколько слов из VFD-B (M1320 выключен, ASCII-режим), (M1320 включен, RTU-режим).



ASCII-режим (M1143 выключен)
Когда X0 включен, ПЛК ⇨VFD-B, ПЛК передает: **“01 10 2000 0002 04 1770 0012 30”**
VFD⇨ПЛК, ПЛК принимает: **“01 10 2000 0002 CD”**

Регистры передаваемых данных (передаваемые сообщения)

Регистр	Данные		Описание	
D1256 младший байт	‘0’	30 H	ADR 1 ADR 0	Адрес VFD: ADR (1,0)
D1256 старший байт	‘1’	31 H		
D1257 младший байт	‘1’	31 H	CMD 1 CMD 0	Командный код: CMD (1,0)
D1257 старший байт	‘0’	30 H		
D1258 младший байт	‘2’	32 H	Адрес данных	
D1258 старший байт	‘0’	30 H		
D1259 младший байт	‘0’	30 H		
D1259 старший байт	‘0’	30 H		
D1260 младший байт	‘0’	30 H	Количество регистров	
D1260 старший байт	‘0’	30 H		
D1261 младший байт	‘0’	30 H		
D1261 старший байт	‘2’	32 H		
D1262 младший байт	‘0’	30 H	Количество байт	
D1262 старший байт	‘4’	34 H		
D1263 младший байт	‘1’	31 H	Содержимое данных 1	Содержимое регистра D50: H1770(K6000)
D1263 старший байт	‘7’	37 H		
D1264 младший байт	‘7’	37 H		
D1264 старший байт	‘0’	30 H		
D1265 младший байт	‘0’	30 H	Содержимое данных 2	Содержимое регистра D51: H0012(K18)
D1265 старший байт	‘0’	30 H		
D1266 младший байт	‘1’	31 H		
D1266 старший байт	‘2’	32 H		
D1267 младший байт	‘3’	33 H	LRC CHK 1 LRC CHK 0	LRC CHK (0,1) ошибка проверки
D1267 старший байт	‘0’	30 H		

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные		Описание
D1070 младший байт	‘0’	30 H	ADR 1 ADR 0
D1070 старший байт	‘1’	31 H	
D1071 младший байт	‘1’	31 H	CMD 1 CMD 0
D1071 старший байт	‘0’	30 H	
D1072 младший байт	‘2’	32 H	Адрес данных
D1072 старший байт	‘0’	30 H	
D1073 младший байт	‘0’	30 H	
D1073 старший байт	‘0’	30 H	
D1074 младший байт	‘0’	30 H	Количество регистров
D1074 старший байт	‘0’	30 H	
D1075 младший байт	‘0’	30 H	
D1075 старший байт	‘2’	32 H	
D1076 младший байт	‘C’	43 H	LRC CHK 1
D1076 старший байт	‘D’	44 H	LRC CHK 0

RTU-режим (M1143 включен)
Когда X0 включен, ПЛК ⇨VFD-B, ПЛК передает: **“01 10 2000 0002 04 1770 0012 EE 0C”**
VFD⇨ПЛК, ПЛК принимает: **”01 10 2000 0002 4A08”**

Регистры передаваемых данных (передаваемые сообщения)

Регистр	Данные	Описание	
D1256 младший байт	01 H	Адрес	
D1257 младший байт	10 H	Функция	
D1258 младший байт	20 H	Адрес данных	
D1259 младший байт	00 H		
D1260 младший байт	00 H	Количество регистров	
D1261 младший байт	02 H		
D1262 младший байт	04 H	Количество байт	
D1263 младший байт	17 H	Содержимое данных 1	Содержимое D50: H1770 (K6000)
D1264 младший байт	70 H		
D1265 младший байт	00 H	Содержимое данных 2	Содержимое D51: H0012 (K18)
D1266 младший байт	12 H		
D1262 младший байт	EE H	CRC CHK младш.	
D1263 младший байт	0C H	CRC CHK старш.	

Регистры принимаемых данных (принимаемые сообщения)

Регистр	Данные	Описание
D1070 младший байт	01 H	Адрес
D1071 младший байт	10 H	Функция
D1072 младший байт	20 H	Адрес данных
D1073 младший байт	00 H	
D1074 младший байт	00 H	Количество регистров
D1075 младший байт	02 H	
D1076 младший байт	4A H	CRC CHK младш.
D1077 младший байт	08 H	CRC CHK старш.

Пример программы 12: COM1 (RS-232) / COM3 (RS-485), функциональный код H10
Функциональный код K16 (H10): чтение нескольких слов. Может быть прочитано до 16 слов.

1. Для ПЛК (COM2) в ASCII-режиме: только 8 слов.
2. ПЛК (COM1/COM3) не обрабатывает принимаемые данные.
3. Рассмотрим связь между ПЛК (COM3) и VFD-B для примера: таблицы поясняют статус, когда ПЛК (COM3) читает несколько слов из VFD-B (M1320 выключен, ASCII-режим), (M1320 включен, RTU-режим).

   Если ПЛК1 применяет COM1 для связи, см. программу ниже (изменения):
   1. D1109→D1036: протокол связи
   2. M1136→M1138: сохранение параметров связи.
   3. D1252→D1249: установка задержки получения данных
   4. M1320→M1139: ASCII/RTU – выбор режима
   5. M1316→M1312: запрос передачи данных
   6. M1318→M1314: флаги завершения приема



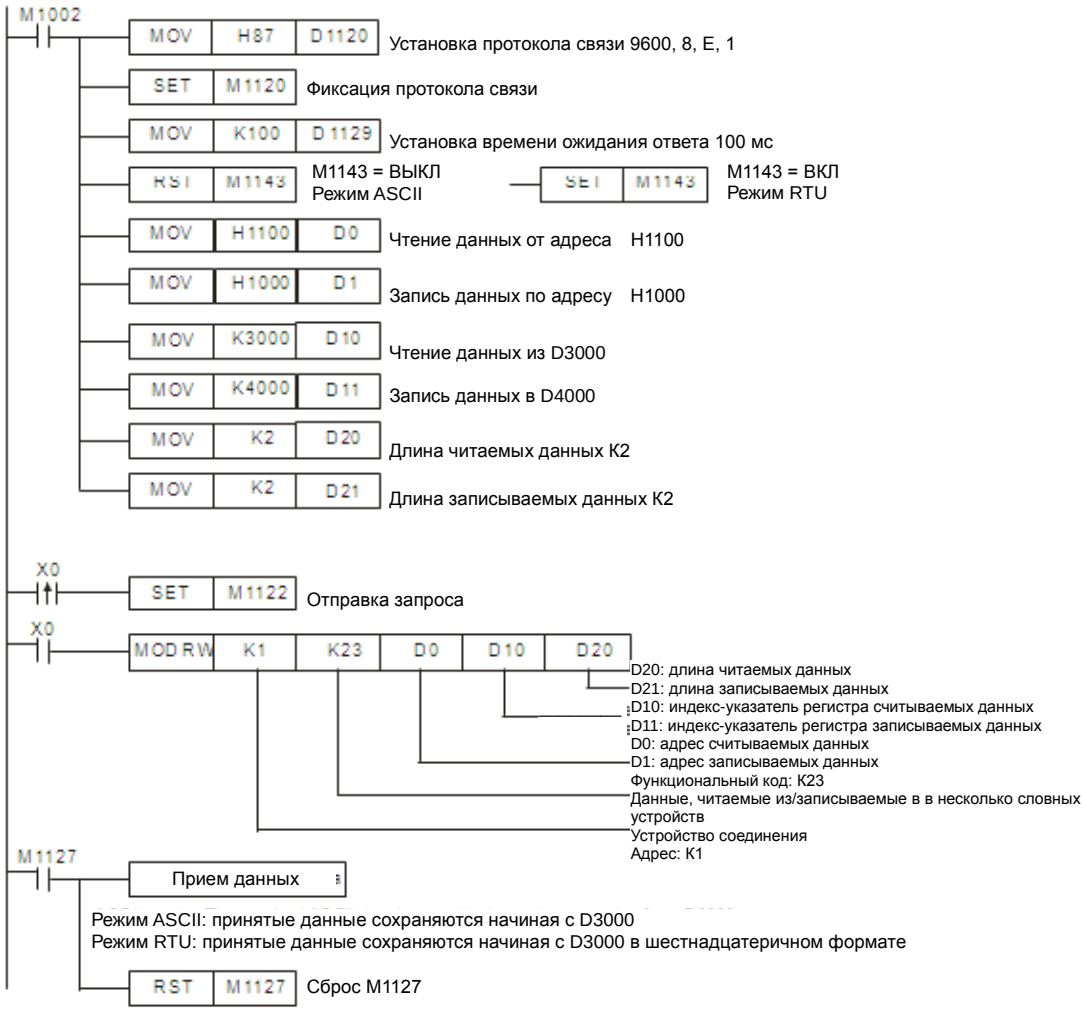
ASCII-режим (COM3: M1320 выключен, COM1: M1139 выключен):
Когда X0 включен, ПЛК ⇨VFD-B, ПЛК передает: **“01 10 2000 0002 04 1770 0012 30”**
VFD-B ⇨ПЛК, ПЛК принимает: **“01 10 2000 0002 CD”**
(Принимаемые данные не обрабатываются)

RTU-режим (COM3: M1320 включен, COM1: M1139 включен):
Когда X0 включен, ПЛК ⇨VFD-B, ПЛК передает: **“01 10 2000 0002 04 1770 0012 EE 0C”**
VFD-B ⇨ПЛК, ПЛК принимает: **”01 10 2000 0002 4A08”**
(Принимаемые данные не обрабатываются)

Пример программы 13: COM2 (RS-485), функциональный код H17.

1. Функциональный код K23 (H17): Данные считываются с нескольких словных устройств и записываются на несколько словных устройств. Максимальное количество таких устройств 16.
2. В режимах ASCII и RTU полученные данные сохраняются в регистрах, начиная с **S**.

3. Соединение между PLC-A (PLC COM2) и PLC-B:
4. Данные считываются из нескольких словных устройств из PLC-B в PLC-A и записываются в несколько словных устройств PLC-B от PLC-A. (M1143 выключен, ASCII режим; M1143 включен, RTU режим)



ASCII режим (M1143 выключен)
Когда X0 включен, **PLC-A** ⇨ **PLC-B**, PLC-A передает: **“01 17 1100 0002 1000 0002 04 1770 0012 06”**
PLC-B ⇨ **PLC-A**, PLC-A принимает: **“01 17 04 0100 1766 66”**

Регистры принятых данных PLC-A (ответные сообщения):

Регистр	Данные		Описание
D3000 младший байт	‘0’	30 H	ADR 1
D3000 старший байт	‘1’	31 H	ADR 0
D3001 младший байт	‘1’	31 H	CMD 1
D3001 старший байт	‘7’	37 H	CMD 0
D3002 младший байт	‘0’	30 H	Количество данных (байт)
D3002 старший байт	‘4’	34 H	
D3003 младший байт	‘0’	30 H	Содержимое адреса 1100H
D3003 старший байт	‘1’	31 H	
D3004 младший байт	‘0’	30 H	
D3004 старший байт	‘0’	30 H	
D3005 младший байт	‘1’	31 H	Содержимое адреса 1101H
D3005 старший байт	‘7’	37 H	
D3006 младший байт	‘6’	36 H	
D3006 старший байт	‘6’	36 H	
D3007 младший байт	‘6’	36 H	
D3007 старший байт	‘6’	36 H	LRC CHK 1 LRC CHK 0

RTU режим (M1143 включен)
Когда X0 включен, **PLC-A** ⇨ **PLC-B**, PLC-A передает: **“01 17 1100 0002 1000 0002 04 1770 0012 A702”**
PLC-B ⇨ **PLC-A**, PLC-A принимает: “**01 17 04 0100 1766 7701**”

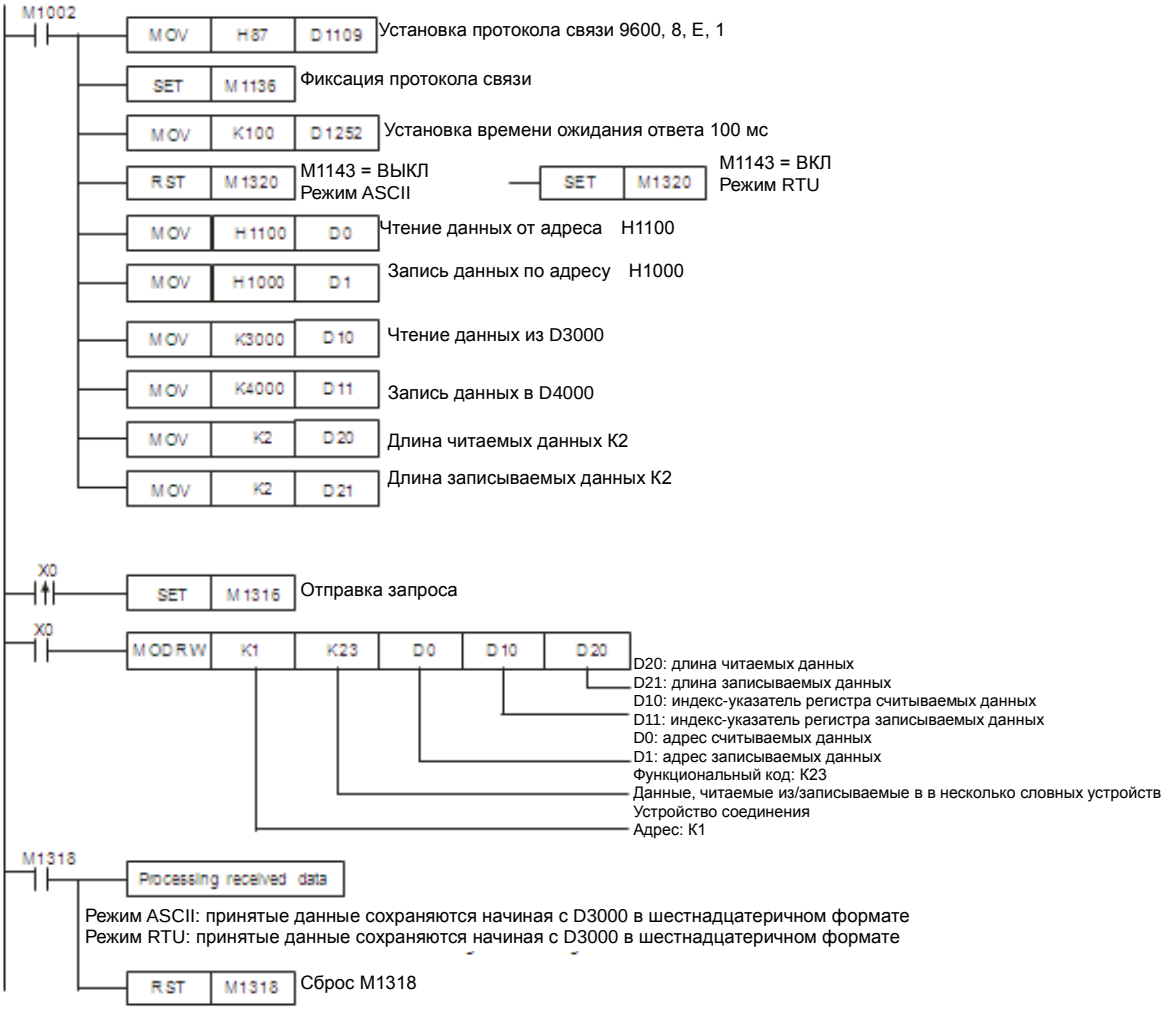
Регистры принятых данных PLC-A (ответные сообщения):

Регистр	Данные	Описание
D3000 младший байт	01 H	Адрес
D3001 младший байт	17 H	Функция
D3002 младший байт	04 H	Количество данных (байт)
D3003 младший байт	01 H	Содержимое адреса 1100H
D3004 младший байт	00 H	
D3005 младший байт	17 H	Содержимое адреса 1101H
D3006 младший байт	66 H	
D3007 младший байт	77 H	CRC CHK младш.
D3008 младший байт	01 H	CRC CHK старш.

Пример программы 14: COM1 (RS-232) / COM3 (RS-485), функциональный код H17.

1. Функциональный код K23 (H17): Данные считываются с нескольких словных устройств и записываются на несколько словных устройств. Максимальное количество таких устройств 16.
2. В режимах ASCII и RTU полученные по COM1/COM3 данные сохраняются в регистрах, начиная с **S+1**. Используйте команду DTM для изменения или перемещения данных.
3. Связь между PLC-A (PLC COM3) и PLC-B:
   - Данные записываются в несколько словных устройств PLC-B от PLC-A. (M1320 выключен, ASCII режим; M1320 включен, RTU режим)
   - Если используется COM1 на PLC-A, программа меняется следующим образом:

   1. D1109→D1036: Протокол связи
   2. M1136→M1138: Сохранение настроек связи
   3. D1252→D1249: Задержка связи
   4. M1320→M1139: Выбор режима ASCII или RTU
   5. M1316→M1312: Отправка данных запроса на связь
   6. M1318→M1314: Получение данных завершено



ASCII режим (COM3: M1320 выключен; COM1: M1139 выключен):

Когда X0 включен, **PLC-A** ⇨ **PLC-B**, PLC-A передает: **“01 17 1100 0002 1000 0002 04 1770 0012 06”**

PLC-B ⇨ **PLC-A**, PLC-A принимает: **“01 17 04 0100 1766 66”**

Регистры принятых данных PLC-A (ответные сообщения):

Регистр	Данные	Описание
D3000	0100 H	PLC-A преобразует данные в 1100H и сохраняет их автоматически
D3001	1766 H	PLC-A преобразует данные в 1101H и сохраняет их автоматически

RTU режим (COM3: M1320 включен; COM1: M1139 включен):
Когда X0 включен, **PLC-A** ⇨ **PLC-B**, PLC-A передает: **"01 17 2100 0002 2000 0002 04 1770 0012 A702"**
PLC-B ⇨ **PLC-A**, PLC-A принимает: "**01 17 04 0100 1766 7701**"

Регистры принятых данных PLC-A (ответные сообщения):

Регистр	Данные	Описание
D3000	0100 H	PLC-A преобразует данные в 1100H и сохраняет их автоматически
D3001	1766 H	PLC-A преобразует данные в 1101H и сохраняет их автоматически

API	Команда	Операнды	Функция	Контроллеры	
				EH2/EH3	SV/SV2
151	PWD	(S) (D)	Определение длительности входного импульса		

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*															PWD: 5 шагов
D													*			

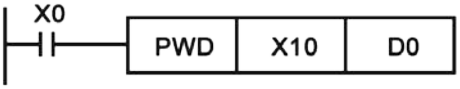
ИМПУЛЬС		16-бит		32-бит	
EH2/EH3	SV/SV2	EH2/EH3	SV/SV2	EH2/EH3	SV/SV2

Операнды:
S: Источник (вход ПЛК) **D**: Регистр для записи длительности входного импульса

Описание (команда используется только моделями ПЛК DVP-EH2/SV/SV2 и DVP-EH3 с прошивкой версии 1.40 и выше):
1. Диапазон **S**: X10 ~ X17; Диапазон **D**: D0 ~ D999, занимает 2 последовательных регистра. Максимальное время 21 474.83647 сек., т.е. 357.9139 мин. или 5.9652 ч.
2. Команда PWD определяет длительность входного сигнала на входах X10 ~ X17; частотный диапазон 1 Гц ~1 кГц. При выключенном M1169 команда определяет длительность по переднему и заднему фронтам входного сигнала (ед. изм.: 100 мкс). При включенном M1169 команда определяет длительность по переднему фронту 2 последовательных входных сигналов (ед. изм.: 1 мкс). Команду PWD нельзя применять одновременно к входам X10 ~ X17 (как команды DCNT и ZRN).
3. Временного ограничения на работу данной команды нет, но одновременно может быть запущена только одна команда PWD.

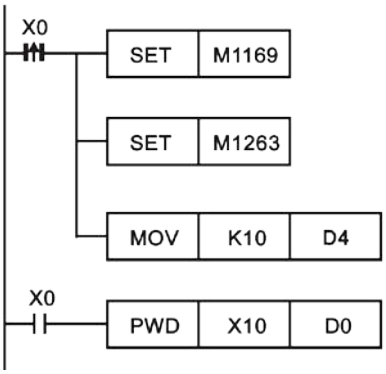
Описание (команда используется только ПЛК DVP-EH3 с прошивкой версии 1.60 и выше):
1. Диапазон **S**: X10, X11, X14, X15 (не могут использоваться повторно)
2. Диапазон **D**: Результат, число входных импульсов и время сохраняются в 5 последовательных регистрах. **D** может находиться в диапазоне D0…D999.
3. При первоначальном выполнении команды режим обнаружения зависит от состояния M1169. После выполнения команды режим обнаружения не может быть изменен.
4. Если M1169 включен, M1154 – флаг определения длительности импульса в режимах «импульс ON/OFF». При выключенном M1154 определяется длительность в режиме «импульс OFF». При включенном M1154 определяется длительность в режиме «импульс ON». При повторном выполнении команды состояние M1154 сохраняется. Состояние M1154 может быть изменено после выполнения команды.
5. Если M1169 включен, M1263 – флаг усреднения. При включенном M1263 частота входного сигнала усредняется и сохраняется число импульсов. При повторном выполнении команды состояние M1263 сохраняется. Состояние M1263 может быть изменено после выполнения команды.
6. Значение, сохраняемое в (**D**+1, **D**) зависит от режима обнаружения. Если определяется длительность импульса в режимах «импульс ON/OFF», в (**D**+1, **D**) сохраняется время (длительность). Максимальное время 21 474.83647 сек., т.е. 357.9139 мин. или 5.9652 ч. Если определяется частота входного сигнала, в (**D** +1, **D**) сохраняется частота. **D**+2, **D**+3, **D**+4 используются только при включенном усреднении частоты входного сигнала. В (**D**+3, **D**+2) сохраняется число входных импульсов. В **D**+4 сохраняется усредненное значение из диапазона K1…K20. Если значение в **D**+4 выше, чем K20, берется значение K20. Если значение в **D**+4 меньше, чем K1, берется значение K1.
7. Команда может быть выполнена не более трех раз в программе.

Пример программы (команда используется только моделями ПЛК DVP-EH2/SV/SV2 и DVP-EH3 с прошивкой версии 1.40 и выше)



При включенном X0 происходит запись длительности сигнала входа X10, и результат сохраняется в D1 и D0.

Пример программы (команда используется только ПЛК DVP-EH3 с прошивкой версии 1.60 и выше):



При включенном X0 частота входных импульсов от входа X10 усредняется K10, усредненная частота сохраняется в (D1, D0), число импульсов сохраняется в (D3, D2).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
154	D	RAND	P	S_1 S_2 D	Генератор случайных чисел				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	RAND, RANDP: 7 шагов DRAND, DRANDP: 13 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

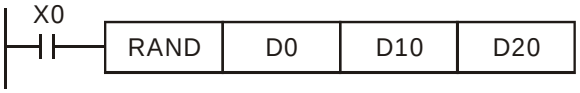
S_1: Минимальное значение случайного числа **S_2**: Максимальное значение случайного числа **D**: Результат

Описание:

Диапазон 16-битных операндов **S_1**, **S_2**: K0≦**S_1** , **S_2**≦K32,767; диапазон 32-битных операндов **S_1**, **S_2**: K0≦**S_1** , **S_2**≦K2,147,483,647.

При **S_1** > **S_2** возникает ошибка. Команда прекращает выполнение, включаются флаги M1067, M1068 и в регистр D1067 записывается код ошибки 0E1A (в шестнадцатеричном формате).

Пример программы:



При включенном X10 команда RAND генерирует случайное число в диапазоне между мин. значением D0 и макс. значением D10. Результат сохраняется в D20.

API	Команда			Операнды	Функция	Контроллеры				
						SA2/SE	SX2/SC	EH2/EH3	SV/SV2	ES2/EX2
155	D	ABSR		S D_1 D_2	Чтение абсолютной текущей позиции					

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S	*	*	*	*												DABSR: 13 шагов
D_1		*	*	*												
D_2								*	*	*	*	*	*	*		

ИМПУЛЬС						16-бит						32-бит					
ES2/EX2	SS2	SA2/SE	SX2/SC	EH2/EH3	SV/SV2	ES2/EX2	SS2	SA2/SE	SX2/SC	EH2/EH3	SV/SV2	ES2/EX2	SS2	SA2/SE	SX2/SC	EH2/EH3	SV/SV2

Операнды:

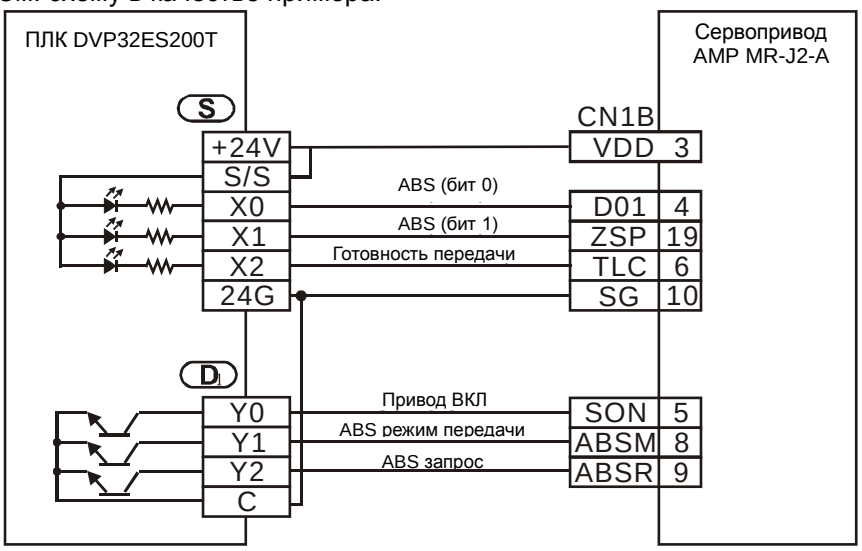
S: Входной сигнал от сервопривода (занимает 3 последовательных адреса)

D_1: Управляющий сигнал на сервопривод (занимает не более 3 последовательных адресов)

D_2: Значение абсолютной позиции (32-бит), считанное с сервопривода (занимает не более 4 последовательных адресов).

Описание (команда используется только моделями ПЛК DVP-ES2/EX2 версии 3.00 / DVP-SA2 версии 2.40 / DVP-SE версии 1.20 / DVP-SX2 версии 2.20 / DVP-SS2):

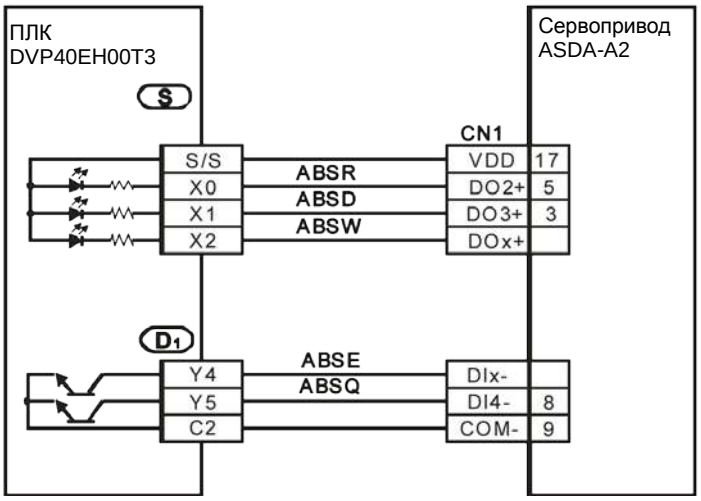
1. Эта команда считывает абсолютную позицию (ABS) с функцией проверки абсолютной позиции сервоприводов MITSUBISHI MR-J2.
2. Команда ABSR может быть только 32-битной (т.е., всегда DABSR) и может быть использована только один раз в программе.
3. **S** занимает 3 последовательных адреса **S**, **S** +1, **S** +2. **S** и **S** +1 связаны с ABS (бит 0, бит 1) сервопривода для передачи данных. **S** +2 связан с сервоприводом для индикации готовности передачи данных.
4. **D_1** занимает 3 последовательных адреса **D_1**, **D_1**+1, **D_1**+2. **D_1** связан с включением сервопривода (servo ON (SON), **D_1**+1 связан с ABS режимом передачи привода и **D_1**+2 связан с запросом ABS.
   См. схему в качестве примера:



5. **D_2** занимает 2 последовательных адреса **D_2**, **D_2**+1. **D_2** – младшее слово, **D_2**+1 - старшее. Данные о позиции записываются в регистрах (D1337, D1336) импульсного выхода CH0 (Y0, Y1) и регистры (D1339, D1338) импульсного выхода CH1 (Y2, Y3) – для DVP-EH, поэтому предлагается изначально назначить эти регистры. Для DVP-SC данные регистры будут: (D1348, D1349) импульсного выхода CH0 (Y10) и (D1350, D1351) импульсного выхода CH1 (Y11)
6. После выполнения команды DABSR включается флаг M1029. M1029 сбрасывается пользователем.
7. Используйте H/O контакты для запуска команды DABSR. Если контакт окажется выключенным во время выполнения команды, привод остановится и покажется ошибка.
8. Если контакт выключается после выполнения команды DABSR, сигнал servo ON (SON) сервопривода, определенный в **D_1** также отключится и привод остановится.

Описание (команда используется только моделями ПЛК DVP-ES2/EX2 версии 3.20 / DVP-SA2 версии 2.60 / DVP-SE версии 1.40 / DVP-SX2 версии 2.40):

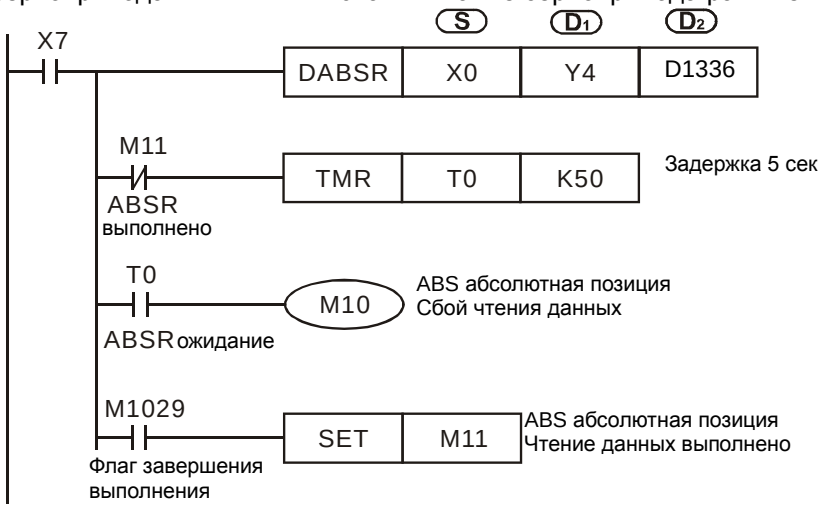
1. Эта команда считывает абсолютную позицию (ABS) с функцией проверки абсолютной позиции сервоприводов MITSUBISHI MR-J2 и абсолютную позицию (ABS) с функцией проверки абсолютной позиции сервоприводов Delta ASDA-A2 (прошивка 1.045 sub12 и выше).
2. Выбор сервопривода определяется состоянием флага M1177. При выключенном M1177 применяется привод MITSUBISHI MR-J2. При включенном M1177 – привод Delta ASDA-A2.
3. **S** занимает 3 последовательных адреса. **S**, **S** +1 и **S** +2 подсоединены соответственно к ABSR, ABSD, ABSW на сервоприводе. См. пример соединения ниже.
4. **D_1** занимает 2 последовательных адреса, **D_1**, и **D_1** + 1. **D_1** соединен с ABSE. **D_1**+1 соединен с ABSQ на сервоприводе. См. пример соединения ниже.
5. **D_2** занимает 4 последовательных адреса **D_2**, **D_2** +1. **D_2** +2 и **D_2** +3. Состояние системы абсолютных координат (P0-50) сохраняются в **D_2**, позиция абсолютного энкодера (многооборотная) (P0-51) сохраняется в **D_2** +1. Младшие 16 бит абсолютной позиции энкодера (импульсы/оборот или PUU) (P0-52) сохраняются в **D_2** +2. Старшие 16 бит абсолютной позиции энкодера (импульсы/оборот или PUU) (P0-52) сохраняются в **D_2** +3.
6. После чтения абсолютной позиции командой DABSR включается M1580. При возникновении во время чтения ошибки включается M1581.



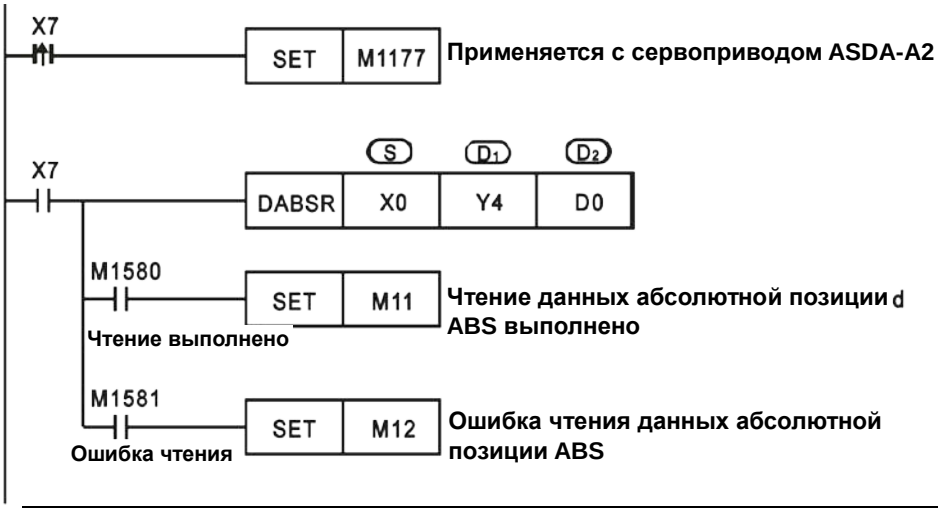
7. **D2** занимает 4 последовательных адреса **D2**, **D2** +1. **D2** +2 и **D2** +3. Состояние системы абсолютных координат (P0-50) сохраняются в **D2**, позиция абсолютного энкодера (многооборотный) (P0-51) сохраняется в **D2** +1. Младшие 16 бит абсолютной позиции энкодера (импульсы/оборот или PUU) (P0-52) сохраняются в **D2** +2. Старшие 16 бит абсолютной позиции энкодера (импульсы/оборот или PUU) (P0-52) сохраняются в **D2** +3.
8. После чтения абсолютной позиции командой DABSR включается M1580. При возникновении во время чтения ошибки включается M1581.
9. При работе команды DABSR укажите H/O контакт. Если контакт команды DABSR выключается во время начала работы команды, чтение текущего абсолютного значения будет прервано из-за некорректности данных.
10. Если входные сигналы X0~X17, команда выполняется 2 сек., если входные сигналы от X20, команда выполняется 3 сек. Время зависит от времени цикла сканирования.

Пример программы (команда используется только моделями ПЛК DVP-ES2/EX2 версии 3.00 / DVP-SA2 версии 2.40 / DVP-SE версии 1.20 / DVP-SX2 версии 2.20 / DVP-SS2):

1. При включенном X7 32-битные данные абсолютной позиции, считанные с сервопривода, сохраняются в регистрах (D1337, D1336) CH0 (серия EH). В это же время таймер T10 включает отсчет 5 сек. Если чтение абсолютной позиции не выполнено за 5 сек, включается M10, показывающий ошибку чтения.
2. При запуске системы необходимо синхронизировать входную мощность ПЛК DVP-PLC и сервопривода AMP или включать питание сервопривода раньше питания ПЛК.



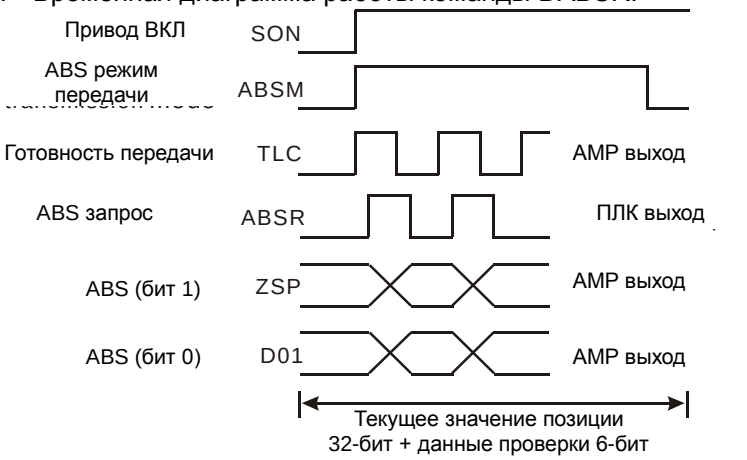
Пример программы (команда используется только моделями ПЛК DVP-ES2/EX2 версии 3.20 / DVP-SA2 версии 2.60 / DVP-SE версии 1.40 / DVP-SX2 версии 2.40):



1. При включенном X7 абсолютная позиция, считываемая с сервопривода Delta ASDA-A2, сохраняется в регистрах D0~D3. Состояние флагов M1580 и M1581 показывает успешность проведения операции чтения данных.

Заметки:

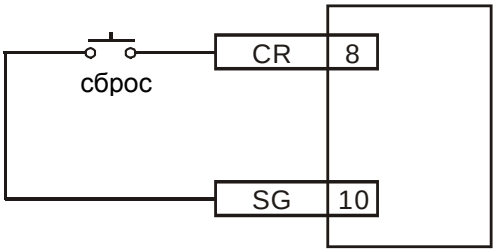
1. Временная диаграмма работы команды DABSR:



2. При выполнении команды DABSR, включение привода и режим передачи данных ABS приводятся для выхода.
3. Для сигналов “готовность передачи” и “запрос ABS”, пользователи могут подтвердить передачу и получение состояния обеих сторон, а также обработать 32-битные данные о ABS положении и 6-битные данные проверки.
4. Данные по передаче ABS (бит 0, бит1).
5. Эта команда применима только для сервопривода с функцией проверки абсолютного положения, т.е. MITSUBISHI MR-J2-A.
6. Выберите один из следующих методов для первоначальной команды ABSR:

- Выполнение команды API 156 ZRN с функцией возврата в нулевую точку.
- Применение JOG-функции или ручной настройкой возврата в нулевую точку привода (сброса входного сигнала на привод). См. диаграмму ниже для осуществления метода сброса сигнала. См. также описание команды API 159 DRVA для подробных сведений о соединении ПЛК DVP и привода MITSUBISHI MR-J2-A.

Пример: Mitsubishi MR-J2-A



API	Команда			Операнды	Функция	Контроллеры			
156	D	ZRN		S_1 S_2 S_3 D	Возврат в исходную позицию	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DZRN: 17 шагов
S_1					*	*	*	*	*	*	*	*	*	*		
S_2					*	*	*	*	*	*	*	*	*	*		
S_3	*															
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Заданная частота для возврата в исходную позицию (макс. 100 кГц) **S_2**: Пониженная частота (выключатель DOG) **S_3**: Операнд выключателя DOG **D**: Импульсный выход

Описание:

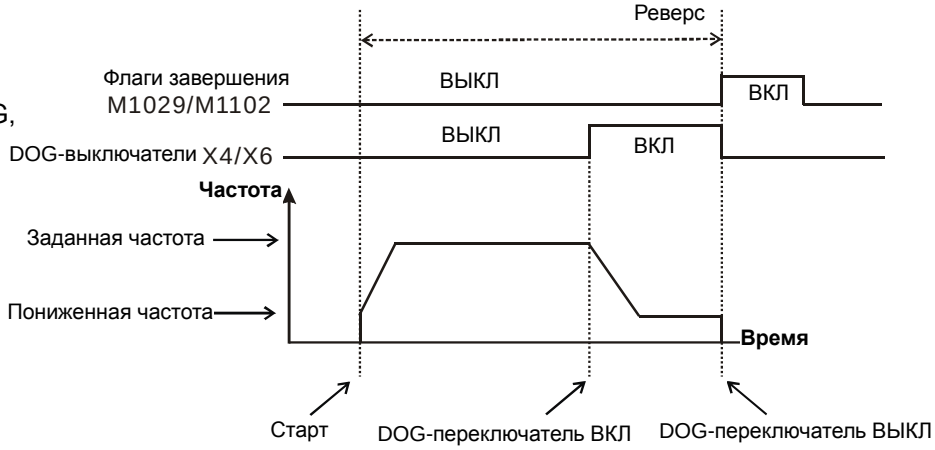
1. **S_2** должна быть меньше **S_1**. Пониженная частота (переключатель DOG) также является начальной частотой.
2. Операнды **S_3** и **D**, которые должны быть использованы как входы/выходы, устанавливаются в соответствии с таблицей ниже, т.е. когда **S_3** определен как X4, **D** - как Y0; а если **S_3** определен как X6, **D** - как Y2.

3. M1307 включает/отключает ограничение перемещения влево для CH0 (Y0, Y1) и CH1 (Y2, Y3). M1307 должен быть установлен перед запуском команды. M1305 и M1306 могут менять направление выходных импульсов на Y1 и Y3, они также должны устанавливаться до начала выполнения команды. Ограничитель перемещения влево для CH0 (Y0, Y1) - X5; для CH1 (Y2, Y3) - X7. Функции, входы и выходы:

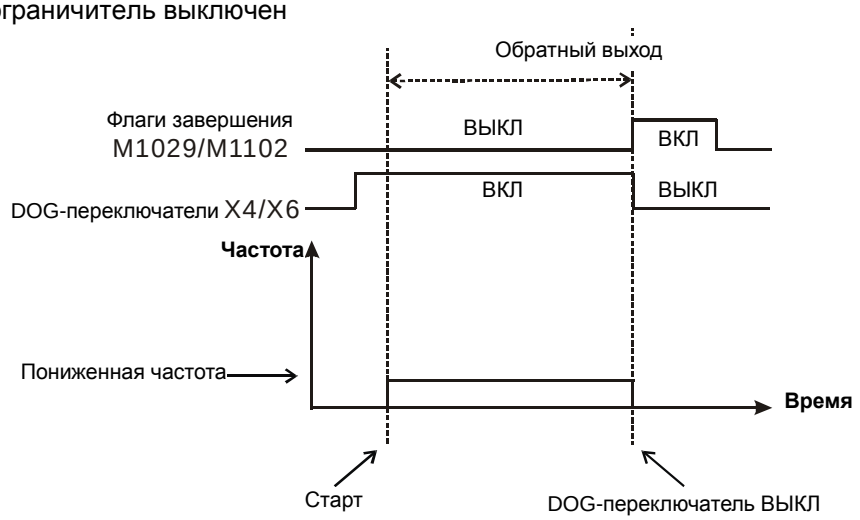
Канал / Вход	CH0(Y0,Y1)	CH1(Y2,Y3)	Примечание
Точка DOG	X4	X6	
Ограничитель перемещения влево (M1307 включен)	X5	X7	
Левый концевой выключатель срабатывает либо по переднему, либо по заднему фронту сигнала. (ВЫКЛ: передний фронт; ВКЛ: задний фронт)	M1585	M1585	ES2/EX2/ES2-C V3.20 SA2 V2.80 SX2 V2.60 SS2 V3.0 SEV1.4
Реверс	M1305	M1306	
Выбор нулевой точки	M1106	M1107	см. п. 7.
M1346 включен Сигнал сброса	Y4	Y5	см. п. 8
D1312 != 0	M1308 выключен (достижение сигнала Z-фазы)		см. п. 9
	X2	X3	
D1312 != 0	M1308 включен (вывод назначенного числа импульсов)		см. п. 10

4. Когда **D** определен как Y0, его управляющий сигнал Y1; когда **D** определен как Y2, его управляющий сигнал Y3.
5. При окончании выдачи импульсов включаются флаги завершения M1029 (CH0), M1102 (CH1) и регистр указания текущего положения сбрасывается до 0.
6. При выполнении команды DZRN внешние прерывания I400/I401 (X4) или I600/I601 (X6) будут отключены до завершения работы команды. Кроме того, если ограничитель перемещения влево (X5/X7) включен во время выполнения команды, внешние прерывания I500/I501 (X5) или I700/I701 (X7) также будут отключены.
7. Выбор нулевой точки: по умолчанию положение нулевой точки слева от выключателя DOG (вход вкл→выкл) (1 режим). Если пользователю необходимо изменить положение нулевой точки, чтобы она была справа от DOG, включается M1106(CH0) или M1107(CH1) перед выполнением команды DZRN (Функция поддерживается ES2/EX2, V1.20 или выше).
8. Старт функции очистки импульсов на выходе. Когда выключатель DOG останавливается, выводится другой импульс (шириной 20 мс). Когда импульсы прекращаются, включается флаг выполнения. См. временную диаграмму 4 (Функция поддерживается ES2/EX2, V1.20 или выше).
9. Когда значение в D1312 не установлено в 0 и M1308 выключен, запускается функция достижения Z-фазы. Когда значение D1312 положительно (макс. 10), идет поиск сигнала Z-фазы в положительном направлении. Когда значение D1312 отрицательно (мин. -10) – в отрицательном. См. временную диаграмму 5 (функция поддерживается ES2/EX2 V1.20 или выше и SS2/SX2 V1.20 или выше).
10. Когда D1312 не установлен в 0 и M1308 включен, запускается функция вывода заданного числа импульсов. Когда значение D1312 положительное (макс. 30000), это показывает положительное направление выходных импульсов. Когда значение D1312 отрицательно (мин. -30000), это показывает отрицательное направление. См. временную диаграмму 6. (функция поддерживается ES2/EX2 V1.40 или выше и SS2/SX2 V1.40 или выше).

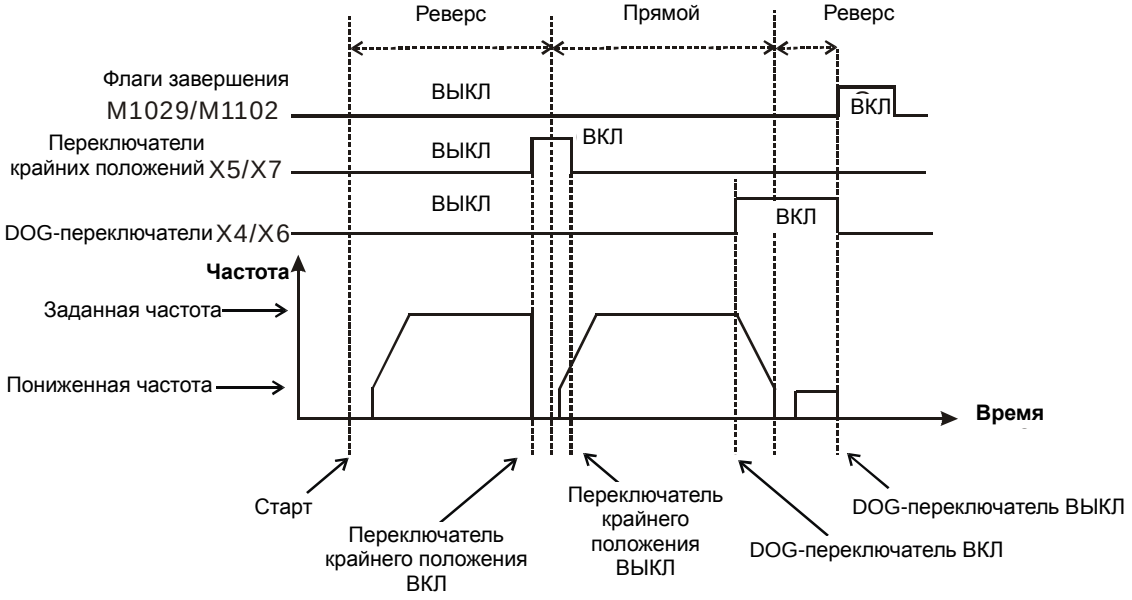
Состояние 1:
Текущее положение на правой стороне переключателя DOG, импульсный выход реверсивный, ограничитель выключен.



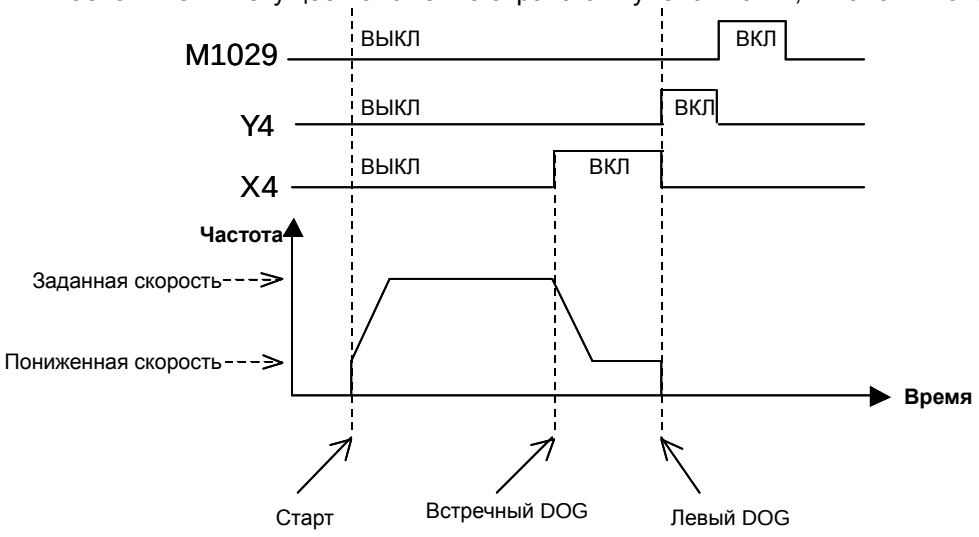
Состояние 2: Переключатель DOG включен, импульсный выход реверсивный, ограничитель выключен



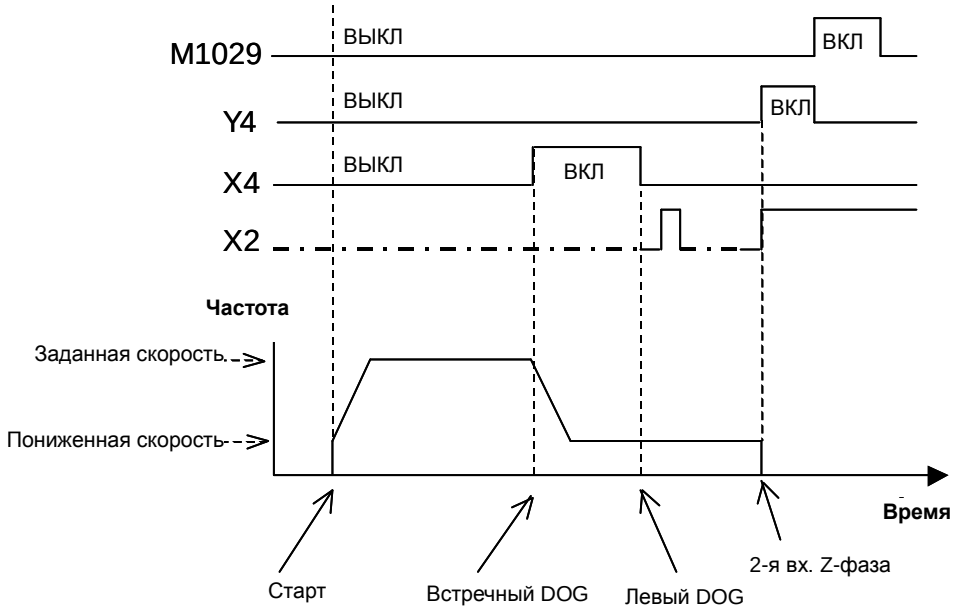
Состояние 3: Текущее положение на левой стороне переключателя DOG, импульсный выход реверсивный, ограничитель включен.



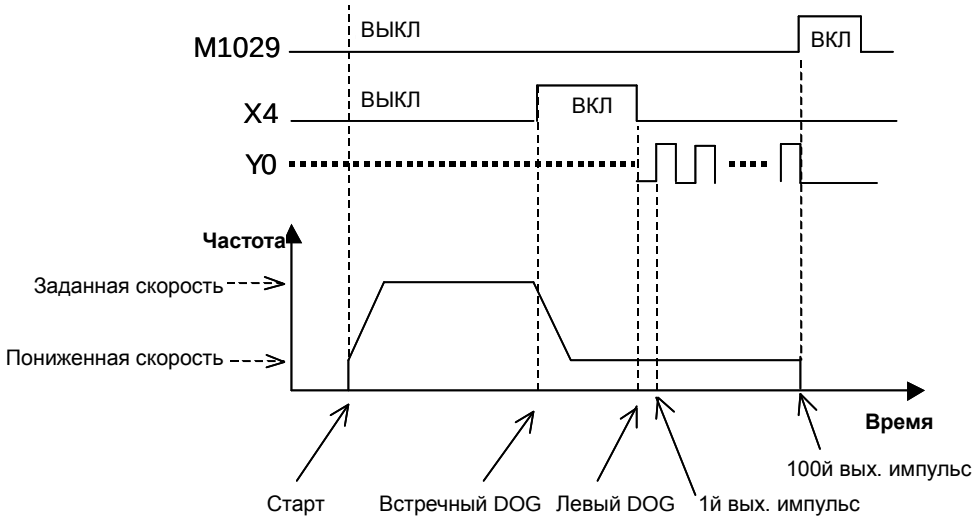
Состояние 4: Текущее положение справа от нулевой точки, M1346 включен.



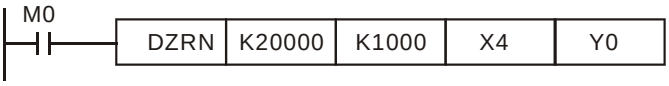
Состояние 5: Текущее положение справа от нулевой точки, D1312=-2, M1308 выключен, M1346 включен.



Состояние 6: Текущее положение справа от нулевой точки, D1312=-100, M1308 включен.

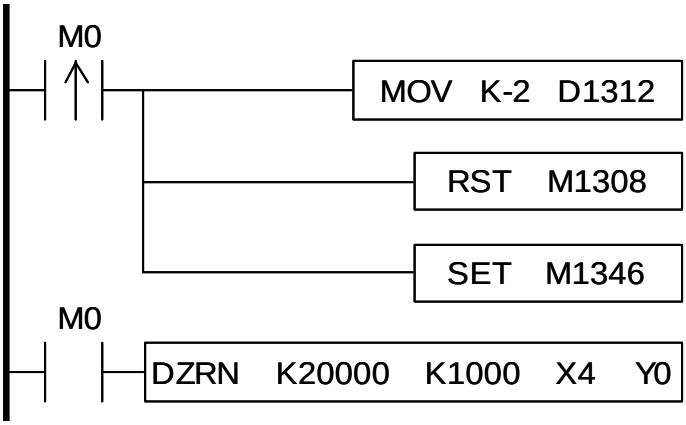


Пример программы 1:



Когда M0 включен, импульсный выход Y0 выполняет возвращение в нулевую точку с частотой 20 кГц. При достижении DOG X4 включается, и частота меняется на пониженную частоту, равную 1 кГц. Y0 останавливается, когда X4 выключается.

Пример программы 2:



Когда M0 включен, импульсный выход Y0 выполняет возвращение в нулевую точку с частотой 20 кГц. При достижении DOG X4 включается, и частота меняется на пониженную частоту, равную 1 кГц. Когда X4 выключен, он стремится к X2 (Z-фаза) импульсного выхода (триггерный сигнал по правому краю), и Y4 останавливается после импульса (шириной 20 мс) (M1029 включен).

Заметки: Флаги:

M1029	CH0 (Y0, Y1) обработка импульсного выхода завершена
M1102	Y2/CH1 (Y2, Y3) обработка импульсного выхода завершена
M1106:	Выбор нулевой точки. M1106 включен, изменение нулевой точки справа от DOG для возврата в нулевую точку для канала CH0
M1107:	Выбор нулевой точки. M1107 включен, изменение нулевой точки справа от DOG для возврата в нулевую точку для канала CH1
M1305:	Резервный импульсный выход Y1 для команд управления высокоскоростными импульсными выходами
M1306:	Резервный импульсный выход Y3 для команд управления высокоскоростными импульсными выходами
M1307:	Для команды ZRN: включение ограничителя левого положения
M1308:	Указанные выходные импульсы (D1312) или поиск сигнала Z-фазы при достижении нулевой точки
M1346:	Очистка выходного сигнала при завершении выполнения команды ZRN
	Специальные регистры D:
D1312:	Указанное число дополнительных импульсов и функция поиска Z-фазы для команды ZRN (используется с M1308)

API	Команда			Операнды	Функция
157	D	PLSV		(S) (D1) (D2)	Импульсный выход с заданием частоты и направления вращения серводвигателя

Контроллеры			
ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	PLSV: 7 шагов
S					*	*	*	*	*	*	*	*	*	*	*	DPLSV: 13 шагов
D_1		*														
D_2		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

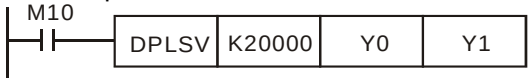
S: Частота выходных импульсов **D_1**: Импульсный выход ПЛК (Y0, Y2) **D_2**: Направление вращения (выходы CH0(Y1) и CH1(Y3))

Описание:

1. Команда поддерживает только импульсные выходы типа: Pulse/Dir.
2. **S** – назначенная частота импульсного выхода. Диапазон: -100,000Hz ~ +100,000 Hz. “+/-” указывает направление выходного сигнала (вперед/назад). Частота может быть изменена во время выдачи импульсов. Однако, если в результате этого меняется направление вращения, то выполнение команды приостановится на 1 цикл и возобновится с измененной частотой.
3. **D_1** – импульсный выход. Применяются только выходы CH0(Y0) и CH1(Y2).
4. **D_2** – направление выходного сигнала. Применяются только выходы CH0(Y1) и CH1(Y3). Значение **D_2** соотносится со знаком “+” или “-“ операнда **S**. Если **S** со знаком “+”, **D_2** выключен; если **S** со знаком “-“, **D_2** включен.
5. M1305 и M1306 могут менять направление CH0/CH1, заданное в **D_2**. При **S** со знаком “-“, **D_2** включен, но, если M1305/M1306 включаются до начала выполнения команды, **D_2** выключится во время выполнения команды.
6. Команда PLSV не поддерживает настройки рамповой функции. Если требуется применение этой функции, используйте команду API 67 RAMP.
7. Если время работы команды условия, запустившие команду, перестанут выполняться (например контакт выключится), то выдача импульсов немедленно прекратится.

Пример программы:

Когда M10 включен, Y0 выдает импульсы с частотой 20 кГц. Y1 выключен, показывая движение вперед.



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
158	D	DRVI	(S1) (S2) (D1) (D2)	Позиционирование в относительных координатах				

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S_1					*	*	*	*	*	*	*	*	*	*	*
S_2					*	*	*	*	*	*	*	*	*	*	*
D_1		*													
D_2		*	*	*											

Шаги программы
DDRVI: 17 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Число импульсов (относительное позиционирование) **S_2**: Частота выходных импульсов
D_1: Импульсный выход **D_2**: Направление вращения

Описание:

1. Команда поддерживает только импульсные выходы типа: Pulse/Dir.
2. **S_1** – число импульсов (относительное позиционирование). Диапазон: -2,147,483,648 ~ +2,147,483,647. “+/-” показывает прямое и обратное движение.
3. **S_2** – частота выходных импульсов. Диапазон: 6 ~ 100,000 Гц.
4. **D_1** – импульсный выход. Применяются только выходы CH0(Y0) и CH1(Y2).
5. **D_2** – направление выходного сигнала. Применяются только выходы CH0(Y1) и CH1(Y3).
6. Значение **D_2** соотносится со знаком “+” или “-“ операнда **S**. Если **S** со знаком “+”, **D_2** выключен; если **S** со знаком “-“, **D_2** включен. **D_2** не будет немедленно отключен после завершения передачи импульсов, отключение произойдет после выключения контакта привода.
7. Заданным значением **S_1** является относительное положение:

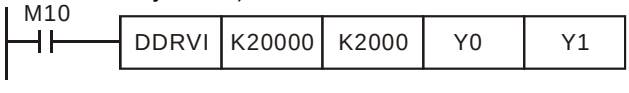
a. текущая позиция (32-битные данные) CH0 (Y0, Y1) хранится в D1031(старший байт), D1030 (младший байт)

b. текущая позиция (32-битные данные) CH1 (Y2, Y3) хранится в D1337(старший байт), D1336 (младший байт).

8. В режиме обратного вращения, значение в (D1031, D1330) и (D1336, D1337) уменьшается.
9. D1343 (D1353) – установка времени рамповой функции CH0 (CH1). Диапазон: 20 ~ 32,767 мс. По умолчанию: 100 мс. При выходе значения за границы диапазона, ПЛК применяет верхнее/нижнее допустимое значение соответственно.
10. D1340 (D1352) – настройка начальной/конечной частоты CH0 (CH1). Диапазон: 6 - 100,000 Гц. При выходе значения за границы диапазона, ПЛК применяет верхнее/нижнее допустимое значение соответственно.
11. M1305 и M1306 может менять направление выходных импульсов CH0/CH1, установленных в **D_2**. Когда **S** со знаком “-“, **D_2** включен, однако, если включаются M1305/M1306 до начала выполнения команды, **D_2** выключается во время выполнения команды.
12. Время рамповой функции CH0 и CH1 может быть изменено с помощью (M1534, D1348) и (M1535, D1349). Когда M1534 / M1535 включены, время рамповой функции CH0 / CH1 определяется D1348 / D1349.
13. Если M1078 / M1104 включены во время выполнения команды, на Y0 / Y2 приостанавливается выдача импульсов, а включенные M1538 / M1540 показывают паузу в работе. Когда M1078 / M1104 выключены, M1538 / M1540 также выключены, Y0 / Y2 выдают оставшиеся импульсы.
14. Команда DRVI поддерживает функций метки привода и маски. См. описание команды API 59 PLSR.

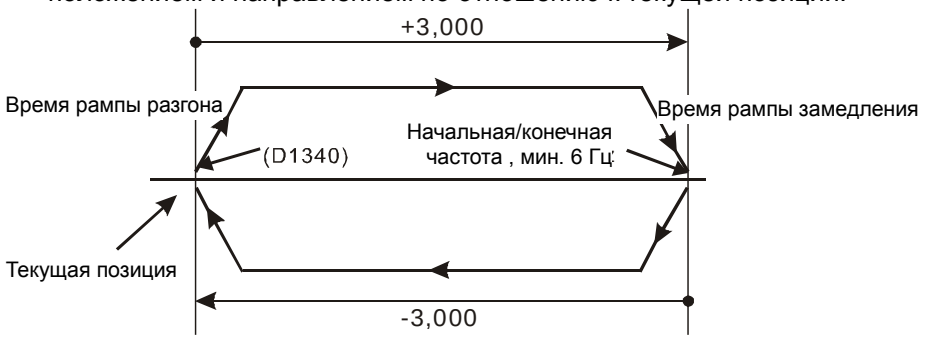
Пример программы:

Когда M10 включен, 20,000 импульсов (относительная позиция) частотой 2 кГц генерируются на Y0. Выключенный Y1 показывает прямое направление движения (положительное направление импульсов).

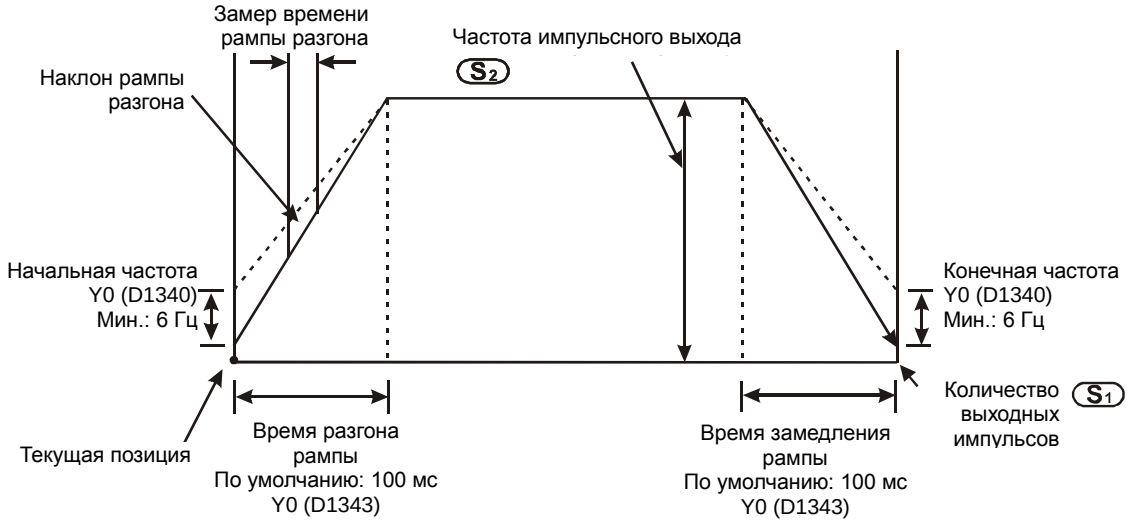


Заметки:

1. Импульсный выход обрабатывается в соответствии с текущим относительным положением и направлением по отношению к текущей позиции:



2. Регистры установки времени рамповой функции и начальной/конечной частоты:
   - Выход Y0:



   - Команда может использоваться в программе несколько раз, но одновременно не более одной команды. Если команда занимает выход Y0, другие команды использовать его не смогут.
   - После запуска команды, параметры не могут быть изменены до завершения работы команды.

3. Флаги:

M1029	CH0 (Y0, Y1) завершение обработки импульсного выхода
M1102	CH1 (Y2, Y3) завершение обработки импульсного выхода
M1078	CH0 (Y0, Y1) немедленная приостановка выдачи импульсов
M1104	CH1 (Y2, Y3) немедленная приостановка выдачи импульсов
M1108	CH0 (Y0, Y1) приостановка выдачи импульсов (рампа вниз)
M1110	CH1 (Y2, Y3) приостановка выдачи импульсов (рампа вниз)
M1156	Включение функций метки привода и маски на I400/I401(X4), относящимся к Y0.
M1158	Включение функций метки привода и маски на I600/I601(X6), относящимся к Y2.
M1305	Изменение направления импульсного выхода Y1 в командах обработки высокоскоростных импульсных выходов
M1306	Изменение направления импульсного выхода Y3 в командах обработки высокоскоростных импульсных выходов
M1347	Автосброс Y0 после завершения обработки высокоскоростного импульсного выхода
M1524	Автосброс Y2 после завершения обработки высокоскоростного импульсного выхода
M1534	Включение времени замедления рамповой функции на Y0. Должен использоваться с D1348
M1535	Включение времени замедления рамповой функции на Y2. Должен использоваться с D1349.
M1538	Указатель состояния паузы CH0 (Y0, Y1)
M1540	Указатель состояния паузы CH1 (Y2, Y3)

4. Специальные регистры D:

D1030	Младшее слово текущего значения импульсного выхода Y0
D1031	Старшее слово текущего значения импульсного выхода Y0
D1336	Младшее слово текущего значения импульсного выхода Y2
D1337	Старшее слово текущего значения импульсного выхода Y2
D1340	Начальная/конечная частота 1-й группы импульсных выходов CH0 (Y0, Y1)
D1352	Начальная/конечная частота 2-й группы импульсных выходов CH1 (Y2, Y3)
D1343	Время разгона/замедления рамповой функции 1-й группы импульсных выходов CH0 (Y0, Y1)
D1353	Время разгона/замедления рамповой функции 2-й группы импульсных выходов CH1 (Y2, Y3)
D1348:	Импульсный выход CH0 (Y0, Y1). Когда M1534 включен, D1348 сохраняет время рамповой функции
D1349:	Импульсный выход CH1 (Y2, Y3). Когда M1535 включен, D1349 сохраняет время рамповой функции
D1232	Количество выходных импульсов для рампы замедления при получении датчиком маски Y0 сигнала (младшее слово)
D1233	Количество выходных импульсов для рампы замедления при получении датчиком маски Y0 сигнала (старшее слово)
D1234	Количество выходных импульсов для рампы замедления при получении датчиком маски Y2 сигнала (младшее слово)
D1235	Количество выходных импульсов для рампы замедления при получении датчиком маски Y2 сигнала (старшее слово)
D1026	Количество импульсов для функции маски Y0 при включенном M1156 (младшее слово)
D1027	Количество импульсов для функции маски Y0 при включенном M1156 (старшее слово)
D1135	Количество импульсов для функции маски Y2 при включенном M1158 (младшее слово)
D1136	Количество импульсов для функции маски Y2 при включенном M1158 (старшее слово)

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
159	D	DRVA	S_1 S_2 D_1 D_2	Позиционирование в абсолютных координатах				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	DRVA: 9 шагов DDRVA: 17 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
D_1		*														
D_2		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Число импульсов (абсолютное позиционирование) **S_2**: Частота выходных импульсов
D_1: Импульсный выход **D_2**: Направление вращения

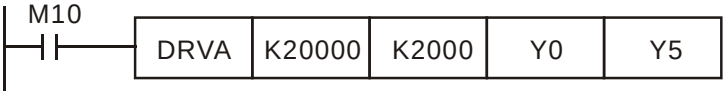
Описание:

1. Команда поддерживает только импульсные выходы типа: Pulse/Dir.
2. **S_1** – число импульсов (относительное позиционирование). Диапазон: -2,147,483,648 ~ +2,147,483,647. “+/-” показывает прямое и обратное движение.
3. **S_2** – частота выходных импульсов. Диапазон: 6 ~ 100,000 Гц.
4. **D_1** – импульсный выход. Применяются только выходы CH0(Y0) и CH1(Y2).
5. **D_2** – направление вращения. Применяются только выходы CH0(Y1) и CH1(Y3).
6. **S_1** – заданное положение абсолютной позиции. Фактическое количество выходных импульсов (**S_1** – текущая позиция) подсчитывается ПЛК. Когда результат положительный, импульсный выход реализует движение вперед, т.е. **D_2** выключен; при отрицательном результате – движение назад, т.е. **D_2** включен.
7. Расчетное количество импульсов определяет относительное положение:
   a. текущая позиция (32-битные данные) CH0 (Y0, Y1) хранится в D1031(старший байт), D1030 (младший байт)
   b. текущая позиция (32-битные данные) CH1 (Y2, Y3) хранится в D1337(старший байт),

D1336 (младший байт).

8. В режиме обратного хода импульсного выхода, значение в (D1031, D1330) и (D1336, D1337) уменьшается.
9. D1343 (D1353) – время рамповой (разгон/замедление) функции (между начальной частотой и частотой импульсного выхода), установленное в CH0 (CH1). Диапазон: 20 ~ 32,767 мс. По умолчанию: 100 мс. При выходе значения за пределы диапазона (выше или ниже), ПЛК установит значение равным 20 мс.
10. D1340 (D1352) – начальная/конечная частота, установленная в CH0 (CH1). Диапазон: 6 ~ 32,767 Гц.
11. M1305 и M1306 может менять направление выходных импульсов CH0/CH1, установленных в **D_2**. Когда **S** со знаком "-", **D_2** включен, однако, если включаются M1305/M1306 до начала выполнения команды, **D_2** выключается во время выполнения команды.
12. Время рамповой функции CH0 и CH1 может быть изменено с помощью (M1534, D1348) и (M1535, D1349). Когда M1534 / M1535 включены, время рамповой функции CH0 / CH1 определяется D1348 / D1349.
13. Если M1078 / M1104 включены во время выполнения команды, на Y0 / Y2 приостанавливается выдача импульсов, а включенные M1538 / M1540 показывают паузу в работе. Когда M1078 / M1104 выключены, M1538 / M1540 также выключены, Y0 / Y2 выдают оставшиеся импульсы.
14. Команда DRVA не поддерживает функций метки привода и маски.

Пример программы:



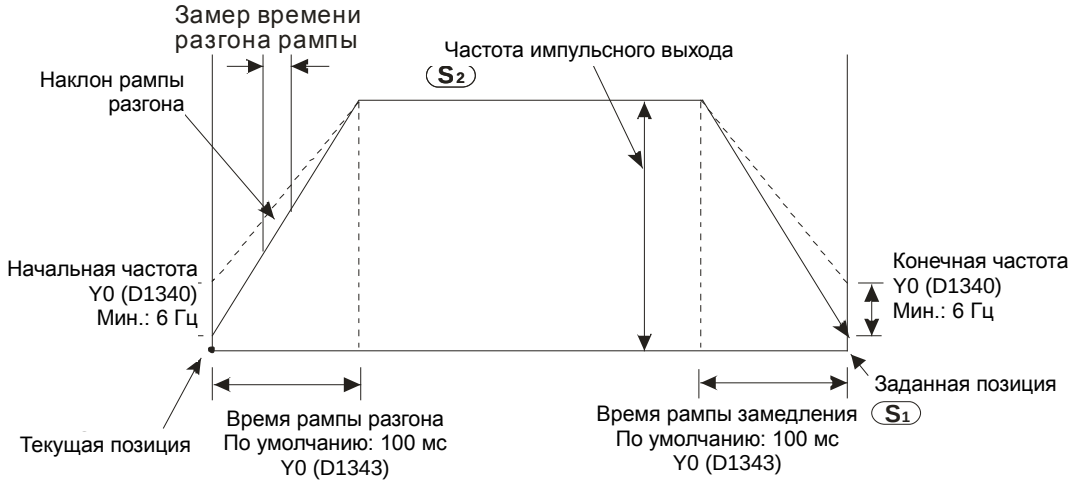
Когда M10 включен, команда DRVA выполняет абсолютное позиционирование на Y0 на заданную позицию 20000, заданная частота 2 кГц. Выключенный Y1 показывает прямое направление движения (положительное направление импульсов).

Заметки:

1. Импульсный выход обрабатывается в соответствии с текущим абсолютным положением и направлением по отношению к нулевой точке:



2. Регистры установки времени рамповой функции и начальной/конечной частоты: Выход Y0:



- Команда может использоваться в программе несколько раз, но одновременно не более одной команды. Если команда занимает выход Y0, другие команды использовать его не смогут.
- После запуска команды, параметры не могут быть изменены до завершения работы команды.
- Флаги и специальные регистры см. **Заметки** команды DDRVI.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
160	TCMP	P	S_1 S_2 S_3 S D	Сравнение времени				

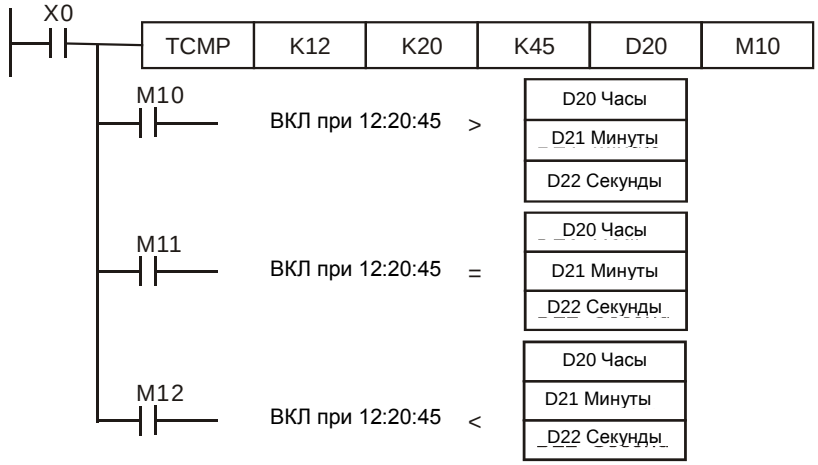
Операнд \ Тип	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	TCMP, TCMPP: 11 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	
S_3					*	*	*	*	*	*	*	*	*	*	*	
S											*	*	*			
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Часы (K0~K23) **S_2**: Минуты (K0~K59) **S_3**: Секунды (K0~K59) **S**: Текущее значение реального времени (RTC) (занимает 3 последовательных адреса) **D**: Результат (занимает 3 последовательных адреса)

Описание:
1. Команда TCMP сравнивает значение времени, заданное в **S_1**, **S_2**, **S_3** с текущим значением RTC-времени в **S** и сохраняет результат в **D**.
2. **S:** «Час» текущего RTC-времени (K0~K23) **S** +1: «Минута» текущего RTC-времени. (K0~K59) **S** +2**:** «Секунда» текущего RTC-времени (K0~K59).
3. Если значение **S** выходит за пределы допустимого диапазона, определяется ошибка и включатся M1067, M1068. D1067 сохраняет код ошибки 0E1A (шестнадцатеричный).

Пример программы:
1. При вкл. X0 текущее RTC-время в D20~D22 сравнивается с заданным значением 12:20:45. Результат см. состояние M10~M12. Когда X0 выкл., команда прекращает работу, однако состояние M10~M12 не меняется.
2. Подключение M10 ~ M12 последовательно или параллельно позволит отобразить результат в виде ≧, ≦ и ≠.



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
161	TZCP	P	S_1 S_2 S D	Зонное сравнение времени				

Операнд \ Тип	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TZCP, TZCPP: 9 шагов
S_2											*	*	*			
S											*	*	*			
D		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Нижняя граница времени для сравнения (занимает 3 последовательных адреса)
S_2: Верхняя граница времени для сравнения (занимает 3 последовательных адреса)

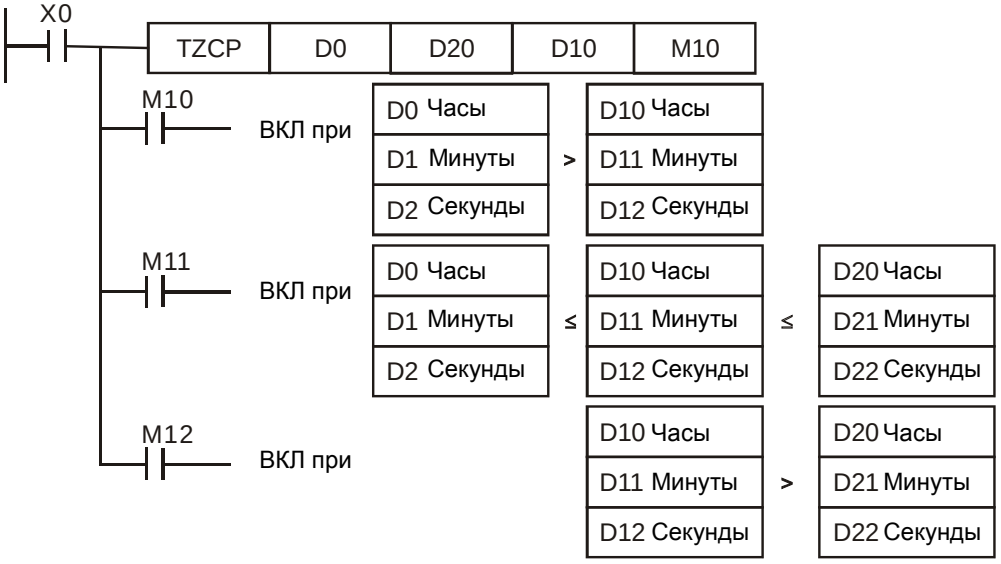
S: Текущее реальное время RTC (занимает 3 последовательных адреса) **D**: Результат (занимает 3 последовательных адреса)

Описание:

1. Команда TZCP сравнивает текущее RTC-время в **S** с диапазоном времени, заданным в S_1~ S_2 и сохраняет результат в **D**.
2. S_1, S_1 +1, S_1 +2: Часы, минуты и секунды нижней границы сравниваемого времени.
3. S_2, S_2 +1, S_2 +2: Часы, минуты и секунды верхней границы сравниваемого времени.
4. **S**, **S** +1, **S** +2: Часы, минуты и секунды текущего RTC-времени.
5. Если значение S, S_1, S_2 выходит за пределы допустимого диапазона, определяется ошибка и включатся M1067, M1068. D1067 сохраняет код ошибки 0E1A (шестнадцатеричный).
6. Если **S** < S_1 и **S** < S_2, включается **D**. Когда **S** > S_1 и **S** > S_2, включается **D**+2. Для прочих условий, включается **D** + 1. (нижняя граница S_1 должна быть меньше верхней S_2.)

Пример программы:

Когда X0 включен, выполняется команда TZCP и включаются M10~M12, отображающие результат сравнения. Когда X0 выключен, команда прекращает работу, но состояние M10~M12 не меняется.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
162		TADD	P	S_1 S_2 D	Сложение времени				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TADD, TADDP: 7 шагов
S_2											*	*	*			
D											*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

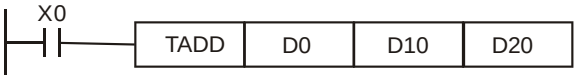
S_1: Слагаемое время (занимает 3 последовательных адреса) S_2: Слагаемое время (занимает 3 последовательных адреса) **D**: Результат (занимает 3 последовательных адреса)

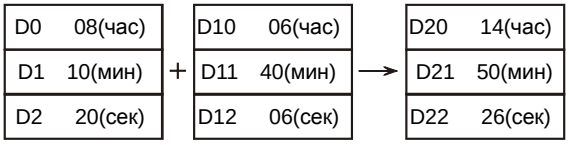
Описание:

1. Команда TADD складывает значение времени (часы, минуты, секунды) из S_1 со значением времени (часы, минуты, секунды) из S_2 и сохраняет результат в **D**.
2. Если значение S_1, S_2 выходит за пределы допустимого диапазона, определяется ошибка и включатся M1067, M1068. D1067 сохраняет код ошибки 0E1A (шестнадцатеричный).
3. Если результат больше 24 ч, включается флаг переноса M1022 и значение в **D** будет “сумма минус 24 ч”.
4. Если сумма равна 0 (00:00:00), включается флаг нуля M1020.

Пример программы:

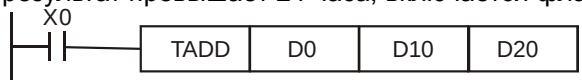
Когда X0 включен, значение времени в D0~D2 суммируется со значением времени в D10~D12. Результат сохраняется в D20~D22.

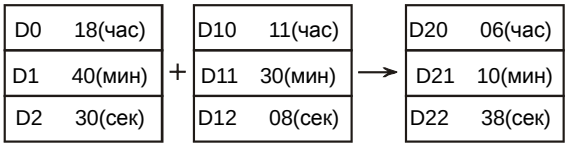




08:10:20 06:40:06 14:50:26

Если результат превышает 24 часа, включается флаг переноса M1022.





18:40:30 11:30:08 06:10:38

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
163		TSUB	P	S_1 S_2 D	Вычитание времени				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			TSUB, TSUBP: 7 шагов
S_2											*	*	*			
D											*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

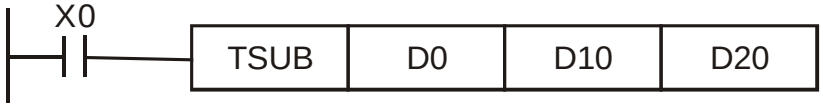
S_1: Уменьшаемое время (занимает 3 последовательных адреса) **S_2**: Вычитаемое время (занимает 3 последовательных адреса) **D**: Разность (занимает 3 последовательных адреса)

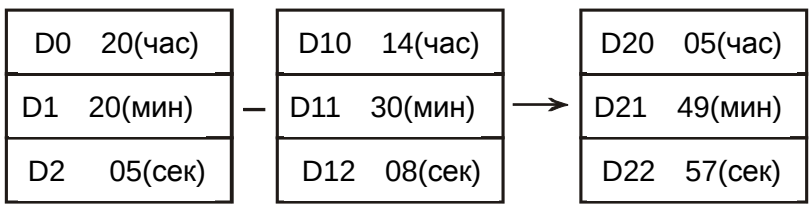
Описание:

1. Команда TSUB вычитает из значения времени (часы, минуты, секунды) из **S_1** значение времени (часы, минуты, секунды) из **S_2** и сохраняет результат в **D**.
2. Если значение **S_1**, **S_2** выходит за пределы допустимого диапазона, определяется ошибка и включатся M1067, M1068. D1067 сохраняет код ошибки 0E1A (шестнадцатеричный).
3. Если разность меньше 0, включается флаг заимствования M1020, и в **D** результат будет “отрицательное значение плюс 24 часа”.
4. Если разность равна 0 (00:00:00), включается флаг нуля M1020.
5. Помимо команды TRD, команда MOV может применяться для перемещения значения RTC-времени в D1315 (часы), D1314 (минуты), D1313 (секунды) для чтения текущего значения RTC-времени.

Пример программы:

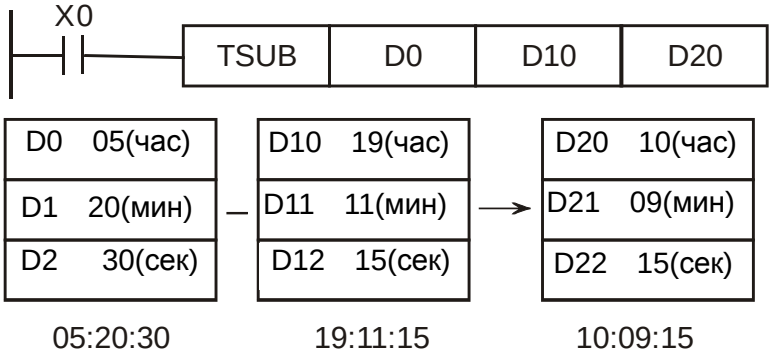
Когда X0 включен, команда TSUB вычитает из значения D0~D2 значение времени в D10~D12. Результат сохраняется в D20~D22.





20:20:05 14:30:08 05:49:57

Если разность меньше 0, включается флаг заимствования M1021.



API	Команда			Операнды	Функция	Контроллеры			
166		TRD	P	(D)	Чтение времени	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D											*	*	*			TRD, TRDP: 3 шага

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнд:

D: Текущее значение реального времени (RTC) (занимает 7 последовательных адресов)

Описание:

1. Команда TRD читает 7 данных реального времени (RTC) – год, день недели, месяц, день (число), час, минута, секунда из D1319~D1313 и сохраняет считанные данные в **D.**
2. Для контроллеров серии SS2: значение реального времени сохраняется только при включенном питании. При отключении питания рекомендуется корректировать значение после включения.
3. Значение реального времени RTC на ПЛК серий SA2/SE V1.0 и ES2/EX2/SX2 V2.0 могут читаться в течение 1-2 недель (в зависимости от температуры окружающей среды) после отключения питания. Если ПЛК был отключен от питания на больший срок, скорректируйте значение RTC-времени.
4. D1319 сохраняет только 2 цифры значения года. Если необходим формат в 4 цифры, см. **Заметки** ниже.
5. Флаги и специальные регистры см. **Заметки**.

Пример программы:

Когда X0 включен, команда TRD читает значение реального времени в специальные регистры D0~D6.
Содержимое D1318: 1 = Пн.; 2 = Вт. … 7 = Вс.



Специальный регистр D	Позиция	Содержимое		Регистр D	Позиция
D1319	Год	00~99	→	D0	Год
D1318	День (Пн.~Вс.)	1~7	→	D1	День (Пн.~Вс.)
D1317	Месяц	1~12	→	D2	Месяц
D1316	Число	1~31	→	D3	Число
D1315	Час	0~23	→	D4	Час
D1314	Минута	0~59	→	D5	Минута
D1313	Секунда	0~59	→	D6	Секунда

Заметки:

1. Есть 2 способа исправления RTC-времени:
   - С помощью команды API167 TWR. См. описание TWR (API 167)
   - Заданиями для внешних устройств. Используйте WPLSoft / ISPSoft
2. Индикация 4-циферного значения года: Необходимо в начале программы ввести команду (справа). D1319 сохраняет только 2-значное значение.



Отображение года 4мя цифрами

3. Флаги и специальные регистры RTC-времени

Флаг	Содержимое	Функция
M1016	Режим отображения года	OFF: D1319 сохраняет 2 цифры ON: D1319 сохраняет 2 цифры + 2000
M1017	±30 сек корректировка времени с точностью до 1 мин)	Коррекция происходит, когда включается M1017 (при 0 ~ 29 сек.: секунды сбрасываются до 0; при 30 ~ 59 сек.: к минуте прибавляется 1, секунды сбрасываются до 0)

Регистр	Содержимое	Диапазон
D1313	Секунда	0-59
D1314	Минута	0-59
D1315	Час	0-23
D1316	Число	1-31
D1317	Месяц	1-12
D1318	День недели (Пн. ~ Вс.)	1-7
D1319	Год	0-99 (отображаются 2 цифры)

API	Команда			Операнды	Функция	Контроллеры			
167		TWR	P	(S)	Запись времени	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			TWR, TWRP: 5 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнд:
S: Заданное значение RTC-времени (занимает 7 последовательных адресов)

Описание:
1. Команда TWR обновляет значение RTC-времени от значения, заданного в **S**.
2. Если значение времени в **S** превышают заданный диапазон, будет показана ошибка. ПЛК автоматически запишет минимально допустимое значение, включатся M1067, M1068 и код ошибки 0E1A (шестнадцатеричный) будет записан в D1067.
3. Флаги и специальные регистры см. **Заметки** команды TRD.

Пример программы 1:

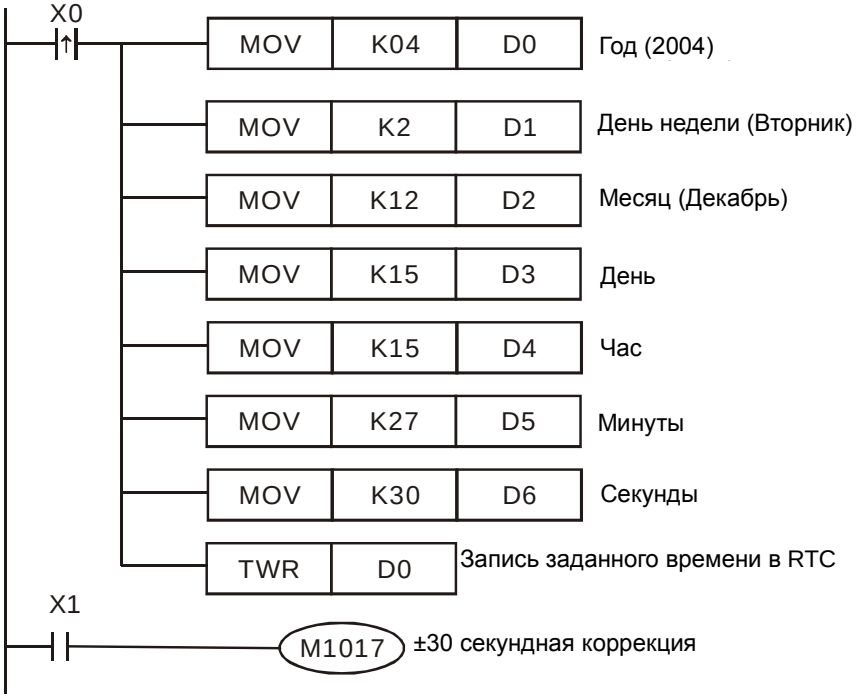


Когда X0 включен, записывается новое RTC-время.

	Регистр D	Позиция	Диапазон		Регистр D	Позиция	
Заданное значение	D20	Год	00~99	→	D1319	Год	RTC -время
	D21	День (Пн.~Вс.)	1~7	→	D1318	День (Пн.~Вс.)	
	D22	Месяц	1~12	→	D1317	Месяц	
	D23	Число	1~31	→	D1316	Число	
	D24	Час	0~23	→	D1315	Час	
	D25	Минута	0~59	→	D1314	Минута	
	D26	Секунда	0~59	→	D1313	Секунда	

Пример программы 2:

1. Установим реальное время 2004/12/15, вторник, 15:27:30.
2. Содержимое D0~D6 – установленное значение для корректировки RTC-времени.
3. Когда X0 включен, обновляется заданное значение времени.
4. Когда X1 включен, выполняется коррекция времени (±30 секунд). Коррекция происходит, когда включается M1017 (секунды в диапазоне 0 ~ 29: секунды сбрасываются до 0. Секунды в диапазоне 30 ~ 59: к минуте прибавляется 1, секунды сбрасываются до 0).



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
168	D	MVM	P	(S1) (S2) (D)	Маскировка и комбинация назначенных битов				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*	*	*	MVM, MVMP: 7 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	DMVM,DMVMP: 13 шагов
D							*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

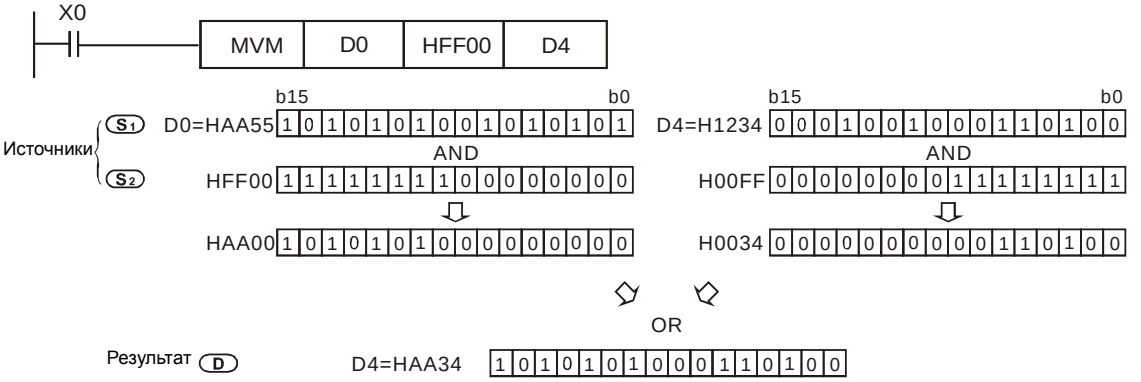
S_1: Источник 1 **S_2**: Биты для маскировки **D**: **D** =(**S_1** & **S_2**) | (**D** & ~ **S_2**)

Описание:

1. Команда проводит операцию «логическое И» сначала между **S_1** и **S_2**, затем между **D** и **S_2**, и комбинирует 1й и 2й результаты в **D** операцией «логическое ИЛИ».
2. Правила операции «логическое И»: 0 И 1 = 0, 1 И 0 = 0, 0 И 0 = 0, 1 И 1 = 1
3. Правила операции «логическое ИЛИ»: 0 ИЛИ 1= 1, 1 ИЛИ 0 = 1, 0 ИЛИ 0 = 0, 1 ИЛИ 1 = 1.

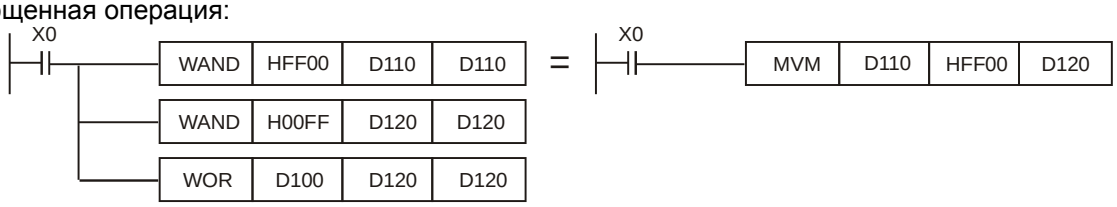
Пример программы 1 :

Когда X0 включен, команда MVM проводит операцию «логическое И» сначала между 16-битным регистром D0 и H'FF00, затем операцию «логическое И» между D4 и H'00FF, и комбинирует 1й и 2й результаты в D4 операцией «логическое ИЛИ».



Пример программы 2 :

Упрощенная операция:



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
169	D	HOUR		(S) (D1) (D2)	Счетчик часов				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	HOUR: 7 шагов
D_1													*			DHOUR: 13 шагов
D_2		*	*	*												

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

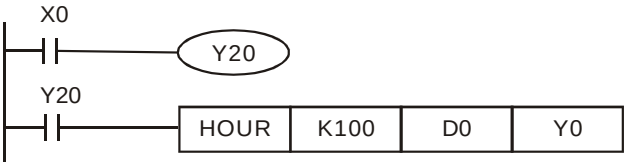
Операнды:

S: Заданное значение времени для включения выхода **D_2** (часы) **D_1**: Измеренное текущее время **D_2**: Выход

Описание:

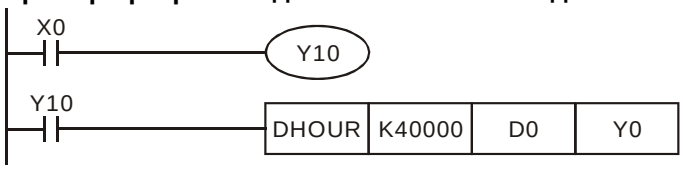
1. Команда HOUR запускает выход D_2, когда измеренное текущее время D_1 достигает заданного значения в **S**.
2. Диапазон **S**: K1~K32 767; ед.изм.: часы. Диапазон D_1 для 16-битной команды: K0~K32 767. Диапазон D_1 +1 (текущее время менее часа): K0 ~K3 599; ед.изм: секунды.
3. После включения устройства, связанного с выходом D_2, текущее время все равно будет измеряться в D_1.
4. Для 16-битной команды, когда текущее измеренное время достигает максимума 32,767 часов / 3,599 секунд, измерение времени остановится. Для повторного запуска измерения D_1 и D_1 + 1 должны быть сброшены.
5. Для 32-битной команды, когда текущее измеренное время достигает максимума 2 147 483 647 часов / 3 599 секунд, измерение времени остановится. Для повторного запуска измерения D_1 ~ D_1 + 2 должны быть сброшены.
6. Если операнд **S** имеет значение F, возможно выполнение только 16-битных команд.
7. Команда HOUR может быть использована до 4-х раз в программе.

Пример программы для 16-битной команды:



X0 включает Y20 и начинается отсчет времени. После отсчета 100 часов, включается Y0 и в D0 записывается текущее значение подсчитанного времени (часы). D1 записывает текущее время, меньшее 1 часа (0 ~ 3 599; сек.).

Пример программы для 32-битной команды:



X0 включает Y10 и начинается отсчет времени. После отсчета 40 000 часов, включается Y0. D1 и D0 записывают отсчитанное время (часы), а D2 записывает текущее время, меньшее 1 часа (0 ~ 3 599; секунды).

3

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
170	D	GRY	P	(S) (D)	Преобразование числа в код Грея				

Тип / Операнд	Биты				Слова										
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F
S					*	*	*	*	*	*	*	*	*	*	*
D								*	*	*	*	*	*	*	*

Шаги программы

GRY, GRYP: 5 шагов
DGRY, DGRYP: 9 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S: Источник данных **D**: Результат операции (код Грея)

Описание:

1. Команда GRY преобразует двоичное значение из **S** в код Грея, сохраняя результат в специальном регистре **D**.
2. Диапазон **S**:
   16-битная команда: 0~32,767, 32-битная команда: 0~2,147,483,647
3. Если значение операнда **S** выходит за пределы допустимого диапазона, возникает ошибка и включаются флаги M1067 и M1068. В D1067 сохраняется код ошибки 0E1A (в шестнадцатеричном формате)
4. Если S и D имеют значение F, возможно только 16-битное выполнение команды.

Пример программы:

При включенном X0 команда GRY преобразует K6513 в код Грея. Результат сохраняется в K4Y20, т.е. Y20 ~ Y37.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
171	D	GBIN	P	(S) (D)	Преобразование кода Грея в число				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*	*	*	GBIN, GBINP: 5 шагов DGBIN, DGBINP: 9 шагов
D								*	*	*	*	*	*	*	*	

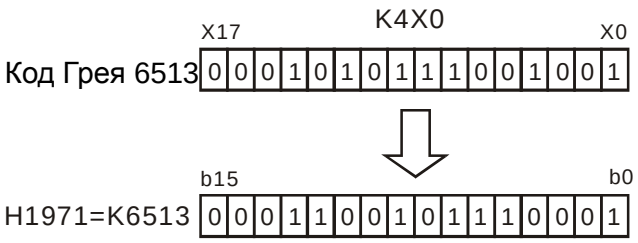
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник данных **D**: Результат операции (двоичное значение)

Описание:
1. Команда GBIN преобразует код Грея из **S** в двоичное значение и сохраняет результат в специальный регистр **D**.
2. Эта команда может применяться для чтения значения абсолютных энкодеров (генерирующих код Грея), подключенных к входами ПЛК.
3. Диапазон **S**: 16-битная команда: 0~32,767; 32-битная команда: 0~2,147,483,647
4. Если значение операнда **S** выходит за пределы допустимого диапазона, возникает ошибка и выполнение команды прекращается.
5. Если S и D имеют значение F, возможно только 16-битное выполнение команды.

Пример программы:
При включенном X20 код Грея от абсолютного энкодера, подключенного к входам X0~X17, преобразуется в двоичное значение и сохраняется в D10.





API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
172	D	ADDR	P	(S1) (S2) (D)	Сложение с использованием регистра				

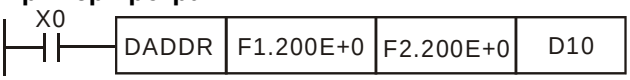
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DADDR, DADDRP: 13 шагов
S_2													*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1 и **S_2:** Слагаемые (число с плавающей запятой) **D:** Сумма

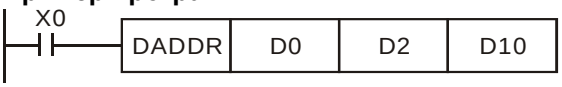
Описание:
1. Команда ADDR суммирует числа с плавающей запятой **S_1** и **S_2** и сохраняет результат в **D**.
2. В команде ADDR числа с плавающей запятой могут быть введены непосредственно в **S_1** и **S_2**, а в DADDR - в **S_1** и **S_2** или сохранены в регистре данных.
3. Когда **S_1** и **S_2** находятся в регистре данных, работа команды DADDR аналогична команде API 120 EADD.
4. **S_1** и **S_2** могут находиться в одном и том же регистре. В этом случае, если команда работает в режиме непрерывного выполнения (как правило, команда DADDRP) и условия выполнения команды включены, сложение будет происходить один раз в каждом цикле.
5. Флаги: M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса)
   Если абсолютное значение результата превышает допустимое, включается флаг переноса M1022. Если абсолютное значение результата ниже допустимого, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



Когда X0 включен, суммируются числа с плавающей запятой F1.200E+0 (задание вида отображаемого числа F1.2 или F1.200E+0 с помощью функции View) с F2.200E+0 и результат F3.400E+0 сохраняется в регистрах D10 и D11.

Пример программы 2:



Когда X0 включен, суммируются значения с плавающей запятой (D1, D0) и (D3, D2), результат сохраняется в (D11, D10).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
173	D	SUBR	P	S_1 S_2 D	Вычитание с использованием регистра				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DSUBR: 13 шагов
S_2													*			
D													*			

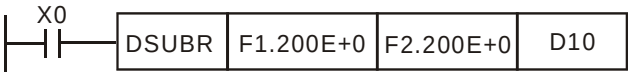
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходное значение; **S_2:** Вычитаемое (**S_1, S_2** - числа с плавающей запятой); **D:** Разница

Описание:

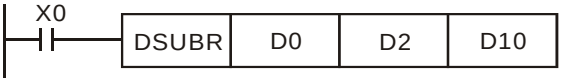
1. Команда SUBR вычитает из **S_1** значение в **S_2** и сохраняет результат в **D**.
2. В команде SUBR числа с плавающей запятой могут быть введены в непосредственно в **S_1** и **S_2**, а в команде DSUBR - в **S_1** и **S_2** или сохранены в регистре данных.
3. Когда **S_1** и **S_2** находятся в регистре данных, команда DADDR работает как API 121 ESUB.
4. **S_1** и **S_2** могут находиться в одном и том же регистре. В этом случае, если команда работает в режиме непрерывного выполнения (как правило, команда DSUBR) и условия выполнения команды включены, вычитание будет происходить один раз в каждом цикле.
5. Флаги: M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса)
   Если абсолютное значение результата превышает допустимое, включается флаг переноса M1022. Если абсолютное значение результата ниже допустимого, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



Когда X0 включен, из числа с плавающей запятой F1.200E+0 (задание вида отображаемого числа F1.2 или F1.200E+0 с помощью функции View) вычитается F2.200E+0 и результат F-1.000E+0 сохраняется в регистрах D10 и D11.

Пример программы 2:



Когда X0 включен, из числа с плавающей запятой (D1, D0) вычитается число с плавающей запятой (D3, D2), результат сохраняется в (D11, D10).

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
174	D	MULR	P	S_1 S_2 D	Умножение с использованием регистра				

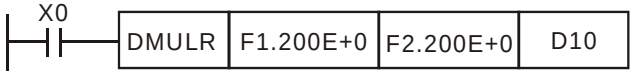
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1													*			DMULR, DMULRP: 13 шагов
S_2													*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Множимое **S_2:** Множитель (**S_1, S_2** - числа с плавающей запятой) **D:** Произведение

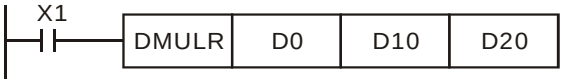
Описание:
1. Команда MULR перемножает числа из **S_1** и **S_2** и сохраняет результат в **D**.
2. В команде MULR числа с плавающей запятой могут быть введены непосредственно в **S_1** и **S_2**, а в команде DMULR - в **S_1** и **S_2** или сохранены в регистре данных.
3. Когда **S_1** и **S_2** находятся в регистре данных, команда DMULR работает как API 122 EMUL.
4. **S_1** и **S_2** могут находиться в одном и том же регистре. В этом случае, Если команда работает в режиме непрерывного выполнения (как правило, команда DMULR) и условия выполнения команды включены, умножение будет происходить один раз в каждом цикле.
5. Флаги: M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса)
   Если абсолютное значение результата превышает допустимое, включается флаг переноса M1022. Если абсолютное значение результата ниже допустимого, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



Когда X0 включен, перемножаются числа с плавающей запятой F1.200E+0 (пользователь может задать вид отображаемого числа как F1.2 или F1.200E+0 с помощью функции View) с F2.200E+0 и результат F2.640E+0 сохраняется в регистрах D10 и D11.

Пример программы 2:



Когда X0 включен, перемножаются значения с плавающей запятой (D1, D0) и (D3, D2), результат сохраняется в (D11, D10).

API	Команда			Операнды	Функция	Контроллеры			
175	D	DIVR	P	S_1 S_2 D	Деление с использованием регистра	ES2/EX2	SS2	SA2/SE	SX2

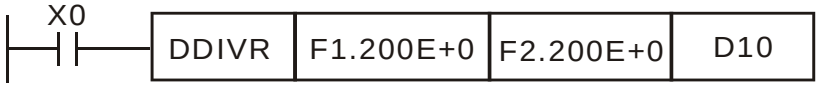
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DIVR: 13 шагов
S_1													*			
S_2													*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Делимое **S_2:** Делитель (**S_1, S_2** - числа с плавающей запятой) **D:** Частное

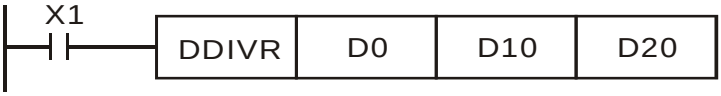
Описание:
1. Команда DIVR делит значение из **S_1** на значение в **S_2** и сохраняет результат в **D**.
2. В команде DIVR числа с плавающей запятой могут быть введены в **S_1** и **S_2**, а в команде DDIVR - в **S_1** и **S_2** или сохранены в регистре данных.
3. Когда **S_1** и **S_2** находятся в регистре данных, команда DDIVR работает как API 123 EDIV.
4. **S_1** и **S_2** могут находиться в одном и том же регистре. В этом случае, Если команда работает в режиме непрерывного выполнения (как правило, команда DDIVR) и условия выполнения команды включены, вычитание будет происходить один раз в каждом цикле.
5. Если **S_2** = 0, возникает ошибка и включаются флаги M1067, M1068. В D1067 сохраняется код ошибки 0E19 (шестнадцатеричный).
6. Флаги: M1020 (флаг нуля), M1021 (флаг заимствования) и M1022 (флаг переноса)
   Если абсолютное значение результата превышает допустимое, включается флаг переноса M1022. Если абсолютное значение результата ниже допустимого, включается флаг заимствования M1021. Если результат равен 0, включается флаг нуля M1020.

Пример программы 1:



Когда X0 включен, число с плавающей запятой F1.200E+0 (пользователь может задать вид отображаемого числа как F1.2 или F1.200E+0 с помощью функции View) делится на F2.200E+0 и результат F0.545E+0 сохраняется в регистрах D10 и D11.

Пример программы 2:



Когда X0 включен, число с плавающей запятой (D1, D0) делится на число с плавающей запятой (D3, D2), результат сохраняется в (D11, D10).

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
176	MMOV	P	(S) (D)	Перевод 16 бит в 32 бит				

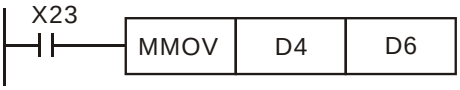
Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*	*	*	*	*	*	*	*			MMOV, MMOVP: 5 шагов
D											*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

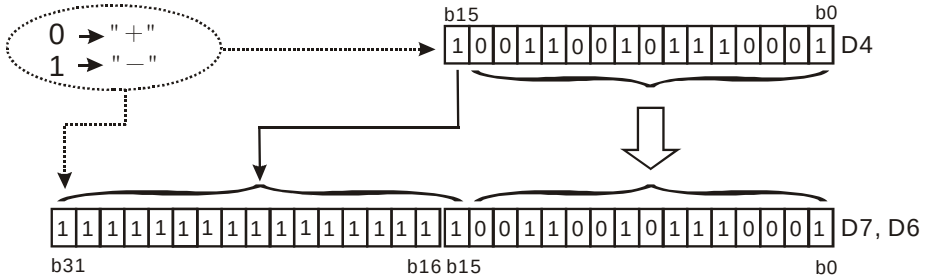
Операнды:
S: Источник данных (16-битные данные) **D:** Результат перевода (32-битные данные)

Описание: Команда MMOV передает данные из 16-битного **S** в 32-битный **D**. Входящие биты (MSB) из источника будут скопированы побитово в старший байт **D**.

Пример программы:



Когда X23 включен, 16-битные данные из D4 будут переданы в D6 и D7.



В примере b15 из D4 передается в b15~b31 в D7/D6, поэтому все биты в b15~b31 «отрицательны».

API	Команда	Операнды	Функция	Контроллеры			
				ES2/EX2	SS2	SA2/SE	SX2
177	GPS	(S) (D)	Прием GPS данных				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			GPS: 5 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Идентификатор принимаемых данных GPS **D**: Операнд данных обратной связи

Описание:
1. Команда GPS работает только на порте COM1 (RS-232), формат связи: 9600,8,N,1, протокол: NMEA-0183, частота связи: 1 Гц.
2. Операнд **S** – идентификатор принимаемых данных GPS. K0: $GPGGA, K1: $GPRMC.
3. Операнд **D** сохраняет принимаемые данные. Данные занимают до 17 последовательных слов. См. описание операндов **D**.

Когда **S** имеет значение K0, идентификатор - $GPGGA. Описания **D**:

№.	*Содержимое*	*Диапазон*	*Формат*	*Примечание*
D *+ 0*	*Часы*	*0 ~ 23*	*Слово*	
D *+ 1*	*Минуты*	*0 ~ 59*	*Слово*	
D *+ 2*	*Секунды*	*0 ~ 59*	*Слово*	

№.	Содержимое	Диапазон	Формат	Примечание
D + 3~4	Широта	0 ~ 90	Плавающ.	формат: гг.ммммм
D + 5	Север / Юг	0 или 1	Слово	0(+)→север, 1(-)→юг
D + 6~7	Долгота	0 ~ 180	Плавающ.	формат: ггг.ммммм
D + 8	Восток / Запад	0 или 1	Слово	0(+)→восток, 1(-)→запад
D + 9	Данные GPS допустимые / недопустимые	0, 1, 2	Слово	0 = недопустимые
D + 10~11	Высота	0 ~9999.9	Плавающ.	формат: метры
D + 12~13	Широта	-90 ~ 90	Плавающ.	формат: ±гг.гггг
D + 14~15	Долгота	-180 ~ 180	Плавающ.	формат: ±ггг.гггг

Когда **S** имеет значение K1, идентификатор - $GPRMC. Описания **D**:

№.	Содержимое	Диапазон	Формат	Примечание
D + 0	Часы	0 ~ 23	Слово	
D + 1	Минуты	0 ~ 59	Слово	
D + 2	Секунды	0 ~ 59	Слово	
D + 3~4	Широта	0 ~ 90	Плавающ.	формат: гг.ммммм
D + 5	Север / Юг	0 или 1	Слово	0(+)→север, 1(-)→юг
D + 6~7	Долгота	0 ~ 180	Плавающ.	формат: ггг.ммммм
D + 8	Восток / Запад	0 или 1	Слово	0(+)→восток, 1(-)→запад
D + 9	Данные GPS допустимые / недопустимые	0, 1, 2	Слово	0 = недопустимые
D + 10	День	1 ~ 31	Слово	
D + 11	Месяц	1 ~ 12	Слово	
D + 12	Год	2000 ~	Слово	
D + 13~14	Широта	-90 ~ 90	Плавающ.	формат: ±гг.гггг
D + 15~16	Долгота	-180 ~ 180	Плавающ.	формат: ±ггг.гггг

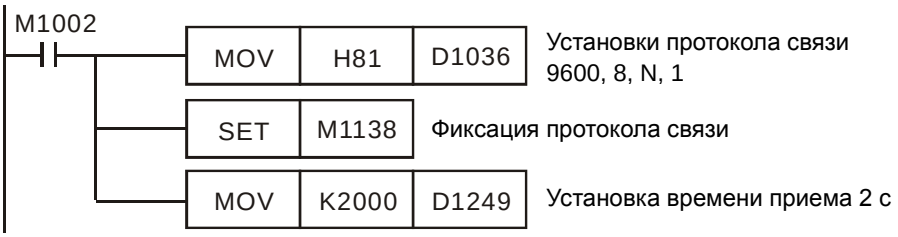
4. При применении команды GPS, COM1 должен применяться в мастер-режиме, т.е. M1312 должен быть включен для отправки запроса. Включенный M1314 показывает завершение приема данных, а включенный M1315 – ошибку приема. (D1250 = K1, задержка приема; D1250 = K2, ошибка контрольной суммы)
5. Флаги и специальные регистры:

№	Функция
M1312	COM1 (RS-232) отправка запроса
M1313	COM1 (RS-232) готовность приема данных
M1314	COM1 (RS-232) завершение приема данных
M1315	COM1 (RS-232) ошибка приема данных
M1138	Сохранение настроек связи порта COM1
D1036	COM1 (RS-232) протокол связи
D1249	COM1 (RS-232) установка задержки приема данных (рекомендуется: >1с).
D1250	COM1 (RS-232) код ошибки связи

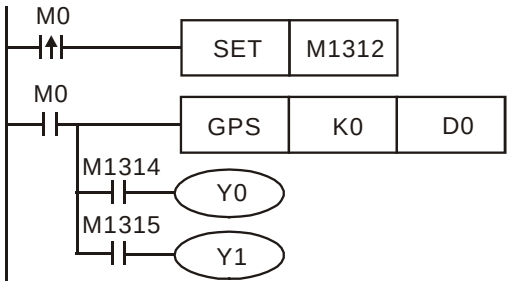
6. Перед использованием полученных данных GPS проверьте значение в **D**+9. Если **D**+9 = 0, данные GPS недопустимые.
7. Если при получении данных возникает ошибка, предыдущие данные в регистре **D** не будут удалены.

Пример программы: Идентификатор: $GPGGA

1. Сначала задается протокол связи COM1



2. Затем включается M0 для запуска команды GPS с идентификатором $GPGGA

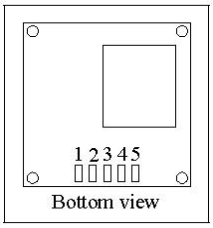


3. После завершения приема включается M1314. При неудачном приеме включается M1315. Принятые данные сохраняются в регистрах, начиная с D0.

№.	Содержимое	№	Содержимое
D0	Часы	D8	Восток / Запад
D1	Минуты	D9	Данные GPS допустимые / недопустимые
D2	Секунды	D10~D11	Высота
D3~D4	Широта	D12~D13	Широта. формат: ±гг.ггггг
D5	Север / Юг	D14~D15	Долгота. формат: ±ггг.ггггг
D6~D7	Долгота		

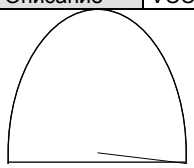
4. Контакты модуля GPS (LS20022)

№ контакта GPS	1	2	3	4	5
Описание	VCC(+5B)	Rx	Tx	GND	GND



5. Контакты ПЛК COM1:

№ контакта COM1	1	2	3	4	5	6	7	8
Описание	VCC(+5B)		--	Rx	Tx	--	--	GND



API	Команда		Операнды	Функция	Контроллеры			
178	D	SPA	(S) (D)	Позиционирование солнечной батареи	ES2/EX2	SS2	SA2/SE	SX2

Тип Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S					*	*							*			DSPA: 9 шагов
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S: Стартовый операнд входных параметров **D**: Стартовый операнд выходных параметров

Описание:

1. Операнд **S** занимает 208 последовательных словных регистров. Функции:

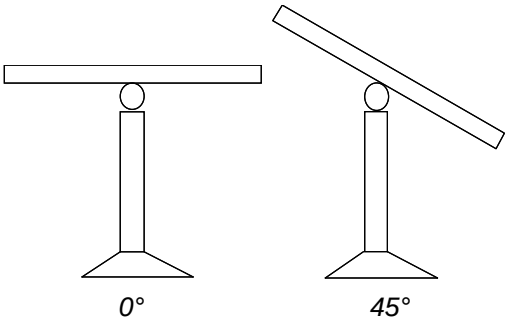
№	Содержимое	Диапазон	Формат	Примечание
S + 0	Год	2000 ~	Слово	Введите точное время местной долготы. Обратитесь к DTM (параметр 11) для использования формулы перевода. См. п. 6.
S + 1	Месяц	1 ~ 12	Слово	
S + 2	День	1 ~ 31	Слово	
S + 3	Часы	0 ~ 23	Слово	
S + 4	Минуты	0 ~ 59	Слово	
S + 5	Секунды	0 ~ 59	Слово	
S + 6~7	Разница во времени (Δt) (сек)	± 8000	Плав.	
S + 8~9	Часовой пояс	± 12	Плав.	Запад: отрицательный
S + 10~11	Долгота	± 180	Плав.	Запад: отрицательный формат: градусы
S + 12~13	Широта	± 90	Плав.	Юг: отрицательный формат: градусы
S + 14~15	Высота	0~ 6500000	Плав.	формат: метры
S + 16~17	Давление	0 ~ 5000	Плав.	формат: миллибары
S + 18~19	Среднегодовая температура (MAT)	-273~6000	Плав.	формат: °C
S + 20~21	Наклон	± 360	Плав.	
S + 22~23	Азимут	± 360	Плав.	
S + 24~25	Атмосферная рефракция между восходом и закатом	± 5	Плав.	
S +26~207	Системные			

2. Операнд **D** занимает 8 последовательных словных регистров. Функции:

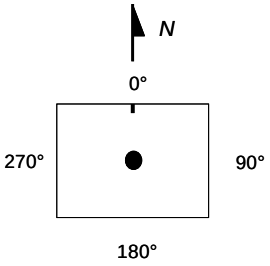
№	Содержимое	Диапазон	Формат	Примечание
D + 0~1	Зенит	0 ~ 90	Плав.	горизонтально=0
D + 2~3	Азимут	0 ~ 360	Плав.	север=0
D + 4~5	Наклон	0 ~ 90	Плав.	
D + 6	Преобразованное DA значение зенита	0 ~ 2000	Слово	1LSB = 0.045 град
D + 7	Преобразованное DA значение азимута	0 ~ 2000	Слово	1LSB = 0.18 град

3. Время работы команды SPA достигает 50 мс, поэтому рекомендуется выполнять данные команды с интервалом не менее 1 сек во избежание слишком большого времени работы ПЛК.

4. Определение зенита: 0° и 45°.

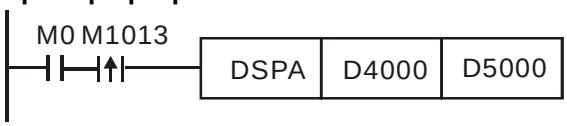


5. Определение азимута:



6. Точное время местной долготы: Предположим, что время равно 8:00:00 в Тайбее, долгота 121.55 град. з.д, тогда точное время долготы Тайбея составит 8:06:12. См. команду API168 DTM (параметр k11).

Пример программы:



Входные параметры, начиная с D4000: 2009/3/23/(г/м/д),10:10:30, Δt = 0, временной пояс = +8, долгота/широта = +119.192345 в.д., +24.593456 с.ш., высота = 132.2 м, давление = 820 мбар, температура MAT = 15.0℃, склон = 0 град., азимут = -10 град. Результаты на выходе: D5000: зенит = F37.2394 град; D5002: азимут = F124.7042 град.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
179	D	WSUM	P	S D n	Суммирование данных с нескольких устройств				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S											*	*	*			WSUM, WSUMP: 7 шагов
n					*	*							*			DWSUM, DWSUMP: 13 шагов
D											*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

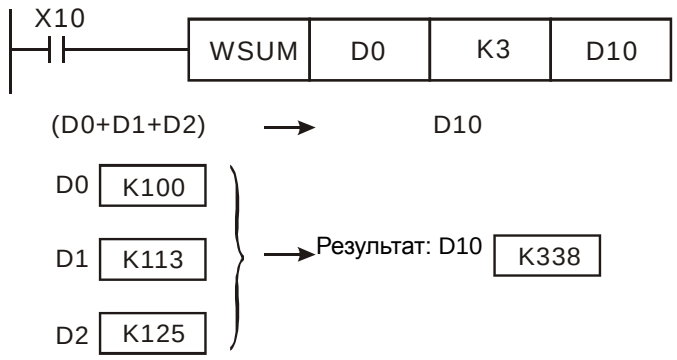
S: Источник данных **n**: Длина суммируемых данных **D**: Адрес хранения результата

Описание:

1. WSUM суммирует данные с **n** источников, начиная с **S,** и сохраняет результат в **D.**
2. Если какие-то значения в **S** выходят за пределы допустимого диапазона, суммируются только значения из источников, находящихся в допустимом диапазоне.
3. Диапазон **n**: 1~64. Если значение **n** выходит за эти пределы (1~64), ПЛК использует соответственно, верхнее значение (64) – при превышении или нижнее (1).

Пример программы:

Когда X10 включен, данные 3-х последовательных адресов (**n** = 3), начиная с D0, суммируются и результат сохраняется в D10



API	Команда			Операнды	Функция	Контроллеры			
180		MAND	P	S_1 S_2 D n	«Логическое И» для матриц	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MAND, MANDP: 9 шагов
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

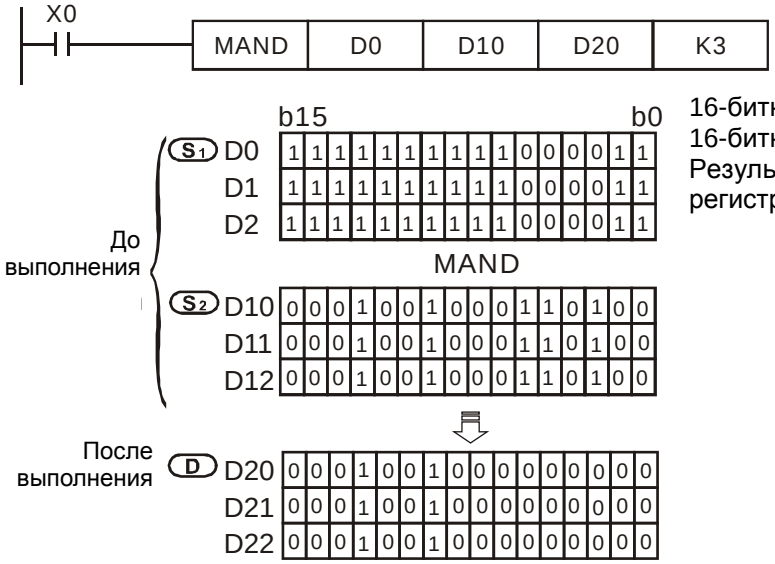
Операнды:

S_1: Исходная матрица 1 **S_2**: Исходная матрица 2 **D**: Результирующая матрица
n: Число строк матрицы (**n** = K1~K256)

Описание:

1. Команда MAND выполняет операцию «логическое И» между исходными матрицами 1 и 2 с количеством строк **n** и сохраняет результат в **D**.
2. Правило операции «логическое И»: результат равен 1 если оба бита равны 1; в других случаях результат равен 0.
3. Если операнды **S_1**, **S_2**, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

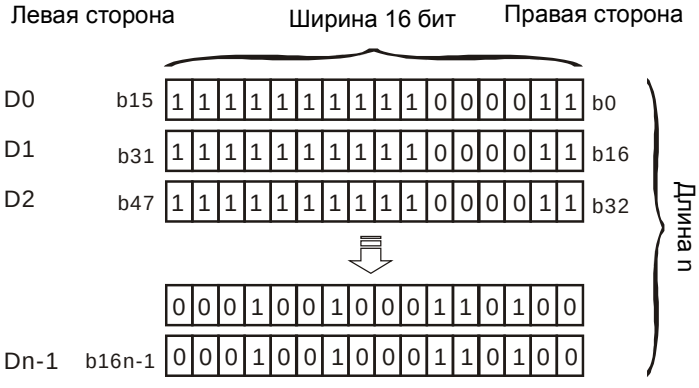
Пример программы:



Когда X0 включен, команда MAND проводит операцию «логическое И» между 16-битными регистрами D0~D2 и 16-битными регистрами D10~D12. Результат сохраняется в 16-битных регистрах D20~D22.

Заметки:

1. Матрица состоит из более чем одной последовательности 16-битных регистров. Количество показывает длину матрицы (n). Матрица содержит 16 × n битов и команды операций с матрицами проводят их побитно.
2. Команды назначают один бит из 16 × n бит (b_0 ~ b_{16n-1}) для проведения операций. Этот бит не содержит данных.
3. Операциям с матрицами требуется один 16-битный регистр для обозначения одного из 16n битов в матрице – регистр указателя (Pr) матрицы, назначаемый пользователем. Диапазон Pr: 0 ~ 16n -1, соответствует b0 ~ b16n-1 в матрице.
4. Номер бита убывает слева направо (см. рис.). Номер бита определяет его при проведении операций побитового сдвига, поворота и т.п.



5. Ширина матрицы (C) всегда равна 16 бит.
6. Pr: указатель матрицы. Если Pr = 15, назначенный указателем бит - b15.
7. Длина матрицы (R) n: n = 1 ~ 256.
   Пример: Эта матрица состоит из: D0, n = 3; D0 = HAAAA, D1 = H5555, D2 = HAAFF

	C_{15}	C_{14}	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	
R_0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	D0
R_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	D1
R_2	1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	D2

   Пример: Эта матрица состоит из: K2X20, n = 3; K2X20 = H37, K2X30 = H68, K2X40 = H45

	C_{15}	C_{14}	C_{13}	C_{12}	C_{11}	C_{10}	C_9	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1	C_0	
R_0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	X_{20}~X_{27}
R_1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	X_{30}~X_{37}
R_2	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	X_{40}~X_{47}

   Заполните значениями “0” свободные ячейки R0(C_{15}-C_8), R1(C_{15}-C_8) и R2(C_{15}-C_8).

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
181	MOR	P	(S1) (S2) (D) (n)	Логическое ИЛИ для матриц				

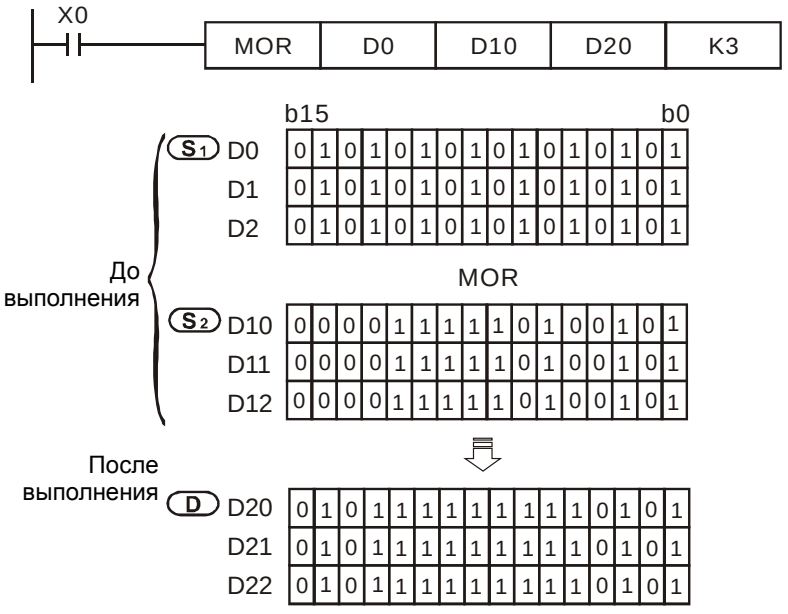
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MOR, MORP: 9 шагов
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходная матрица 1 **S_2**: Исходная матрица 2 **D**: Результирующая матрица
n: Число строк матрицы (**n** = K1~K256)

Описание:
1. Команда MOR выполняет операцию «логическое ИЛИ» между исходными матрицами 1 и 2 с количеством строк **n** и сохраняет результат в **D**.
2. Правило операции «логическое ИЛИ»: результат равен 1, если хотя бы один бит равен 1; если оба бита равны 0, результат равен 0.
3. Если операнды **S_1**, **S_2**, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:
Когда X0 включен, команда MOR проводит операцию «логическое ИЛИ» между 16-битными регистрами D0~D2 и 16-битными регистрами D10~D12. Результат сохраняется в 16-битных регистрах D20~D22.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
182		MXOR	P	S_1 S_2 D n	«Исключающее ИЛИ» для матриц				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MXOR, MXORP: 9 шагов
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

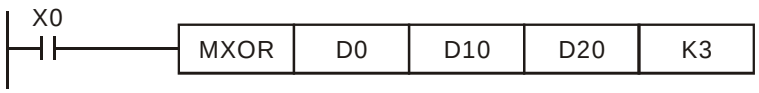
Операнды:

S_1: Исходная матрица 1 **S_2**: Исходная матрица 2 **D**: Результирующая матрица
n: Число строк матрицы (**n** = K1~K256)

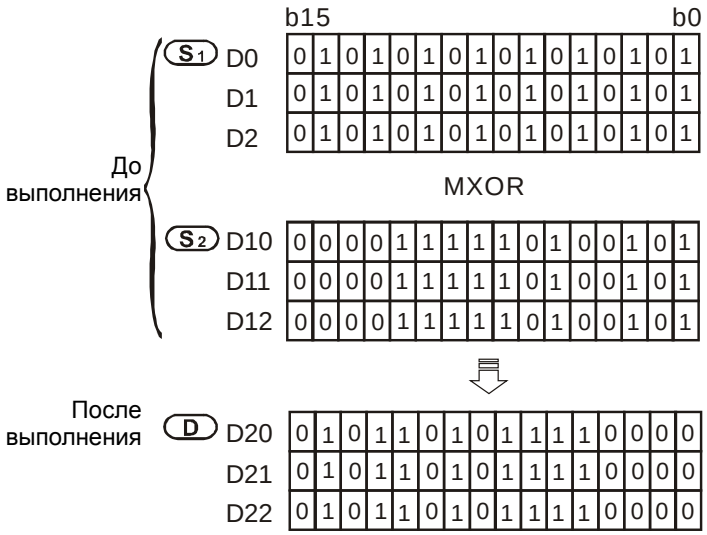
Описание:

1. Команда MXOR выполняет операцию «исключающее ИЛИ» между исходными матрицами 1 и 2 с количеством строк **n** и сохраняет результат в **D**.
2. Правило операции «исключающее ИЛИ»: результат равен 1, если оба бита разные и 0, если одинаковые.
3. Если операнды **S_1**, **S_2**, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:



Когда X0 включен, команда MXOR проводит операцию «исключающее ИЛИ» между 16-битными регистрами D0~D2 и 16-битными регистрами D10~D12. Результат сохраняется в 16-битных регистрах D20~D22.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
183		MXNR	P	S_1 S_2 D n	«Исключающее НЕ-ИЛИ» для матриц				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MXNR, MXNRP: 9 шагов
S_2							*	*	*	*	*	*	*			
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

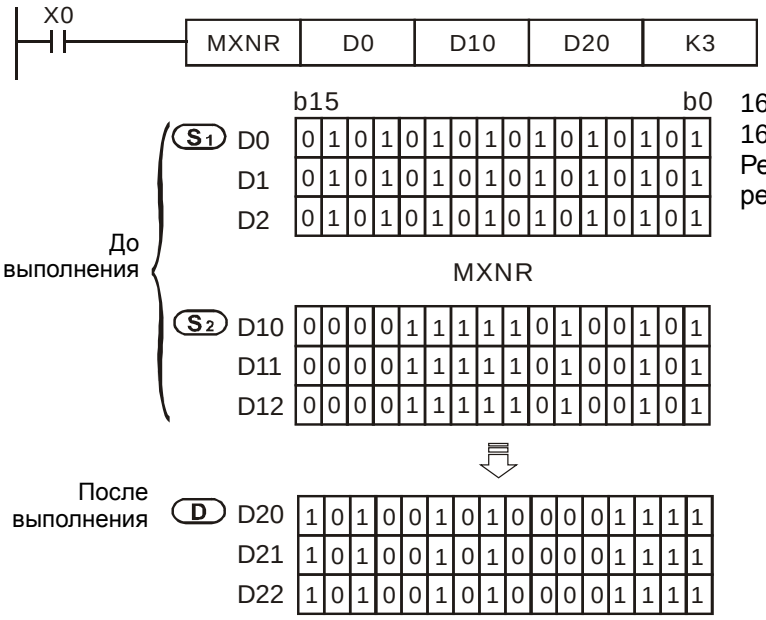
Операнды:

S_1: Исходная матрица 1 **S_2**: Исходная матрица 2 **D**: Результирующая матрица
n: Число строк матрицы (**n** = K1~K256)

Описание:

1. Команда MXNR выполняет операцию «исключающее НЕ-ИЛИ» между исходными матрицами 1 и 2 с количеством строк **n** и сохраняет результат в **D**.
2. Правило операции «исключающее ИЛИ»: результат равен 1, если оба бита одинаковы, и 0, если разные.
3. Если операнды $\mathbf{S_1}$, $\mathbf{S_2}$, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:



Когда X0 включен, команда MXNR проводит операцию «исключающее НЕ-ИЛИ» между 16-битными регистрами D0~D2 и 16-битными регистрами D10~D12. Результат сохраняется в 16-битных регистрах D20~D22.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/ SE	SX2
184	MINV	P	S D n	Инверсия матрицы				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MINV, MINVP: 7 шагов
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

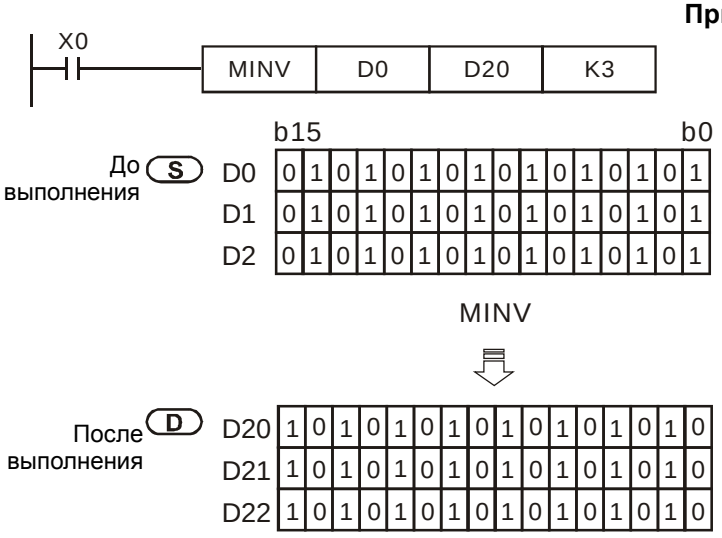
Операнды:

S: Исходная матрица **D**: Результирующая матрица **n**: Число строк матрицы (**n** = K1~K256)

Описание:

1. Команда MINV осуществляет инверсию матрицы (каждого бита) **S** с числом строк **n** , результат сохраняется в **D**.
2. Если операнды $\mathbf{S_1}$, $\mathbf{S_2}$, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:



Когда X0 включен, команда MINV проводит инверсию матрицы с

16-битными регистрами D0~D2. Результат сохраняется в 16-битных регистрах D20~D22.

API	Команда			Операнды	Функция	Контроллеры			
185		MCMP	P	S_1 S_2 n D	Сравнение матриц	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1							*	*	*	*	*	*	*			MCMP, MCMPP: 9 шагов
S_2							*	*	*	*	*	*	*			
n					*	*							*			
D								*	*	*	*	*	*	*	*	

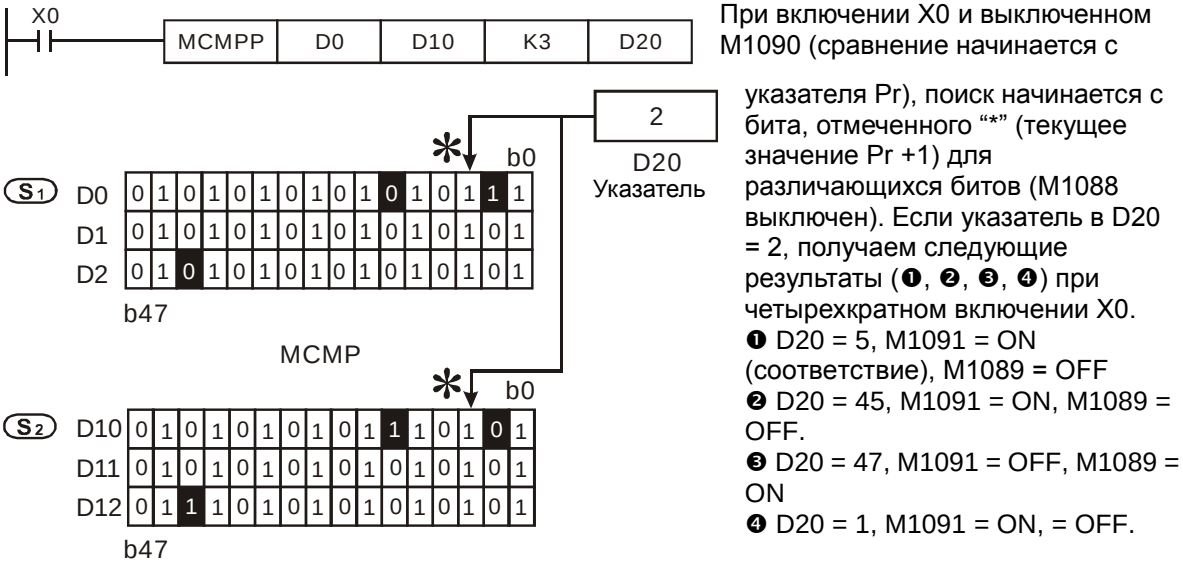
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходная матрица 1 **S_2**: Исходная матрица 2 **n**: Число строк матрицы (**n** = K1~K256)
D: Указатель Pr; результат сравнения (номер бита)

Описание:

1. Команда MCMP сравнивает побитово матрицы **S_1** и **S_2** и сохраняет номер первого отличающегося бита в **D**. При следующем выполнении команды сравнение начнется со следующего после указанного в D бита.
2. Флаг результата сравнения (M1088) включен, если матрицы эквивалентны и выключен, если матрицы различны. После завершения сравнения включается флаг окончания M1091, показывающий, что результат получен (найдено несовпадение битов и результат записан в **D**). После сравнения всех битов включается M1089, и последний бит записывается в **D**. Следующий цикл начинается с первого бита (бит 0), включенный флаг M1090 показывает начало сравнения. Если значение **D** (Pr) выходит за пределы допустимого диапазона, включается M1092,показывающий ошибку и выполнение команды будет прервано.
3. Операциям с матрицами требуется один 16-битный регистр для обозначения одного из 16n битов в матрице – регистр указателя (Pr), назначаемый пользователем. Диапазон Pr: 0 ~ 16n -1, соответствует b0 ~ b16n-1 в матрице. Указатель не должен меняться в ходе выполнения команды во избежание ошибки.
4. При одновременном наступлении событий, показываемых флагами M1089 и M1091, оба флага будут включены.
5. Если операнды **S_1**, **S_2**, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:



При включении X0 и выключенном M1090 (сравнение начинается с указателя Pr), поиск начинается с бита, отмеченного "*" (текущее значение Pr +1) для различающихся битов (M1088 выключен). Если указатель в D20 = 2, получаем следующие результаты (❶, ❷, ❸, ❹) при четырехкратном включении X0.
❶ D20 = 5, M1091 = ON (соответствие), M1089 = OFF
❷ D20 = 45, M1091 = ON, M1089 = OFF.
❸ D20 = 47, M1091 = OFF, M1089 = ON
❹ D20 = 1, M1091 = ON, = OFF.

Заметки: Флаги и регистры:

M1088:	Результат сравнения. Матрицы одинаковы (M1088 вкл.) или различны (M1088 выкл.)
D1089:	Индикация завершения сравнения. При достижении последнего бита, M1089 вкл.
D1090:	Индикация начала сравнения. Когда сравнение начинается с первого бита, M1090 включается
D1091:	Индикация получения результата сравнения. При получении результата M1091 включается
D1092:	Индикация ошибки указателя. Когда указатель Pr выходит за пределы допустимого диапазона, M1092 включается.

API	Команда		Операнды	Функция	Контроллеры			
186	MBRD	P	(S) (n) (D)	Чтение битов в матрице	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBRD, MBRDP: 7 шагов
n					*	*							*			
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

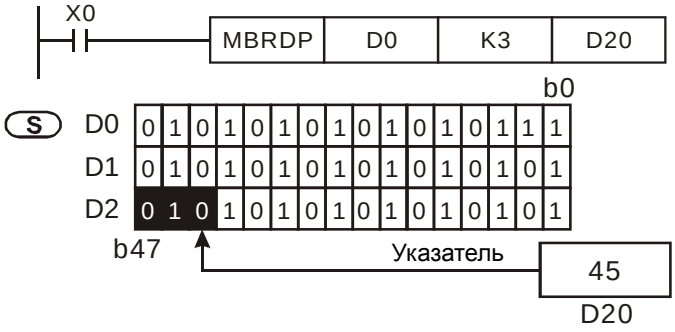
Операнды:

S: Исходная матрица **n**: Число строк матрицы (K1~K256). **D**: Указатель Pr (номер бита)

Описание:

1. Команда MBRD считывает состояние отдельных битов в матрице. При выполнении команды MBRD сначала проверяется состояние флага M1094 (флаг сброса указателя Pr). Если M1094 включен, значение Pr в **D** будет сброшено и команда начинает считывать с первого бита. Состояние битов передается в M1095 (флаг переноса для операций с матрицами). После чтения состояния бита команда проверяет состояние M1093 (флаг увеличения указателя). Если M1093 включен, команда MBRD начинает считывать следующий бит, т.е. значение Pr увеличивается на 1. После чтения последнего бита включается M1089, показывающий завершение процесса, в **D** записывается номер последнего бита. После этого команда останавливается.
2. Операциям с матрицами требуется один 16-битный регистр для обозначения одного из 16n битов в матрице – регистр указателя (Pr), назначаемый пользователем. Диапазон Pr: 0 ~ 16n -1, соответствует b0 ~ b16n-1 в матрице. Указатель не должен меняться в ходе выполнения команды во избежание ошибки.
3. Если операнды $\mathbf{S_1}$, $\mathbf{S_2}$, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:



1. Когда X0 включается при включенных M1094 (сброс значения Pr) и M1093 (увеличение значения Pr), чтение начинается с первого бита и значение Pr увеличивается на 1 после считывания каждого бита.
2. Если D20 = 45, получаем следующие 3 результата (❶, ❷, ❸) при трехкратном включении X0.

a) ❶ D20 = 45, M1095 = OFF, M1089 = OFF

b) ❷ D20 = 46, M1095 = ON, M1089 = OFF.

c) ❸ D20 = 47, M1095 = OFF, M1089 = ON

Заметки: Флаги и регистры:

M1089:	Индикация завершения чтения. При достижении последнего бита, M1089 вкл.
M1092:	Индикация ошибки указателя. Когда указатель Pr выходит за пределы допустимого диапазона, M1092 включается
M1093	Флаг увеличения указателя. Добавление 1 к текущему значению Pr
M1094	Флаг сброса указателя. Сбрасывает текущее значение Pr до 0
M1095	Флаг переноса для матричных операций

API	Команда		Операнды	Функция	Контроллеры			
187	MBWR	P	(S) (n) (D)	Запись битов в матрицу	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBWR, MBWRP: 7 шагов
n					*	*							*			
D								*	*	*	*	*	*	*	*	

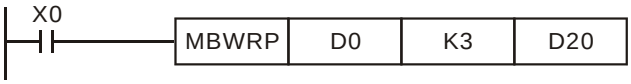
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Исходная матрица **n**: Число строк матрицы (K1~K256). **D**: Указатель Pr (номер бита)

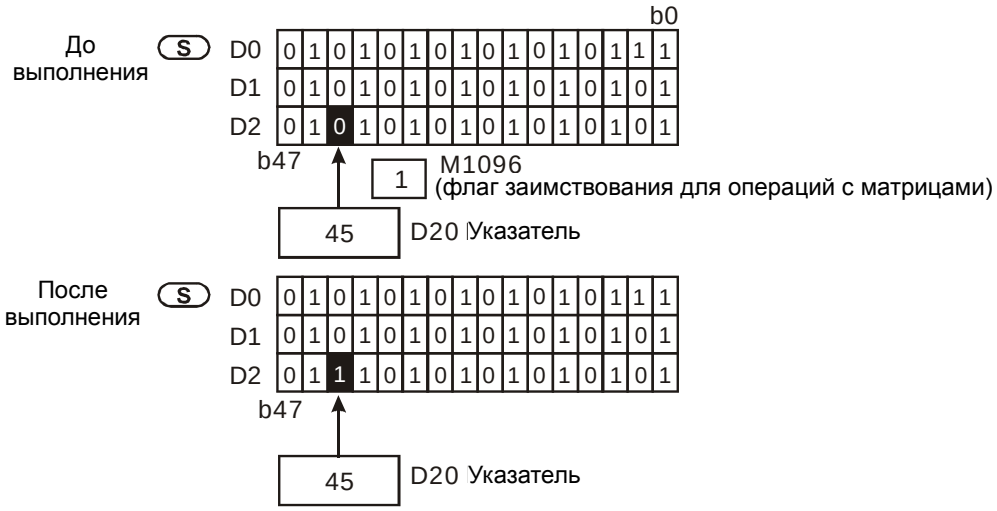
Описание:

1. Команда MBWR записывает состояние отдельных битов в матрицу. При выполнении команды MBWR, сначала проверяется состояние флага M1094 (флаг сброса указателя Pr). Если M1094 включен, значение Pr в **D** будет сброшено и команда начинает записывать матрицу с 0-бита. Состояние (0/1) каждого бита будет зависеть от состояния специального реле M1096 (флаг заимствования для операций с матрицами). После записи состояния бита команда проверяет состояние M1093 (флаг увеличения указателя). Если M1093 включен, команда MBWR начинает записывать следующий бит, т.е. значение Pr увеличивается на 1. После записи последнего бита включается M1089, показывающий завершение процесса, в **D** записывается номер последнего записанного бита. После этого команда останавливается.
2. Операциям с матрицами требуется один 16-битный регистр для обозначения одного из 16n битов в матрице – регистр указателя (Pr), назначаемый пользователем. Диапазон Pr: 0 ~ 16n -1, соответствует b0 ~ b16n-1 в матрице. Указатель не должен меняться в ходе выполнения команды во избежание ошибки.
3. Если операнды $\mathbf{S_1}$, $\mathbf{S_2}$, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.

Пример программы:



1. Когда X0 вкл. при M1094 (сброс значения Pr) и M1093 (увеличение значения Pr) равных 1, запись начинается с нулевого бита и значение Pr увеличивается на 1 после записи каждого бита.
2. Если указатель D20 = 45 и M1096 включен, получаем следующий результат при однократном включении X0.



Заметки:
Флаги и регистры:

M1089:	Индикация завершения записи. При достижении последнего бита, M1089 вкл.
M1092:	Индикация ошибки указателя. Когда указатель Pr выходит за пределы допустимого диапазона, M1092 включается
M1093	Флаг увеличения указателя. Добавление 1 к текущему значению Pr
M1094	Флаг сброса указателя. Сбрасывает текущее значение Pr до 0
M1096	Флаг заимствования для матричных операций

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
188		MBS	P	S D n	Сдвиг битов в матрице				

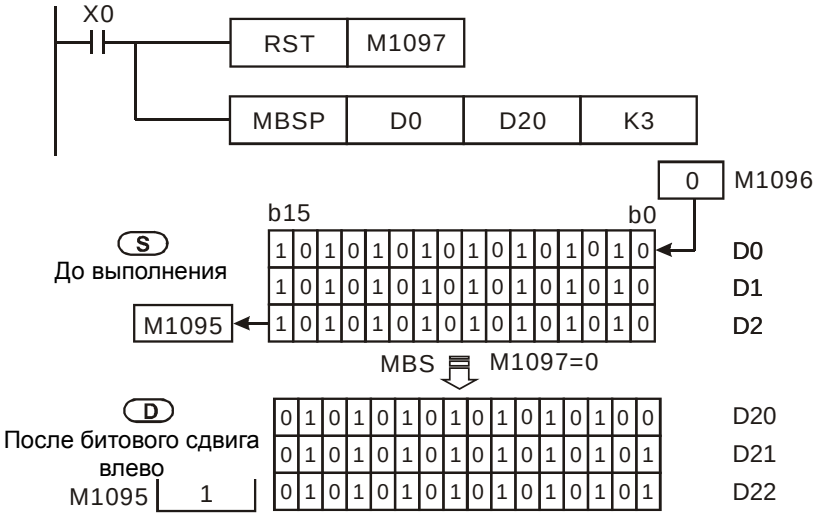
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBS, MBSP: 7 шагов
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Исходная матрица **D**: Результирующая матрица **n**: Число строк в матрице (K1~K256)

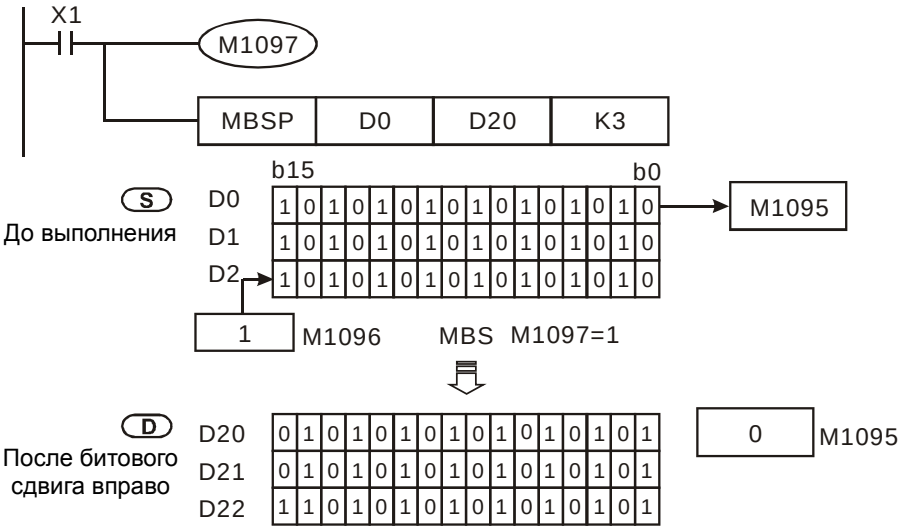
Описание:
1. Команда MBS сдвигает биты в матрице влево или вправо. Если M1097 выключен, биты сдвигаются влево, M1097 включен - вправо. Пустой бит (левый сдвиг: b0; правый сдвиг: b16**n**-1) после каждого сдвига будет заполняться значением M1096 (флаг заимствования для операций с матрицами). Бит, смещающийся из матрицы (левый сдвиг: b16**n**-1; правый сдвиг: b0) будет направлен в M1095 (флаг переноса для операций с матрицами) и результат операции сохраняется в **D**.
2. Как правило, применяется импульсная команда (MBSP).
3. Если операнды $\mathbf{S_1}$, $\mathbf{S_2}$, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.
4. M1095: Флаг переноса для матричных операций; M1096: Флаг заимствования для матричных операций; M1097: Флаг направления для матричных операций

Пример программы 1:



Когда X0 вкл. и M1097 выкл. биты сдвигаются влево. Если флаг заимствования M1096 выкл. и 16-битные регистры D0 ~ D2 сдвигаются влево, результат будет сохранен в 16-битных регистрах D20 ~ D22, а флаг переноса M1095 будет включен.

Пример программы 2:



огда X1 и M1097 вкл., происходит сдвиг битов вправо. Если флаг заимствования M1096 включен и 16-битные регистры D0 ~ D2 сдвигаются вправо, результат будет сохранен в 16-битных регистрах D20 ~ D22, а флаг переноса M1095 будет выключен.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
189	MBR	P	S D n	Вращение битов в матрице				

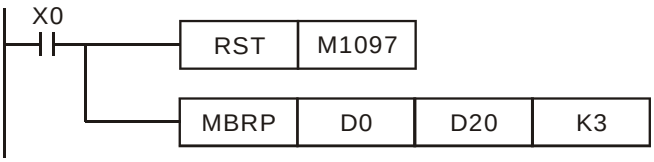
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBR, MBRP: 7 шагов
D								*	*	*	*	*	*			
n					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

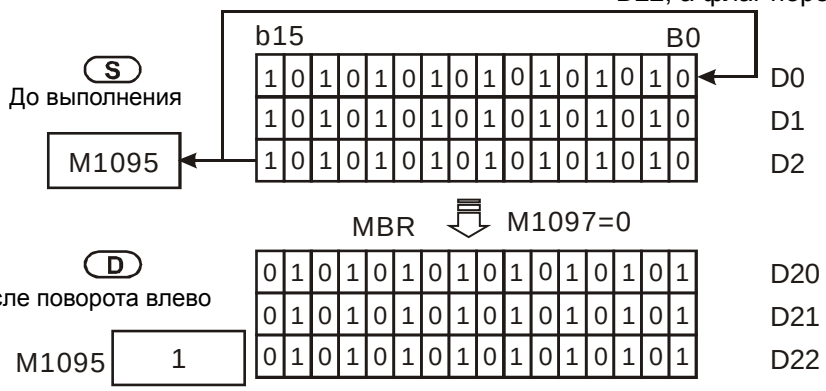
Операнды:
S: Исходная матрица **D**: Результирующая матрица **n**: Число строк в матрице (K1~K256)

Описание:
1. Команда MBR вращает (кольцевой сдвиг) биты в матрице влево или вправо. Если M1097 выключен, биты вращаются влево, M1097 включен - вправо. Пустой бит (левый сдвиг: b0; правый сдвиг: b16**n**-1) после каждого поворота будет заполнен битом, вращающимся из матрицы (левый сдвиг: b16**n**-1; правый сдвиг: b0), результат операции будет сохранен в **D**. Кроме того, вращаемые из матрицы биты также помещаются в M1095 (флаг переноса для операций с матрицами).
2. Как правило, применяется импульсная команда (MBRP).
3. Если операнды S_1, S_2, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.
4. Флаги для матричных операций: M1095: Флаг переноса; M1097: Флаг направления

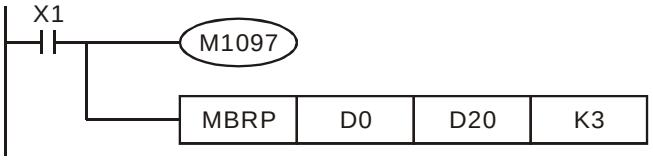
Пример программы 1:



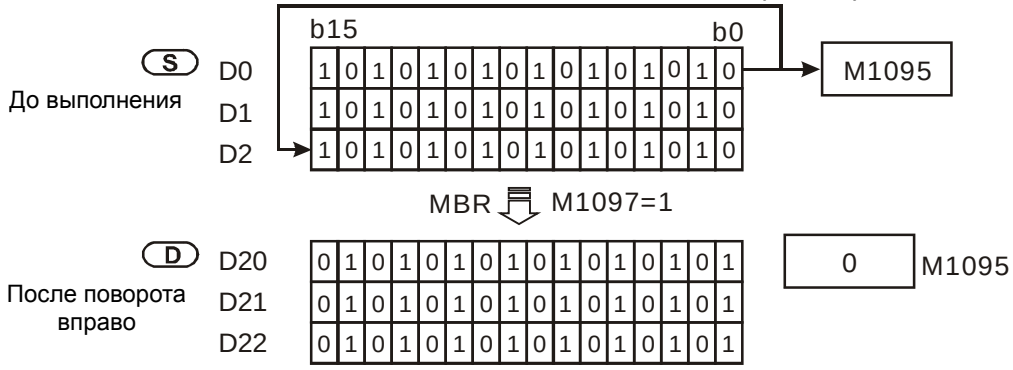
Когда X0 включен и M1097 выключен, выполняется вращение битов влево. 16-битные регистры D0 ~ D2 вращаются влево, результат будет сохранен в 16-битных регистрах D20 ~ D22, а флаг переноса M1095 будет вкл.



Пример программы 2:



Когда X0 и M1097 оба включены, выполняется вращение битов вправо. 16-битные регистры D0 ~ D2 вращаются вправо, результат будет сохранен в 16-битных регистрах D20 ~ D22, а флаг переноса M1095 выкл.



API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
190		MBC	P	S n D	Счетчик битов в матрице				

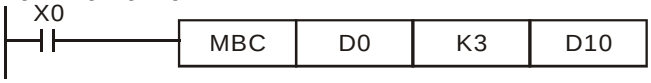
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S							*	*	*	*	*	*	*			MBC, MBCP: 7 шагов
n					*	*							*			
D								*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Исходная матрица **n**: Число строк в матрице (K1~K256) **D**: Результат

Описание:
1. Команда MBC подсчитывает число единичных или нулевых битов в матрице из **n** строк и сохраняет результат в **D**.
2. Если операнды S_1, S_2, **D** имеют формат KnX, KnY, KnM, KnS, применимо только **n** = 4.
3. Когда M1098 включен, команда MBC подсчитывает единичные биты. M1098 выключен – нулевые биты. Если результат равен 0, включается флаг M1099.
4. M1098: Подсчет числа битов со значениями “1” или “0”; M1099: Результат равен “0”.

Пример программы:



Когда X0 и M1098 оба включены, команда MBC подсчитывает число единичных битов в D0~D2 и сохраняет результат в D10. Когда X0 включен, а M1098 выключен, команда подсчитывает число нулевых битов D0~D2 и сохраняет результат в D10.

D0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1
D2	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1

D10 | 12 | M1098=0

D10 | 36 | M1098=1

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
191	D	PPMR		S_1 S_2 S D	Двухосевая относительная синхронизация по двум точкам				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DPPMR: 17 шагов
S_1					*	*							*			
S_2					*	*							*			
S					*	*							*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Количество выходных импульсов по оси X S_2: Количество выходных импульсов по оси Y
S: Максимальная выходная частота при перемещении «из точки в точку» **D**: Импульсный выход

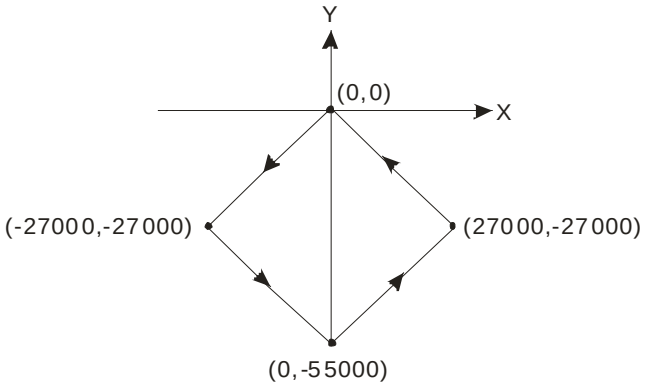
Описание:
1. Для типов ES2/EX2: функция поддерживается V1.20 или выше.
2. Команда поддерживает только импульсные выходы типа: Pulse/Dir.
3. S_1 и S_2 определяют количество выходных импульсов (относительное позиционирование) по оси X (Y0) и оси Y (Y2). Диапазон: -2,147,483,648 ~ +2,147,483,647 (знак “+/-“ показывает прямое/обратное направление перемещения). Для прямого направления: текущее значение на импульсном выходе CH0 (D1031 старшее слово, D1030 младшее слово), CH1 (D1337 старшее слово, D1336 младшее слово) увеличивается. Для обратного направления - уменьшается.
4. **S**: если максимальная выходная частота больше или меньше 100 Гц, выход будет работать с частотой 100 Гц.
5. **D** может быть только Y0.
   Y0 – импульсный выход для оси X;
   Y1 – направление перемещения по оси X (OFF: положительное; ON: отрицательное)
   Y2 – импульсный выход для оси Y;
   Y3 – направление перемещения по оси Y (OFF: положительное; ON: отрицательное)
   После обработки импульсного выхода сигнал направления не будет отключен при отключении условия, запустившего команду.
6. D1340 – заданная начальная/конечная частота осей X/Y. Если заданное значение меньше 6 Гц, ПЛК принимает в качестве заданной величины 6 Гц. D1343 – заданное

время рампы разгона/торможения осей X/Y. Если время меньше 20 мс, устанавливается время 20 мс. По умолчанию: 100 мс.

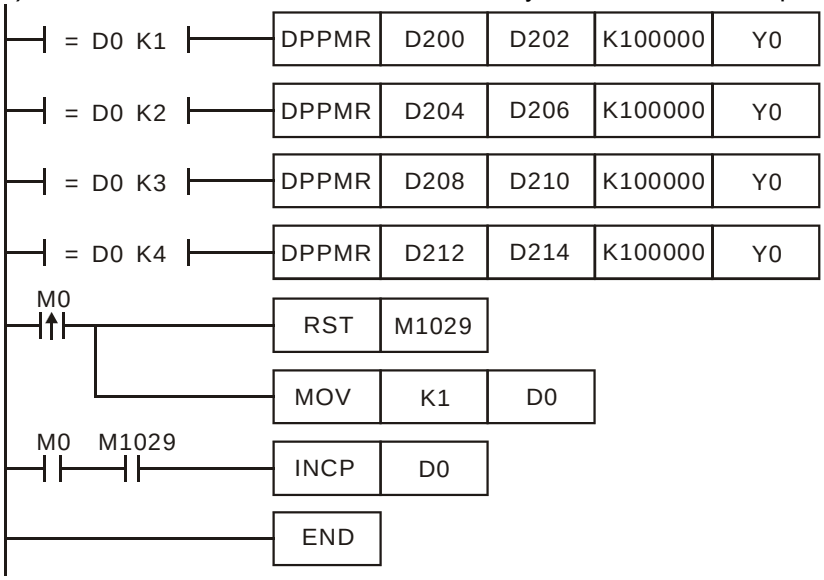
7. Во время работы команды PPMR начальная частота и время разгона/торможения по оси Y будут аналогичны настройкам по оси X. Кроме того, индивидуальная установка времени замедления D1348/D1349 не рекомендуется, т.к. это может привести к несоответствию осей X и Y. Флаги «паузы импульсного выхода» не применяются. Для прекращения выдачи выходных импульсов необходимо отключить условие, запустившее команду.
8. Для импульсного выхода с рамповой функцией – если число выходных импульсов задано только на одной оси (т.е., на второй – 0), импульсы будут генерироваться только на этой оси. Если количество импульсов меньше 20, то рамповая функция будет отключена, и импульсный выход будет работать на частоте 3 кГц.
9. Не существует ограничений на количество одновременно работающих команд, но если CH0, CH1 или импульсный выход уже используются, оси X и Y синхронизированы не будут.
10. M1029 включается, когда 2-х осевая синхронизация импульсного выхода выполнена.

Пример программы:

1. Нарисуем ромб, см. ниже.



2. Шаги:
   a) Задаем 4 координаты (0,0), (-27000, -27000), (0, -55000), (27000, -27000) (на рис. выше). Рассчитываем относительные координаты 4-х точек и получаем (-27000, -27000), (27000, -28000), (27000, 27000) и (-27000, 27000). Помещаем их в 32-битные регистры (D200, D202), (D204, D206), (D208, D210), (D212, D214).
   b) Составляем программу.
   c) Включаем ПЛК. Включаем M0 для запуска 2-х осевой синхронизации.



3. Работа:
   При включенном M0, начинается первое движение «из точки в точку» с частотой 100 кГц. D0 увеличится на 1 после завершения первого движения «из точки в точку», второе движение начнется автоматически. Операция будет повторяться до завершения четвертого движения «из точки в точку».

Заметки:

Флаги и регистры:

M1029:	Завершение работы импульсного выхода CH0 (Y0, Y1)
D1030:	Число выходных импульсов на Y0 (старшее слово).
D1031:	Число выходных импульсов на Y1 (младшее слово).
D1336:	Число выходных импульсов на Y2 (старшее слово)
D1337:	Число выходных импульсов на Y2 (младшее слово)
D1340:	Начальная/конечная частота импульсного выхода CH0 (Y0), CH1(Y2) для команд DPPMR/DPPMA
D1343:	Время рампы разгона/замедления импульсного выхода CH0 (Y0), CH1(Y2) для команд DPPMR/DPPMA

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
192	D	PPMA		S_1 S_2 S D	Двухосевая абсолютная синхронизация по двум точкам				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DPPMA: 17 шагов
S_2					*	*							*			
S					*	*							*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

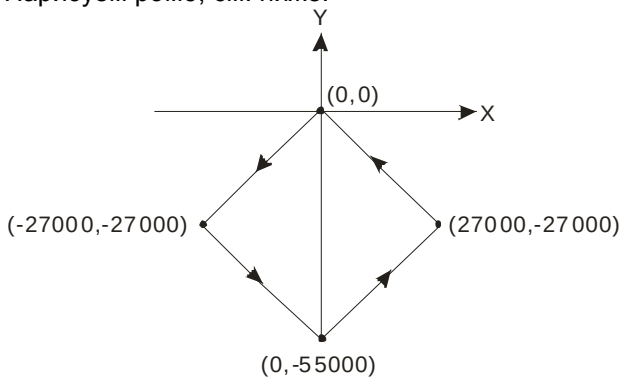
S_1: Количество выходных импульсов по оси X **S_2**: Количество выходных импульсов по оси Y
S: Макс. выходная частота при перемещении «из точки в точку» **D**: Импульсный выход

Описание:

1. Для типов ES2/EX2: функция поддерживается V1.20 или выше.
2. Команда поддерживает только импульсные выходы типа: Pulse/Dir.
3. **S_1** и **S_2** определяют количество выходных импульсов (абсолютное позиционирование) по оси X (Y0) и оси Y (Y2). Диапазон: -2,147,483,648 ~ +2,147,483,647 (знак “+/-“ показывает прямое/обратное направление перемещения). Для прямого направления: текущее значение на импульсном выходе CH0 (D1031 старшее слово, D1030 младшее слово), CH1 (D1337 старшее слово, D1336 младшее слово) увеличивается. Для обратного направления - уменьшается.
4. **D** может быть только Y0.
   Y0 – импульсный выход для оси X;
   Y1 – направление перемещения по оси X (OFF: положительное; ON: отрицательное)
   Y2 – импульсный выход для оси Y;
   Y3 – направление перемещения по оси Y (OFF: положительное; ON: отрицательное)
5. Описание флагов и регистров см. API 191 команда DPPMR.

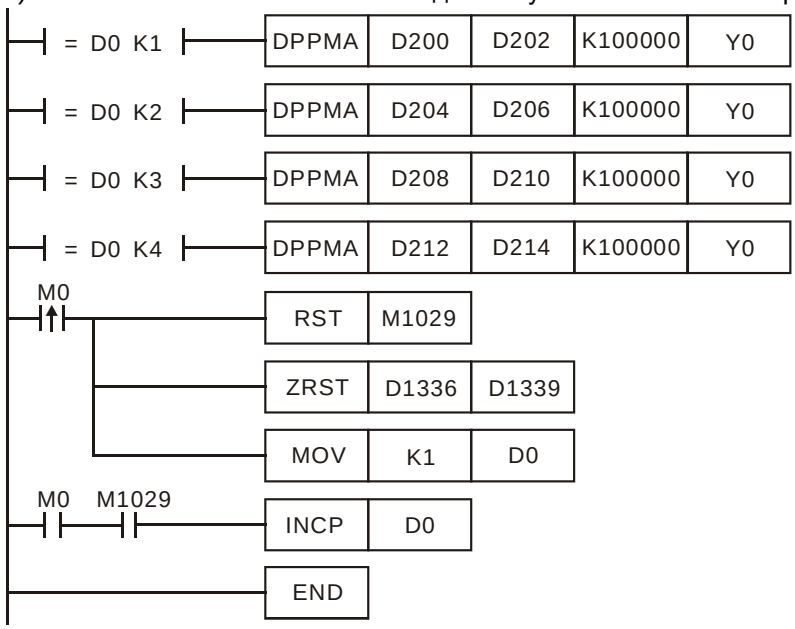
Пример программы:

1. Нарисуем ромб, см. ниже.



2. Шаги:
   a) Задаем 4 координаты (-27000, -27000), (0, -55000), (27000, -27000) и (0,0) (см. рис.). Помещаем их в 32-битные регистры (D200, D202), (D204, D206), (D208, D210), (D212, D214).

b) Составляем программу.
c) Включаем ПЛК. Включаем M0 для запуска 2-х осевой синхронизации.



3. Работа:
При включенном M0 начинается первое движение «из точки в точку» с частотой 100 кГц. D0 увеличится на 1 после завершения первого движения «из точки в точку», второе движение начнется автоматически. Операция будет повторяться до завершения четвертого движения «из точки в точку».

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
193	D	CIMR		S_1 S_2 S D	Двухосная относительная дуговая интерполяция				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			DCIMR: 17 шагов
S_2					*	*							*			
S													*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Количество выходных импульсов по оси X **S_2**: Количество выходных импульсов по оси Y
S: Задания параметров **D:** Импульсный выход

Описание:
1. Для типов ES2/EX2: функция поддерживается V1.20 или выше.
2. Команда поддерживает только импульсные выходы типа: Pulse/Dir.
3. **S_1** и **S_2** определяют количество выходных импульсов (относительное позиционирование) по оси X (Y0) и оси Y (Y2). Диапазон: -2,147,483,648 ~ +2,147,483,647 (знак “+/-“ показывает прямое/обратное направление перемещения). Для прямого направления: текущее значение на импульсном выходе CH0 (D1031 старшее слово, D1030 младшее слово), CH1 (D1337 старшее слово, D1336 младшее слово) увеличивается. Для обратного направления - уменьшается.
4. Младшее слово **S** (настройки направления и дискретизация): K0 – 20-сегментный выход по часовой стрелке; K1 – 20-сегментный выход против часовой стрелке; A – 90° дуга (см. рис. 1 и 2).
5. Старшее слово **S** (настройка времени движения, шаг: 0.1 сек): Диапазон: K2 ~ K200 (0.2 сек. ~ 20 сек.). Команда ограничивает максимальную частоту выходных импульсов, поэтому когда установленное время меньше реально необходимого времени, происходит корректировка.

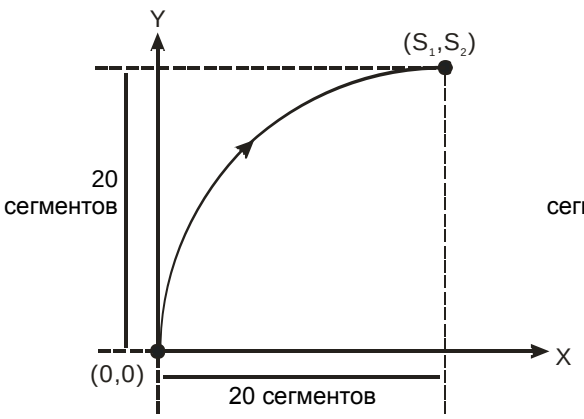


Рисунок 1

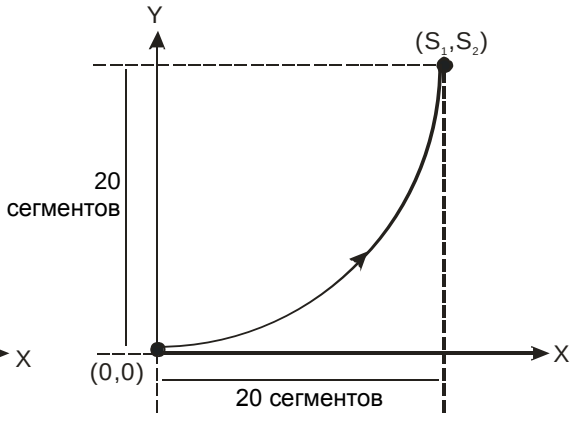


Рисунок 2

a. Изобразим четыре 90° дуги (см. рис. ниже).
При подаче управляющего сигнала, направление положительное (QI, QIV). При отсутствии управляющего сигнала, направление отрицательно (QII, QIII). Когда S определен как K0, дуга – по часовой стрелке (см. рис 3). Когда S определен как K, дуга – против часовой стрелки (см. рис 4).

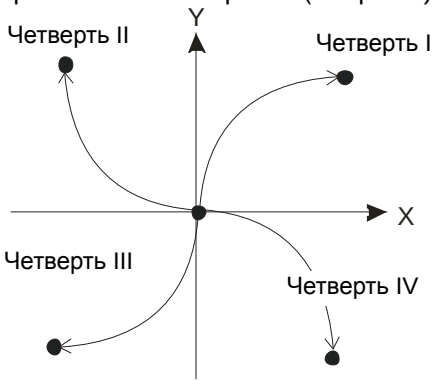


Рисунок 3

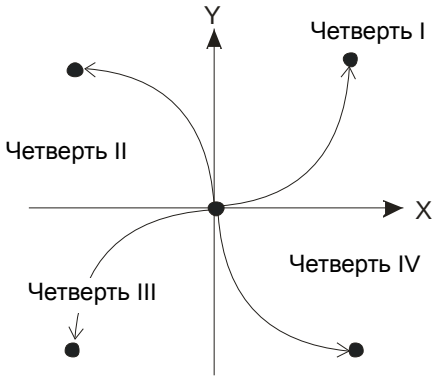


Рисунок 4

b. настройки направления и дискретизации в младшем слове **S** могут быть только K0 ~ K1
c. настройка времени движения в старшем слове **S** не может быть меньше минимально предложенного времени. Если время движения не указано, ПЛК будет использовать минимально предложенное время из настроек. См. таблицу.

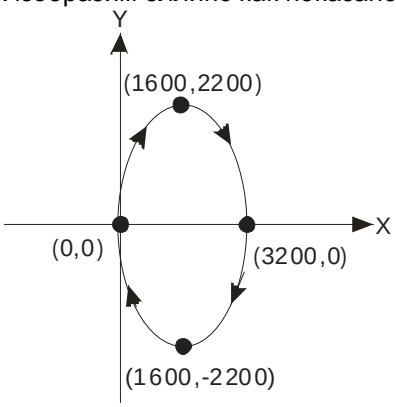
Сегменты	Макс. конечная позиция (импульсы)	Настройка минимального предложенного времени (шаг:100 мс)
20-сегментное разрешение	500 ~ 20,000	2
	20,000 ~ 29,999	3
	:	:
	менее 10,000,000	менее 200

D может быть только Y0.
Y0 – импульсный выход для оси X;
Y1 – направление перемещения по оси X (OFF: положительное; ON: отрицательное)
Y2 – импульсный выход для оси Y;
Y3 – направление перемещения по оси Y (OFF: положительное; ON: отрицательное)
После обработки импульсного выхода сигнал направления не будет отключен при отключении условия, запустившего команду.

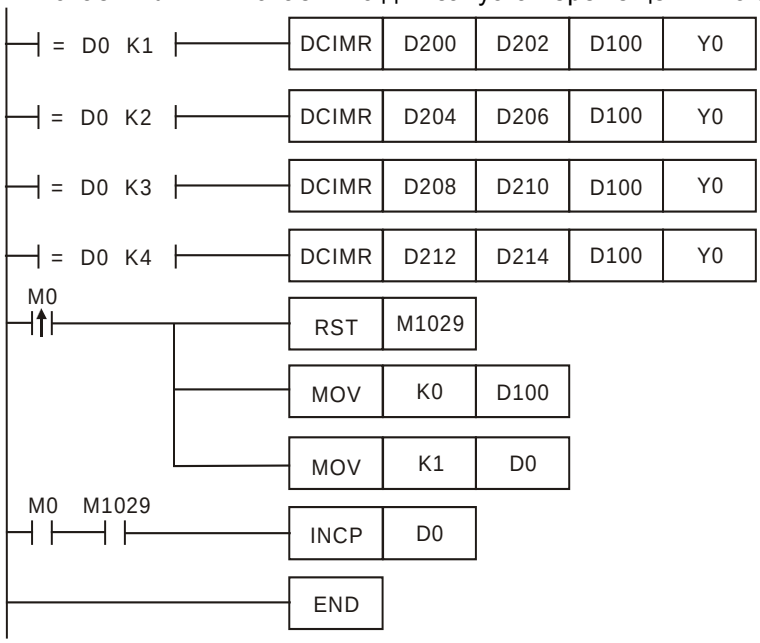
d. Выполнение 2-осевой интерполяции в 20 сегментах занимает около 2 мс. Если количество импульсов на выходе задано только для одной оси, ПЛК выполнит позиционирование только этой оси в соответствии с заданным временем. Если для одной из осей задано число импульсов менее 500, ПЛК будет выполнять 2-осевую интерполяцию автоматически. Если хотя бы одной из осей задается число импульсов выше 10 000 000, команда работать не будет.
e. При каждом выполнении команды может быть выполнена только одна 90° дуга. Перемещение при этом может быть и не 90°, т.е. количество заданных импульсов на осях X и Y может быть различным.
f. Существуют настройки стартовой частоты и времени рампы по умолчанию.
g. Не существует ограничений на количество одновременно работающих команд, но если CH0, CH1 или импульсный выход уже используются, оси X и Y синхронизированы не будут.

Пример программы 1:

1. Изобразим эллипс как показано ниже.



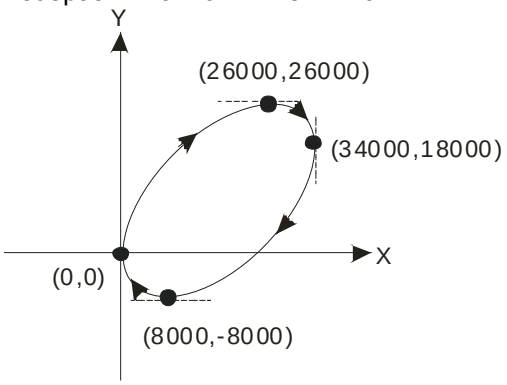
2. Шаги:
   a) Задаем 4 координаты (0,0), (1600, 2200), (3200, 0), (1600, -2200) (см. рис.). Рассчитаем относительные координаты 4-х точек и получаем (1600, 2200), (1600, -2200), (-1600, -2200) и (-1600, 2200). Поместим их в 32-битные регистры (D200, D202), (D204, D206), (D208, D210), (D212, D214).
   b) Задаем “дуга по часовой стрелке”, “время перемещения” по умолчанию (**S**=D100=K0).
   c) Включаем ПЛК. Включаем M0 для запуска перемещения по эллипсу.



3. Работа: При включенном M0 ПЛК начнет перемещение по первому сектору дуги. D0 увеличится на 1 после завершения движения в первом секторе, второе движение начнется автоматически. Операция будет повторяться до завершения движения по четвертому сектору.

Пример программы 2:

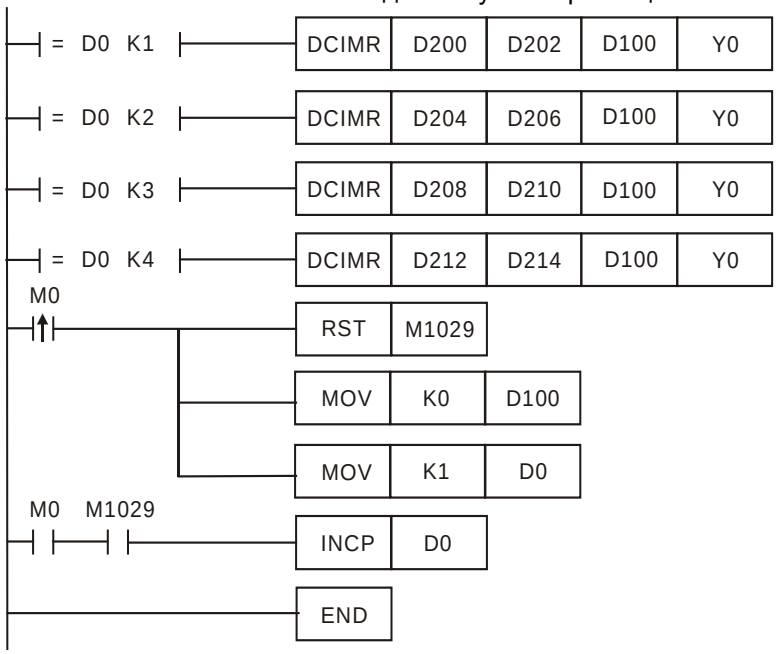
1. Изобразим наклонный эллипс.



2. Шаги:
   a) Найдем максимальные и минимальные координаты на осях X и Y (0,0), (26000,26000), (34000,18000), (8000,-8000) (см. рис.). Рассчитаем относительные координаты 4-х

точек и получаем (26000,26000), (8000,-8000), (-26000,-26000), (-8000,8000). Поместим их в 32-битные регистры (D200,D202), (D204,D206), (D208,D210) и (D212,D214).

a) Задаем “дуга по часовой стрелке”, “время перемещения” по умолчанию (**S**=D100=K0).
b) Включаем ПЛК. Включаем M0 для запуска перемещения по наклонному эллипсу.



3. Работа: При включенном M0 ПЛК начнет перемещение по первому сектору дуги. D0 увеличится на 1 после завершения движения в первом секторе, второе движение начнется автоматически. Операция будет повторяться до завершения движения по четвертому сектору.

Заметки: Флаги и регистры:

M1029: Завершение работы импульсного выхода CH0 (Y0, Y1)
D1030: Число выходных импульсов на Y0 (старшее слово).
D1031: Число выходных импульсов на Y1 (младшее слово).
D1336: Число выходных импульсов на Y2 (старшее слово)
D1337: Число выходных импульсов на Y2 (младшее слово)

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
194	D CIMA		S_1 S_2 S D	Двухосная абсолютная дуговая интерполяция				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCIMA: 17 шагов
S_1					*	*							*			
S_2					*	*							*			
S													*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Количество выходных импульсов по оси X S_2: Количество выходных импульсов по оси Y
S: Задания параметров **D:** Импульсный выход

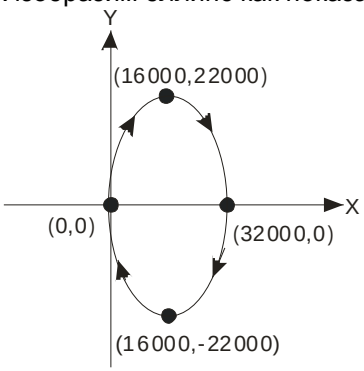
Описание:

1. Для типов ES2/EX2: функция поддерживается V1.20 или выше.
2. Команда поддерживает только импульсные выхода типа: Pulse/Dir.
3. S_1 и S_2 определяют количество выходных импульсов (относительное позиционирование) по оси X (Y0) и оси Y (Y2). Диапазон: -2,147,483,648 ~ +2,147,483,647. Когда S_1 и S_2 больше заданного значения на выходе CH0 (D1031 старшее слово, D1030 младшее слово)/CH1 (D1337 старшее слово, D1336 младшее слово), импульсный выход будет реализовывать положительное направление движения, а сигналы направления выходов Y1, Y3 будут отключены. Когда S_1 и S_2 меньше заданного значения, импульсный выход будет реализовывать отрицательное направление движения, а сигналы направления выходов Y1, Y3 будут включены.

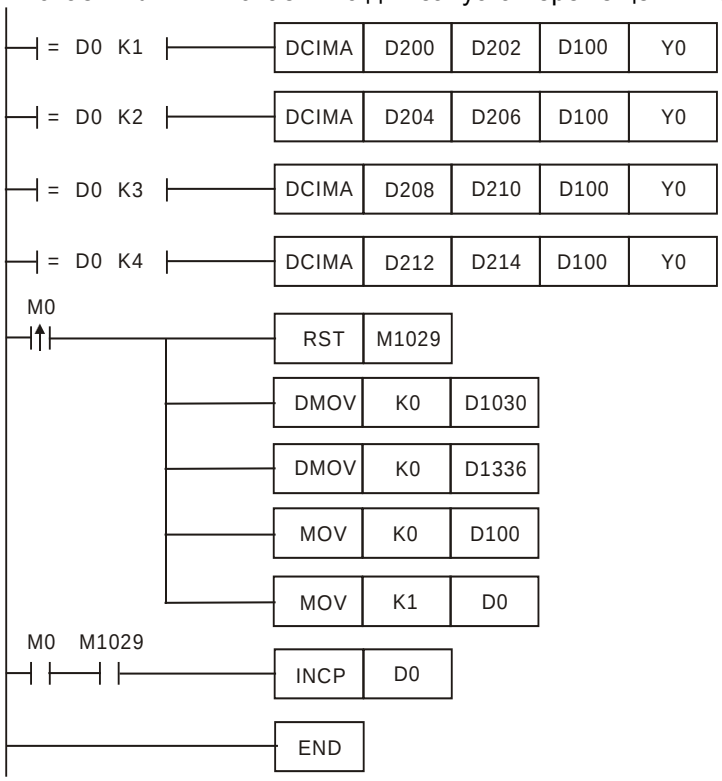
4. Описания флагов и специальных регистров см. API 193 команда DCIMR.

Пример программы 1:

1. Изобразим эллипс как показано ниже.



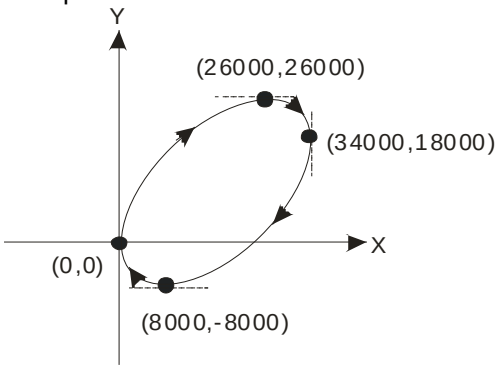
2. Шаги:
   a) Задаем 4 координаты (0,0), (16000, 22000), (32000, 0), (16000, -22000) (см. рис.). Поместим их в 32-битные регистры (D200, D202), (D204, D206), (D208, D210), (D212, D214).
   b) Задаем “дугу по часовой стрелке”, “время перемещения” по умолчанию (**S**=D100=K0).
   b) Включаем ПЛК. Включаем M0 для запуска перемещения по эллипсу.



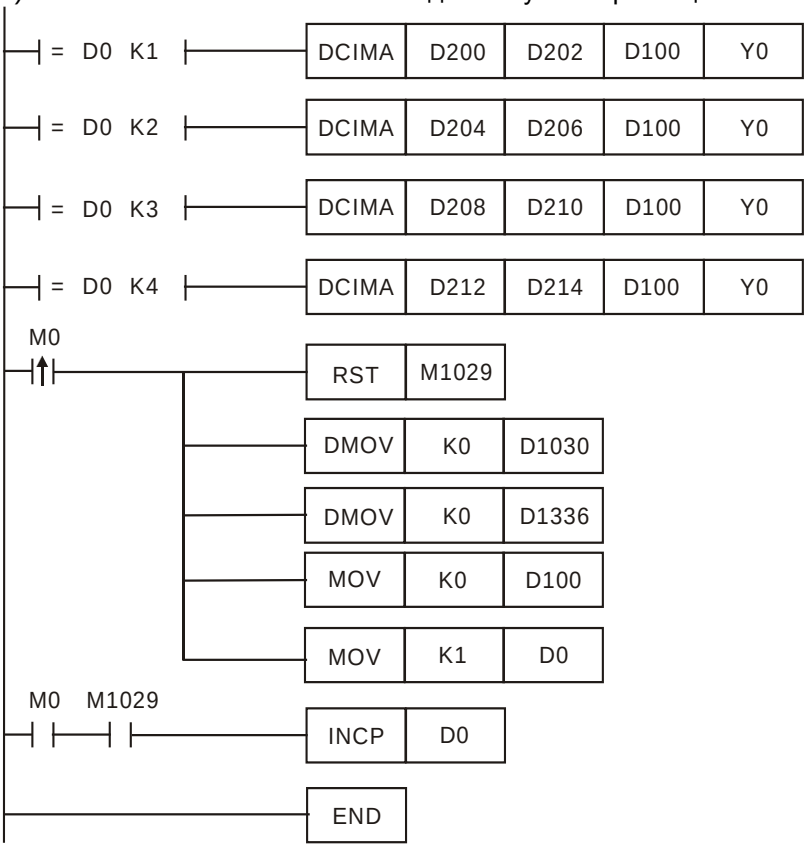
3. Работа: При включенном M0 ПЛК начнет перемещение по первому сектору дуги. D0 увеличится на 1 после завершения движения в первом секторе, второе движение начнется автоматически. Операция будет повторяться до завершения движения по четвертому сектору.

Пример программы 2:

1. Изобразим наклонный эллипс.



2. Шаги:
   a) Найдем максимальные и минимальные координаты на осях X и Y (0,0), (26000,26000), (34000,18000), (8000,-8000) (см. рис.). Поместим их в 32-битные регистры (D200,D202), (D204,D206), (D208,D210) и (D212,D214).
   c) Задаем “дугу по часовой стрелке”, “время перемещения” по умолчанию (**S**=D100=K0).
   b) Включаем ПЛК. Включаем M0 для запуска перемещения по наклонному эллипсу.



3. Работа: При включенном M0 ПЛК начнет перемещение по первому сектору дуги. D0 увеличится на 1 после завершения движения в первом секторе, второе движение начнется автоматически. Операция будет повторяться до завершения движения по четвертому сектору.

API	Команда	Операнды	Функция	Контроллеры
195	D PTPO	S_1 S_2 D	Таблица одноосевого импульсного выхода	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DPTPO: 13 шагов
S_1													*			
S_2													*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Источник данных **S_2**: Количество сегментов **D**: Импульсный выход

Описание:

1. **S_1** определяет выходную частоту и количество импульсов в соответствии с количеством сегментов, установленных в **S_2**. Каждый сегмент занимает 4 последовательных регистра **S_1**. В (**S_1**+0) и (**S_1**+1) сохраняется выходная частота; в (**S_1**+2) и (**S_1**+3) сохраняется количество импульсов.
2. Диапазон выходных частот для **S_1**: 6 Гц~100 000 Гц.
3. **S_2** + 0: общее количество сегментов (диапазон: 1 ~ 40). **S_2** + 1: Номер текущего сегмента. Число в **S_2** + 1 обновляется каждый цикл работы команды.
4. **D** поддерживает только импульсные выходы (Y0 и Y2), пользователь должен применять другие команды в случае, когда контроль направления выходного сигнала не требуется.
5. Эта команда не поддерживает рамповую функцию, поэтому при остановке команды, выдача выходных импульсов немедленно прекращается.

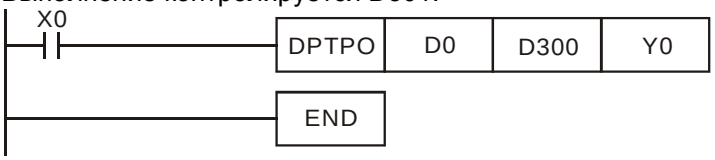
6. В течение одного цикла на каждом выходном канале может быть применена команда PTPO только один раз.
7. После начала работы команды изменение параметров невозможно.
8. Цикличная работа может быть осуществлена включением флага M1262.

Пример программы:

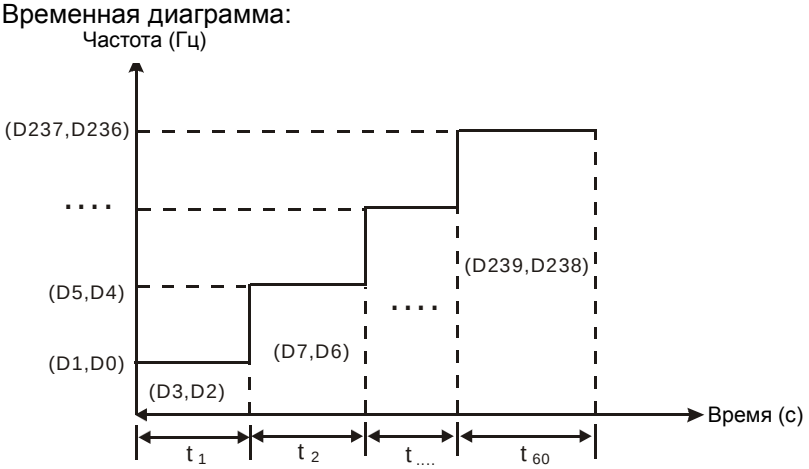
1. Когда M0 включен, импульсный выход будет функционировать в соответствии с заданной частотой и количеством импульсов.
2. Формат таблицы:

S_2 = D300, кол-во сегментов (D300 = K60)	S_1 = D0, частота (S_1 + 0)	S_1 = D0, кол-во выходных импульсов (S_1 + 2)
K1 (1й сегмент)	D1, D0	D3, D2
K2 (2й сегмент)	D5, D4	D7, D6
:	:	:
K60 (60й сегмент)	D237, D236	D239, D238

3. Выполнение контролируется D301.



4. Временная диаграмма:



Заметки:

1. Флаги:
   - M1029 выдача импульсов завершена на CH0 (Y0).
   - M1102 выдача импульсов завершена на CH1 (Y2)
   - M1078 выдача импульсов приостановлена на CH0 (Y0)
   - M1104 выдача импульсов приостановлена на CH1 (Y2)
   - M1262 Включение цикличной работы для табличной функции команды DPTPO ON = включение.
   - M1538 Индикация паузы Y0
   - M1540 Индикация паузы Y2
2. Специальные регистры:
   - D1030 Младшее слово заданного значения импульсного выхода Y0
   - D1031 Старшее слово заданного значения импульсного выхода Y0
   - D1336 Младшее слово заданного значения импульсного выхода Y2
   - D1337 Старшее слово заданного значения импульсного выхода Y2

API	Команда			Операнды	Функция	Контроллеры
197	D	CLLM		S_1 S_2 S_3 D	Управление замкнутой системой позиционирования	ES2/EX2 SS2 SA2/SE SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DCLLM: 17 шагов
S_1	*											*				
S_2					*	*							*			
S_3					*	*							*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Источник данных обратной связи **S_2**: Заданное количество импульсов обратной связи **S_3**: Заданная выходная частота **D**: Импульсный выход

Описание:

1. Указатели прерывания, соответствующие **S_1**:

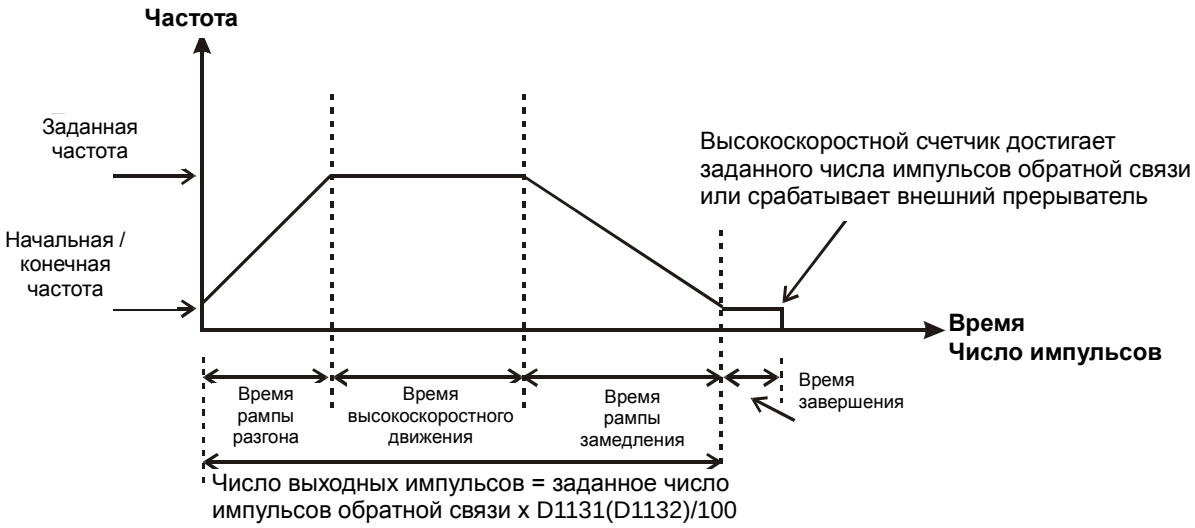
Источник	X4	X6	C243 ~ C254	
Связанный выход	Y0	Y2	Y0	Y2
№ прерывателя	I40□	I60□	I010	I030

□ = 1: срабатывание по переднему фронту; □ = 0: срабатывание по заднему фронту

   a) Когда **S_1** является входом X и значение на импульсном входе достигает заданного в **S_2** количества импульсов обратной связи, выход будет работать на конечной частоте до появления прерывания на входе X.
   b) Когда **S_1** является высокоскоростным счетчиком и количество импульсов на выходе достигает заданного количества импульсов обратной связи, выход будет работать на конечной частоте до того, как количество импульсов обратной связи достигнет заданного в **S_2** количества.
   c) **S_1** может быть высокоскоростным счетчиком C или входом X с внешним прерывателем. Если **S_1** это C, команда DCNT должна быть выполнена до включения функции высокоскоростного счета, а команда EI с I0x0 должна быть запущена для внешних прерывателей. Если **S_1** это X, команда EI с I0x0 должна быть запущена для внешних прерывателей.
   d) Если **S_1** со счетчиками, команда DHSCS должна быть запрограммирована в программе пользователя. См. **Пример программы 2**.
2. Диапазон **S_2**: -2,147,483,648 ~ +2,147,483,647 (+ / - показывает положительное/отрицательное направление движения). Заданное значение импульсного выхода в CH0 (Y0, Y1) и CH1 (Y2, Y3) увеличивается при положительном направлении и уменьшается при отрицательном. Регистры, сохраняющие значение на импульсном выходе: CH0 (D1031 старший байт, 1030 младший байт), CH1(D1337 старший байт, D1336 младший байт)
3. Если **S_3** меньше 6 Гц, выход работает на 6 Гц; если больше 100 кГц – на 100 кГц.
4. **D** может быть назначен как Y0 (сигнал направления: Y1) или Y2 (сигнал направления: Y3). Сигнал направления может быть выключен, только если выключен контакт, т.е. окончание работы импульсного выхода не сбрасывает Y1 или Y3.
5. D1340 и D1352 сохраняют начальную/конечную частоту CH0 и CH1. Мин. 6 Гц, по умолчанию: 100 Гц.
6. D1343 и D1353 время рампы разгона/замедления CH0 и CH1. Если это время меньше 20 мс, ПЛК работает с 20 мс. По умолчанию: 100 мс.
7. Время рампы замедления в CH0 и CH1 может быть указано с помощью установки (M1534, D1348) и (M1535, D1349). Когда M1534 / M1535 включены, время рампы замедления в CH0 и CH1 устанавливается в D1348 и D1349.
8. D1131 и D1132 коэффициент масштабирования (отношение выхода/входа (в %)) замкнутого контура управления CH0 и CH1. K1 это 1 импульс из 100 импульсов обратной связи; K200 это 200 импульсов из 100 импульсов обратной связи. Процентное соотношение, установленное в D1131 и D1132, представляет собой деление количества выходных импульсов (диапазон: K1 ~ K10 000) на фиксированное количество импульсов обратной связи K100 (системная настройка).
9. M1305 и M1306 могут менять направление сигнала импульсных выходов CH0, CH1. Если сигнал направления движения (Y1/Y3) выключен, импульсный выход обеспечивает положительное направление движения. Если M1305/M1306 включаются до завершения работы команды, направление выходного сигнала поменяется.
10. Когда **S_1** это вход X с внешним прерывателем, D1244 / D1255 могут быть применены для ограничения числа импульсов при некорректной работе прерывателя.
11. Команда DCLLM поддерживает функции метки и маски. См. описание команды **PLSR**.

Описание замкнутого контура:

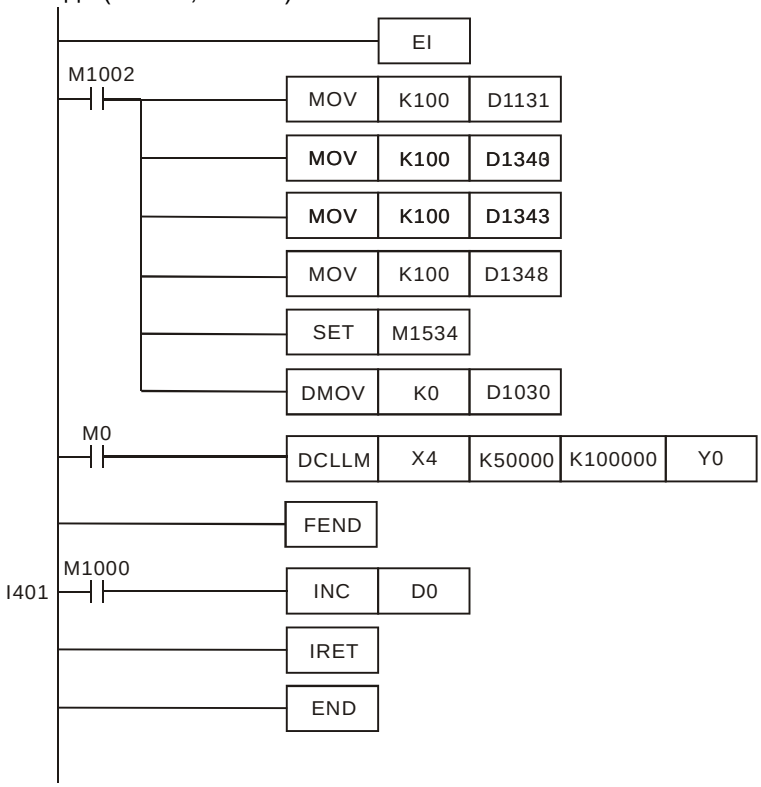
1. Функция: Немедленная остановка высокоскоростного импульсного выхода в соответствии с количеством импульсов обратной связи или сигналом с внешних прерывателей.
2. Временную диаграмму см. на следующей странице



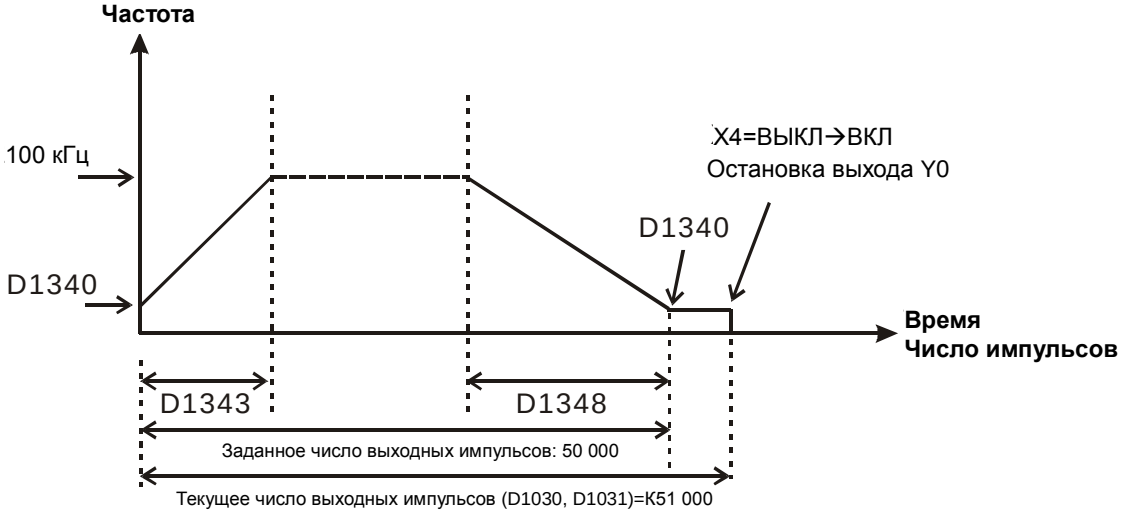
3. Принципы настройки времени завершения позиционирования:
   a) Время завершения позиционирования складывается из “разгон + высокая скорость + замедление + время завершения” (см. рисунок выше). При изменении коэффициента масштабирования (D1131/D1132) увеличивается или уменьшается общее число выходных импульсов и время завершения.
   b) Когда **S_1** – входы с прерывателями X, D1244 / D1255 могут быть применены для простого ограничения количества импульсов, срабатывающего при некорректном прерывании. Некорректное прерывание определяется временем завершения.
   c) Время завершения в работе замкнутого контура не будет одинаковым в каждой операции. Таким образом, если в D отображается большее или меньшее количество выходных импульсов, чем расчетное количество (заданное значение для обратной связи x коэффициент масштабирования / 100), можно исправить ситуацию путем изменения коэффициента масштабирования, времени рамповой функции или заданной частоты.

Пример программы 1: Немедленная остановка высокоскоростного импульсного выхода внешним прерывателем

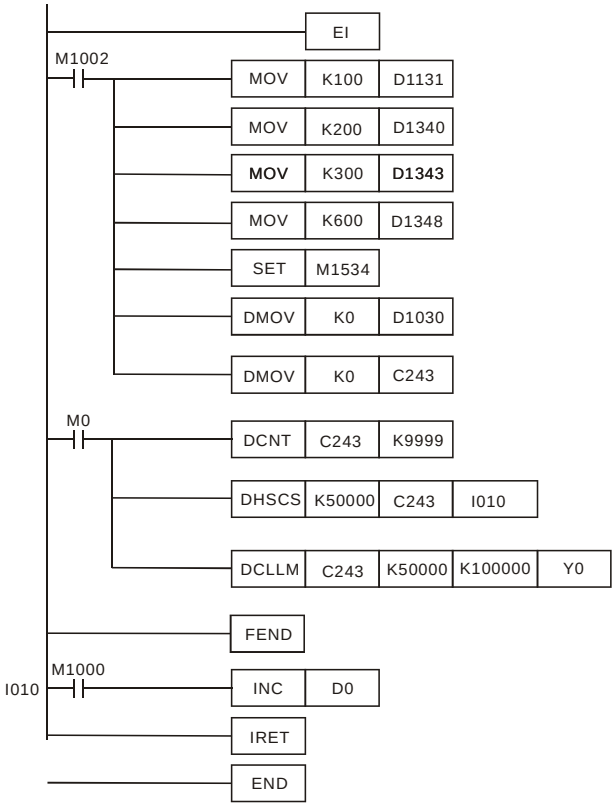
1. Принимаем X4 как вход с внешним прерывателем и I401 (триггер по переднему фронту) как прерыватель. Заданное количество импульсов обратной связи = 50 000; заданная частота = 100 000 Гц; импульсный выход: Y0, Y1 (CH0); начальная/конечная частота (D1340) = 100 Гц; время рампы разгона (D1343) = 100 мс; время рампы замедления (D1348) = 100 мс; процентное соотношение (D1131) = 100; значение импульсного выхода (D1030, D1031) = 0.



2. Результат выполнения:

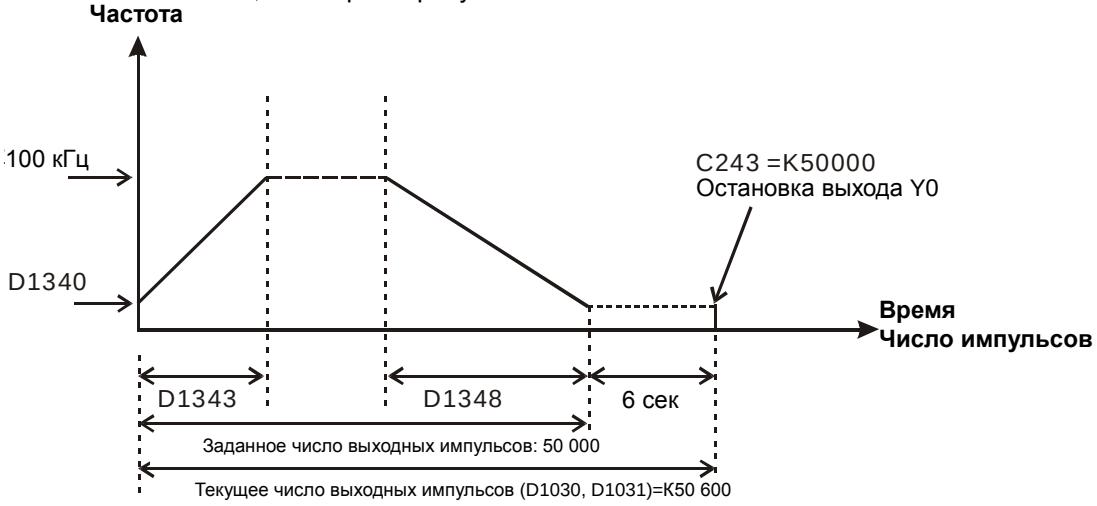


Пример программы 2: Немедленная остановка высокоскоростного импульсного выхода высокоскоростным счетчиком



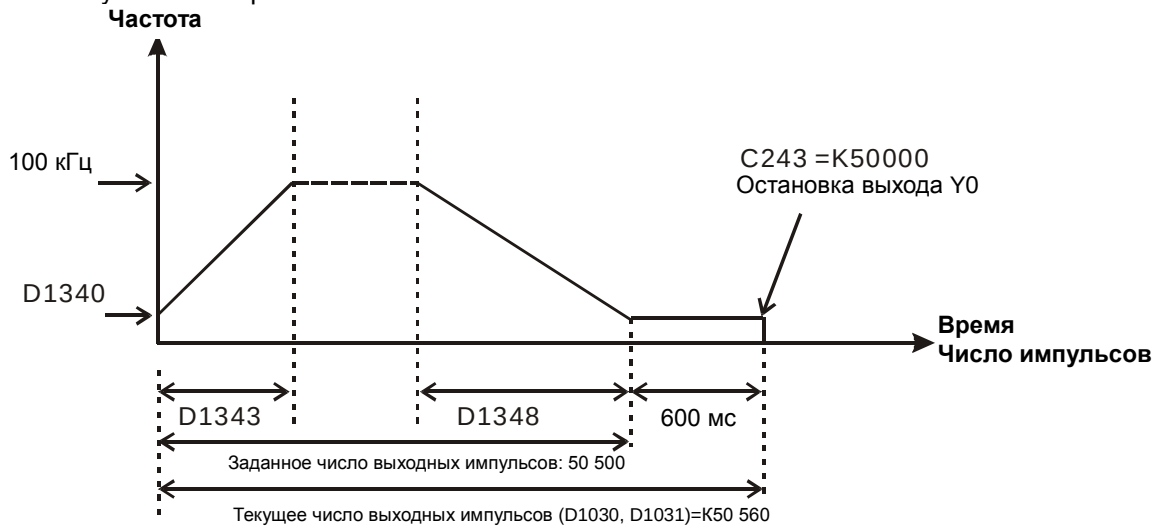
1. Примем счетчик C243 (рекомендуется сбросить перед выполнением) с AB-фазным входом от энкодера. Заданное количество импульсов обратной связи = 50 000; заданная частота = 100 000 Гц; импульсный выход: Y0, Y1 (CH0); начальная/конечная частота (D1340) = 100 Гц; время рампы разгона (D1343) = 100 мс; время рампы замедления (D1348) = 100 мс; процентное соотношение (D1131) = 100; значение импульсного выхода (D1030, D1031) = 0.

2. Предположим, что первые результаты:



3. Результаты первого выполнения:
   a) Текущее число импульсов на выходе 50600 – заданное число импульсов 50000 = 600
   b) 600 x (1/100 Гц) = 6 с (время завершения)
   c) 6 с – это слишком долгое завершение. Увеличим коэффициент масштабирования (D1131) до K101.

4. Результаты второго выполнения:



Результаты второго выполнения:
   a) Текущее число на выходе 50 560 – заданное число 50 500 (с учетом = 60
   b) 60 x (1/100 Гц) = 600 мс (время завершения)
   c) 600 мс – приемлемое значение.

Заметки:

1. Флаги:

M1029	Выдача импульсов завершена на CH0 (Y0).
M1102	Выдача импульсов завершена на CH1 (Y2)
M1078	Выдача импульсов приостановлена на CH0 (Y0)
M1104	Выдача импульсов приостановлена на CH1 (Y2)
M1108	Выдача импульсов приостановлена на CH0 (Y0, Y1) – рампа замедления. M1108 включен во время замедления.
M1110	Выдача импульсов приостановлена на CH1 (Y2, Y3) – рампа замедления. M1110 включен во время замедления.
M1156	Включение функций метки и маски на I400/I401(X4), относящимся к Y0.
M1158	Включение функций маски и метки на I600/I601(X6), относящимся к Y2.
M1538	Индикация паузы CH0 (Y0, Y1). M1538 включен во время паузы.
M1540	Индикация паузы CH1 (Y2, Y3). M1540 включен во время паузы.
M1305	Изменение направления импульсного выхода CH0 (Y0, Y1). M1305 включен при изменении направления.
M1306	Изменение направления импульсного выхода CH1 (Y2, Y3). M1306 включен при изменении направления.
M1347	Автосброс CH0 (Y0, Y1) при завершении работы импульсного выхода.
M1524	Автосброс CH1 (Y2, Y3) при завершении работы импульсного выхода.
M1534	Включение времени рампы замедления на Y0. Применяется с D1348.
M1535	Включение времени рампы замедления на Y2. Применяется с D1349.

2. Специальные регистры:

D1026:	Число импульсов для маски Y0 при включенном M1156 (мл. слово). Функция отключена при установленном значении ≦ 0. (по умолчанию = 0)
D1027:	Число импульсов для маски Y0 при включенном M1156 (ст. слово). Функция отключена при установленном значении ≦ 0. (по умолчанию = 0)
D1135:	Число импульсов для маскировки Y2 при включенном M1156 (мл. слово). Функция отключена при установленном значении ≦ 0. (по умолчанию = 0)
D1136:	Число импульсов для маскировки Y2 при включенном M1156 (ст. слово). Функция отключена при установленном значении ≦ 0. (по умолчанию = 0)
D1030:	Младшее слово текущего значения импульсного выхода CH0 (Y0, Y1)
D1031:	Старшее слово текущего значения импульсного выхода CH0 (Y0, Y1)

D1131: Вх/Вых процентное соотношение на CH0 (Y0, Y1. По умолчанию: K100

D1132: Вх/Вых процентное соотношение на CH1 (Y2, Y3) По умолчанию: K100

D1244: Время простоя (число импульсов), заданное CH0 (Y0, Y1) Функция отключена при установленном значении ≦ 0.

D1245: Время простоя (число импульсов), заданное CH2 (Y2, Y3) Функция отключена при установленном значении ≦ 0.

D1336: Младшее слово текущего значения импульсного выхода CH1 (Y2, Y3)

D1337: Старшее слово текущего значения импульсного выхода CH1 (Y2, Y3)

D1340: Начальная/конечная частота 1-й группы импульсных выходов CH0 (Y0, Y1). По умолчанию: K100

D1352: Начальная/конечная частота 2-й группы импульсных выходов CH1 (Y2, Y3). По умолчанию: K100

D1343: Время рампы 1-й группы импульсных выходов CH0 (Y0, Y1). Умолч.: K100

D1353: Время рампы 2-й группы импульсных выходов CH1 (Y2, Y3). Умолч.: K100

D1348: Импульсный выход CH0(Y0, Y1). Когда M1534 включен, D1348 сохраняет время рампы. По умолчанию: K100

D1349: Импульсный выход CH1(Y2, Y3). Когда M1535 включен, D1349 сохраняет время рампы. По умолчанию: K100

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
198	D	VSPO		S_1 S_2 S_3 D	Переменная скорость на импульсном выходе				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DVSPO: 17 шагов
S_1													*			
S_2					*	*							*			
S_3					*	*							*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

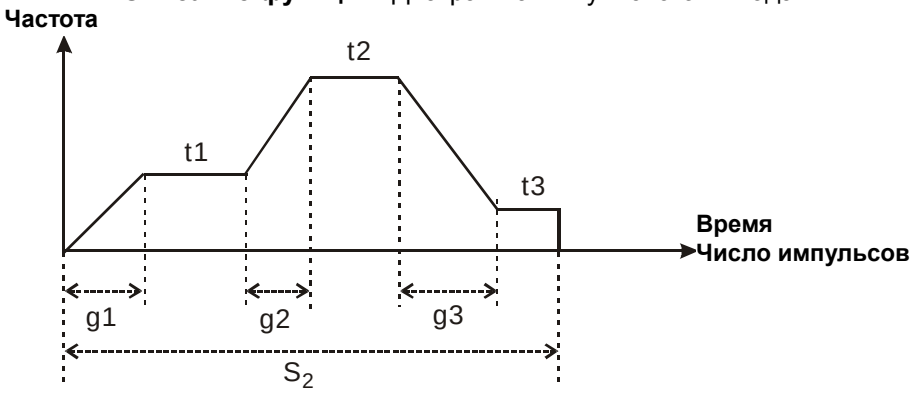
Операнды:

S_1: Заданная выходная частота **S_2**: Заданное число импульсов **S_3**: Шаг изменения частоты и длительность каждого шага **D**: Импульсный выход (Y0, Y2)

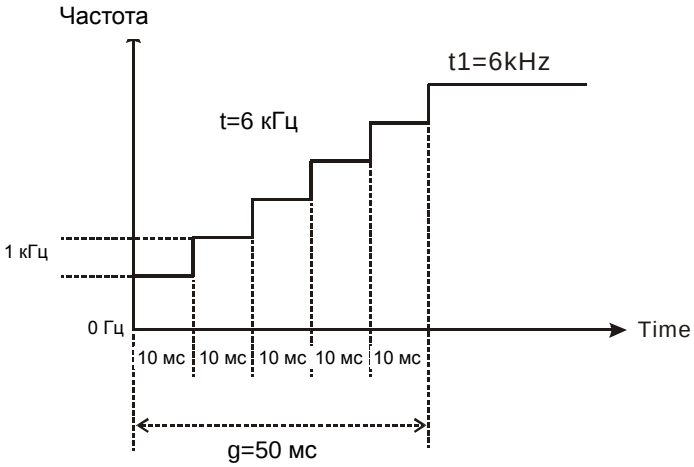
Описание:

1. Максимальная частота **S_1**: 100 кГц. Заданная частота может быть изменена во время выполнения команды. При изменении **S_1** команда VSPO применит рамповую функцию для достижения заданной частоты в соответствии со значениями шага и времени в **S_3**.
2. **S_2** как заданное число импульсов действует только при первом запуске команды. **S_2** не может быть изменено во время выполнения команды. **S_2** может иметь отрицательное значение, но если не указано направление движения в D1220/D1221, ПЛК будет считать его положительным значением. Когда заданное количество импульсов равно 0, ПЛК будет выдавать на выходе непрерывный сигнал.
3. **S_3** занимает 2 последовательных 16-битных адреса. **S_3**+0 сохраняет шаг изменения частоты, **S_3**+1 – длительность каждого шага. Параметры настройки могут быть изменены во время выполнения команды. Диапазон **S_3**+0: 6 Гц ~ 32 767 Гц; диапазон **S_3**+1: 1 мс ~ 80 мс. При выходе за пределы диапазона ПЛК будет оперировать соответственно минимальным или максимальным допустимым значением.
4. **D** поддерживает только импульсные выходы Y0 и Y2. Если Y1 и Y3 применяются для определения направления выходного сигнала, D1220 или D1221 должен иметь значение K1(Pulse/Dir).
5. Параметры, установленные в **S_3**, могут быть изменены только при изменении значения в **S_1**. Когда заданная частота равна 0, ПЛК проводит замедление до значений, установленных в **S_3**. Когда выдача сигнала на выходе остановится, будет включен флаг, показывающий состояние паузы (Y0: M1538, Y2: M1540). Если заданная частота, отличная от 0, задана снова, импульсный выход проводит разгон до заданной частоты и действует до достижения заданного числа импульсов.

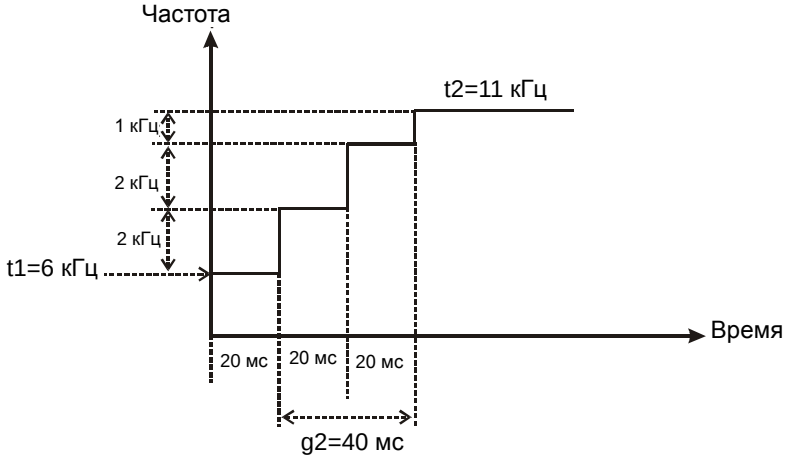
Описание функций: Диаграмма импульсного выхода:



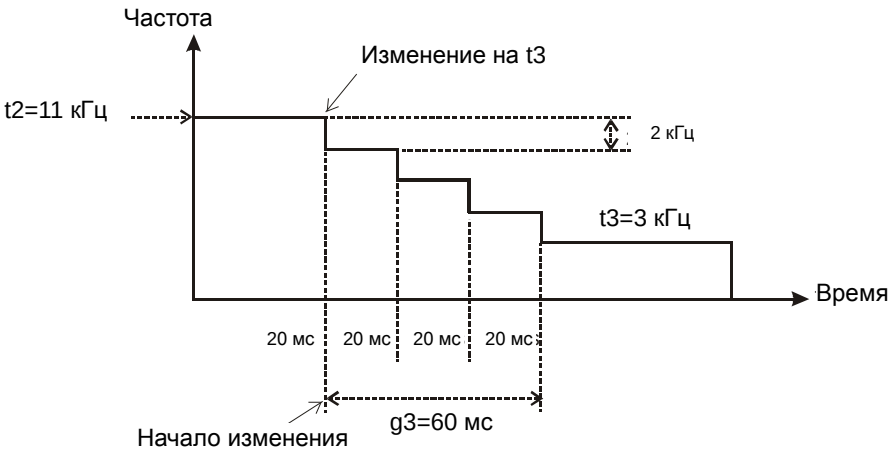
1. Определения:
   t1 → заданная частота 1го участка; g1 → время разгона до 1ой частоты
   t2 → заданная частота 2го участка; g2 → время разгона до 2ой частоты
   t3 → заданная частота 3го участка; g3 → время замедления до 3ей частоты
   S2 → все выходные импульсы
2. Описание каждого шага:
   1й участок: Примем t1 = 6 кГц, шаг изменения частоты = 1 кГц, длительность каждого шага = 10 мс
   Диаграмма разгона 1го участка:



2й участок:
Примем t2 = 11 кГц, шаг изменения частоты = 2 кГц, длительность каждого шага = 20 мс



3й участок:
Примем t3 = 3 кГц, шаг изменения частоты = 2 кГц, длительность каждого шага = 20 мс



3. Пример программы см. API 199

Заметки:

1. Флаги:
   - M1029 Выдача импульсов завершена на CH0 (Y0, Y1)
   - M1102 Выдача импульсов завершена на CH1 (Y2, Y3)
   - M1078 Выдача импульсов приостановлена на Y0
   - M1104 Выдача импульсов приостановлена на Y2
   - M1305 Изменение направления импульсного выхода Y1
   - M1306 Изменение направления импульсного выхода Y3
   - M1538 Индикация паузы Y0
   - M1540 Индикация паузы Y2
2. Специальные регистры:
   - D1030 Младшее слово текущего значения импульсного выхода Y0
   - D1031 Старшее слово текущего значения импульсного выхода Y0
   - D1336 Младшее слово текущего значения импульсного выхода Y2
   - D1337 Старшее слово текущего значения импульсного выхода Y2
   - D1220 Режим настроек импульсного выхода CH0 (Y0, Y1). См. команду PLSY
   - D1221 Режим настроек импульсного выхода CH1 (Y2, Y3). См. команду PLSY

API	Команда		Операнды	Функция	Контроллеры
199	D	ICF	(S1) (S2) (D)	Мгновенное изменение частоты	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	DVSPO: 13 шагов
S_1													*			
S_2					*	*							*			
D		*														

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

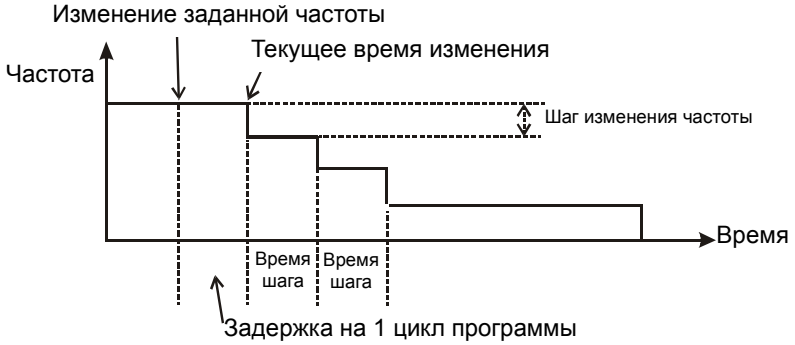
S_1: Заданная частота **S_2**: Шаг изменения частоты и длительность каждого шага
D: Импульсный выход (Y0, Y2)

Описание:

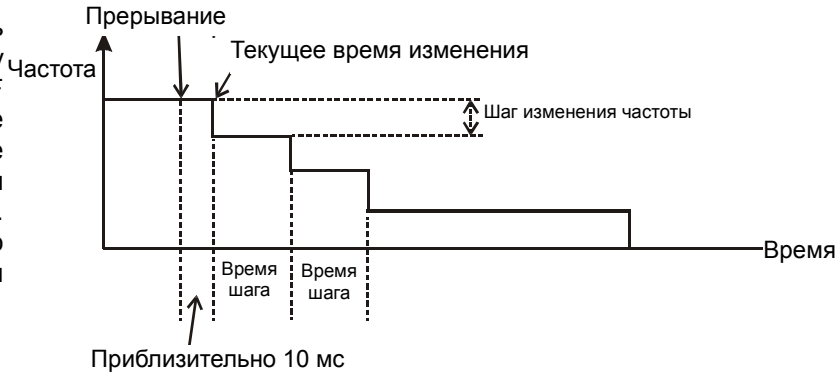
1. Максимальная частота **S_1**: 100 кГц. При выполнении команды ICF изменение частоты начнется сразу же с рампового процесса.
2. Команда ICF должна быть выполнена после выполнения команд DVSPO или DPLSY. Когда команда используется вместе с DVSPO, операнды **S_1**, **S_2**, **D** в DICF должны относиться к тому же устройству, что и **S_1**, **S_3**, **D** в DVSPO. Когда команда выполняется с DPLSY, операнды **S_1**, **D** должны относиться к тому же устройству, что и **S_1**, **D** для DPLSY**.**
3. Если команда ICF используется с командой DPLSY, операнд **S_2** не применяется.
4. Когда команда ICF используется с командой DVSPO, установки параметров в **S_2** такие же, как **S_3** в команде DVSPO, указывают изменение времени и изменение частоты рампового процесса.
5. **D** поддерживает только импульсные выходы Y0 и Y2.
6. Команду можно применять в подпрограмме прерывания для снижения времени отклика.
7. Флаги и регистры см. **Заметки** API 198 команды DVSPO.

Описание функций:

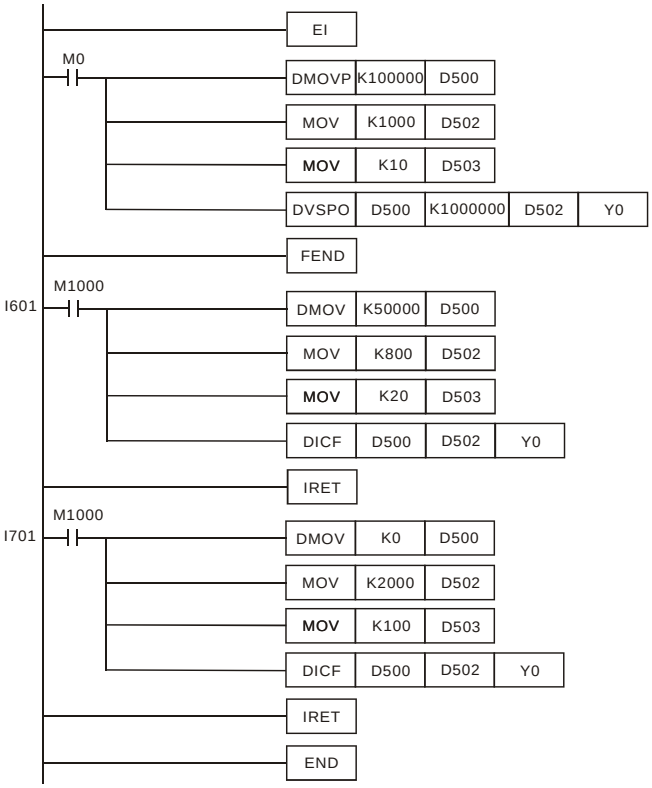
1. Если пользователь меняет заданную частоту с помощью команды DVSPO, фактическое изменение времени будет отложено из-за времени сканирования программы и изменения времени.



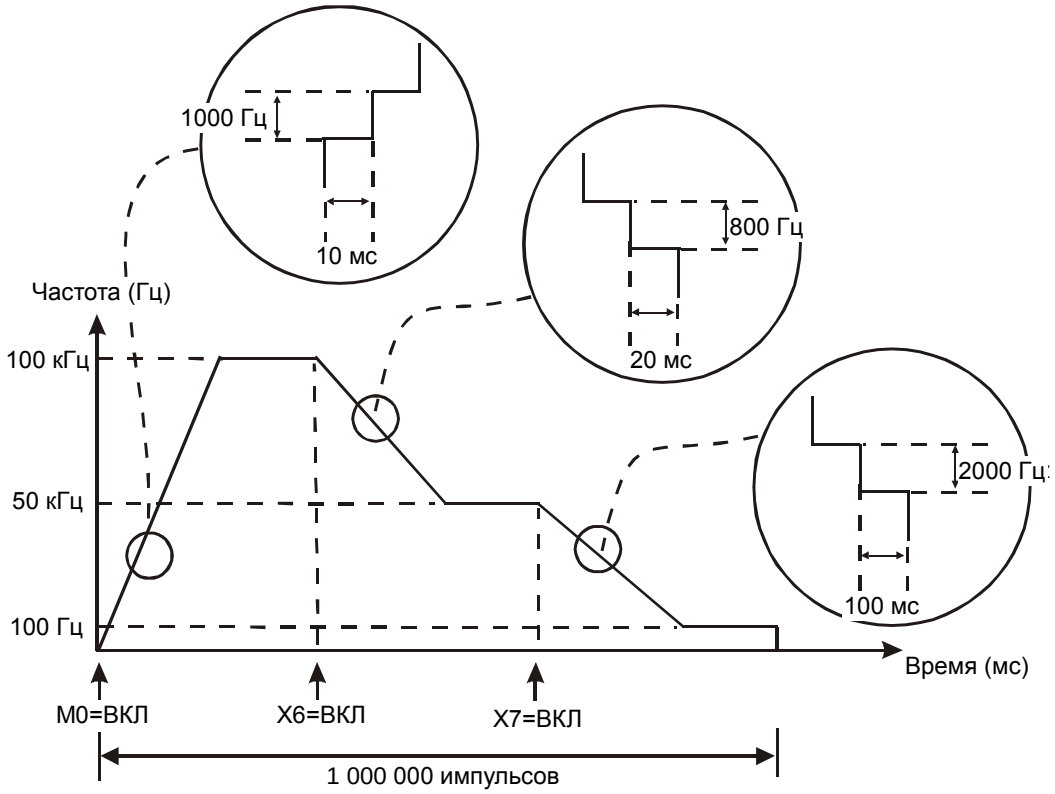
2. Если пользователь меняет заданную частоту с помощью команды DICF в подпрограмме прерывания, фактическое изменение времени начинается сразу. Необходимо только около 10 мкс для выполнения команды DICF.



Пример программы:



1. Когда M0 вкл., импульсный выход разгоняется до 100 кГц. Всего шагов = 100; Шаг изменения частоты = 1000 Гц; длительность шага = 10 мс. Подсчет всех шагов: (100 000-0) ÷ 1000 = 100.
2. Когда X6 вкл. внешним прерывателем, частота падает до 50 кГц. Всего шагов = 63; Шаг изменения частоты = 800 Гц; длительность шага = 20 мс. Подсчет всех шагов
(100 000 - 50 000) ÷ 800 = 63
3. Когда X7 вкл. внешним прерывателем, заданная частота падает до 100 Гц. Всего шагов = 25, шаг изменения частоты = 2000 Гц, длительность шага = 100 мс. Подсчет всех шагов: (50,000 - 100) ÷ 2000 = 25.
4. Когда частота на импульсном выходе достигает 100 Гц, данная частота поддерживается постоянно и импульсный выход прекращает работу по достижении 1 000 000 импульсов.



API	Команда			Операнды	Функция	Контроллеры			
202		SCAL	P	S_1 S_2 S_3 D	Расчет пропорционального значения	ES2/EX2	SS2	SA2/ SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	SCAL,SCLAP: 9 шагов
S_1					*	*							*			
S_2					*	*							*			
S_3					*	*							*			
D													*			

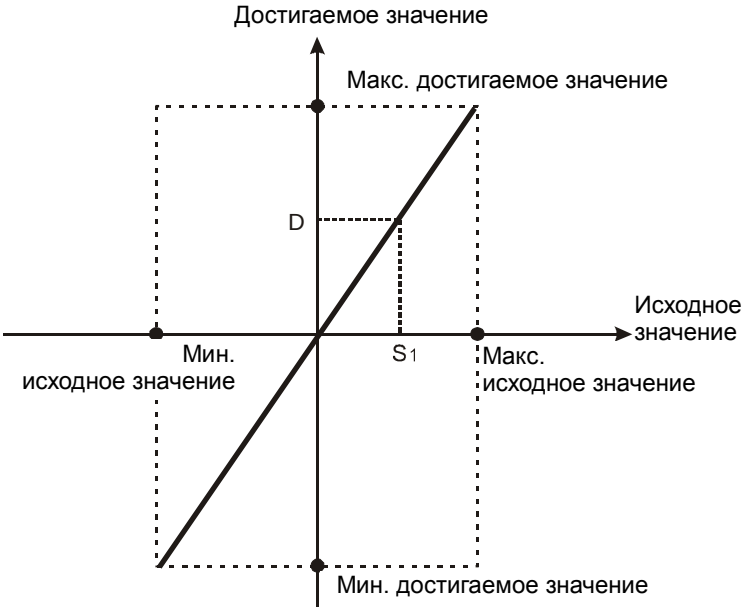
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

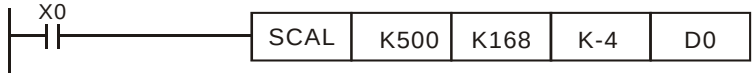
S_1: Исходное значение **S_2**: Пропорциональность (шаг: 0.001) **S_3**: Смещение **D**: Результат Диапазон операндов **S_1, S_2, S_3**: -32768~32767.

Описание:

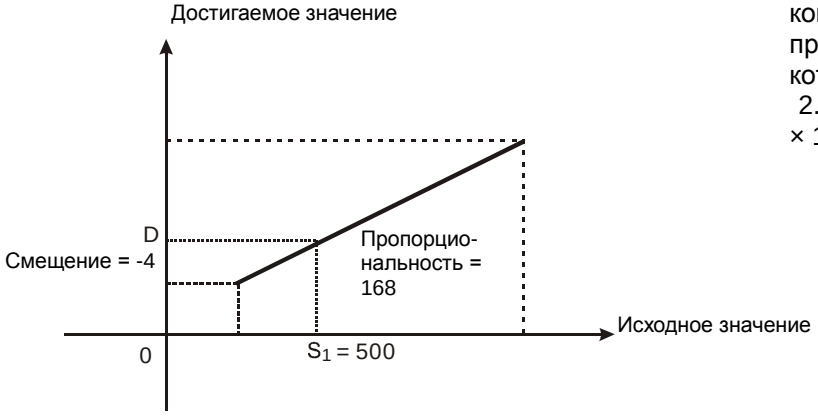
1. Команда SCAL осуществляет пропорциональный расчет согласно уравнению **D** = (**S_1** × **S_2**) ÷ 1000 + **S_3**
2. Сначала необходимо получить значения **S_2** и **S_3** (десятичные округляются для 16-разрядных чисел) для применения в уравнении.
3. Пропорциональность: **S_2** = [(макс. достигаемое значение – мин. достигаемое значение) ÷ (макс. исходное значение – мин. исходное значение)] × 1 000
4. Смещение: **S_3** = мин. достигаемое значение – мин. исходное значение × **S_2** ÷ 1 000



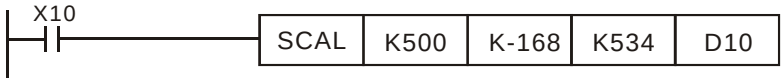
Пример программы 1:



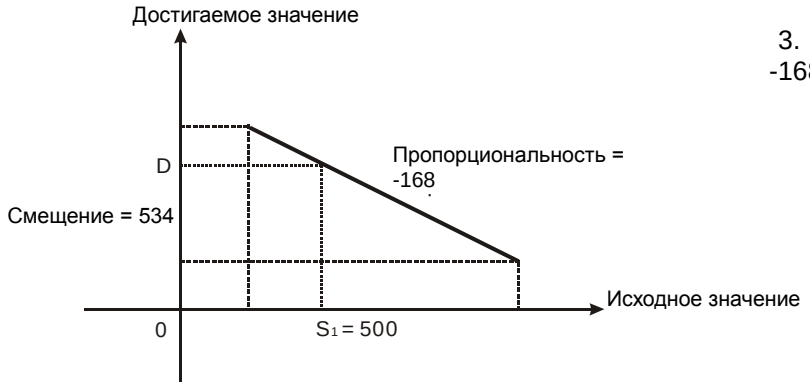
1. Примем **S_1** = 500, **S_2** = 168 и **S_3** = -4. При включенном X0 команда SCAL рассчитывает пропорциональный результат, который будет сохранен в D0.
2. Формула расчета: D0 = (500 × 168) ÷ 1000 + (-4) = 80



Пример программы 2:



1. Примем **S_1** = 500, **S_2** = -168 и **S_3** = 534.
2. При включенном X0 команда SCAL рассчитывает пропорциональный результат, который будет сохранен в D0.



3. Формула расчета: D10 = (500 × -168) ÷ 1000 + 534 = 450

Заметки:

1. Эта команда применяется, когда пропорциональность и смещение известны, если они неизвестны, необходимо применить команду SCLP для их расчета.
2. S_2 должен быть в пределах -32,768 ~ 32,767. Если S_2 находится за пределами указанного диапазона, используйте команду SCLP.
3. Для уравнения пропорциональности: максимальное значение источника должно быть больше минимального, но максимальное достижимое не должно быть больше минимального.
4. Если **D** > 32,767, **D** считается как 32,767. Если **D** < -32,768, **D** считается как -32,768.

API	Команда			Операнды	Функция	Контроллеры			
						ES2/EX2	SS2	SA2/SE	SX2
203	D	SCLP	P	(S1) (S2) (D)	Параметры пропорционального расчета значений				

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*							*			SCLP, SCLPP: 7 шагов DSCLP, DSCLPP: 13 шагов
S_2													*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Исходное значение S_2: Параметры **D**: Результат

Описание:

1. Команда SCLP определяет параметры для пропорционального расчета и сам пропорциональный расчет согласно уравнению пропорциональности.
2. Параметры S_2 для 16-битной команды (занимают 4 последовательных адреса):

Адрес	Параметр	Диапазон
S_2	Макс. исходное значение	-32768~32767
S_2+1	Мин. исходное значение	-32768~32767
S_2+2	Макс. достижимое значение	-32768~32767
S_2+3	Мин. достижимое значение	-32768~32767

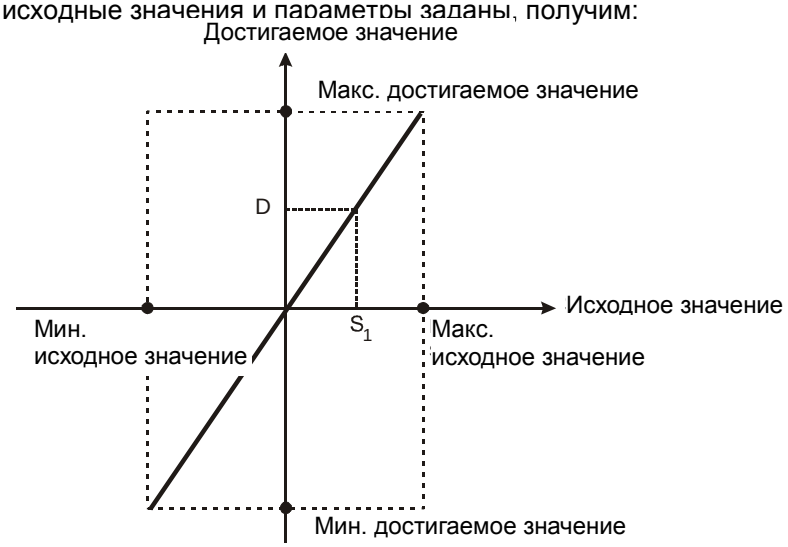
3. Параметры S_2 для 32-битной команды (занимают 8 последовательных адресов):

Адрес	Параметр	Диапазон	
		Целое	Число с плавающей запятой
S_2, S_2+1	Макс. исходное значение	-2,147,483,648~2,147,483,647	Диапазон 32-битных чисел с плавающей запятой
S_2+2, 3	Мин. исходное значение		
S_2+4, 5	Макс. достижимое значение		
S_2+6, 7	Мин. достижимое значение		

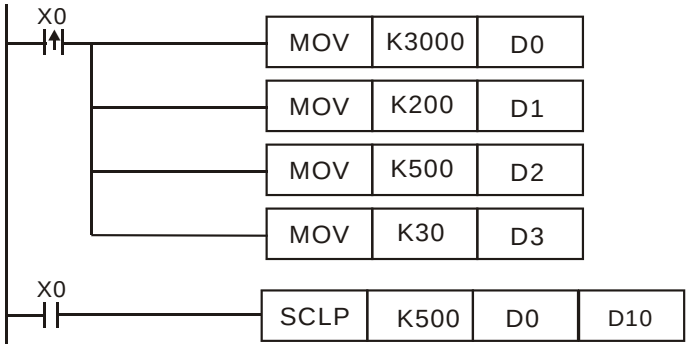
4. Уравнение: **D** = [(S_1 – мин. исходное значение) × (макс. достижимое значение – мин. достижимое значение)] ÷ (макс. исходное значение – мин. исходное значение) + мин. достижимое значение
5. Формула для получения рабочего уравнения команды:
   y = kx + b, где
   y = Достигаемое значение (**D**)
   k = коэффициент пропорциональности = (макс.достижимое значение – мин.достижимое значение) ÷ (макс.исходное значение – мин.исходное значение)
   x = Исходные данные (S_1)
   b = Смещение = мин.достижимое значение – мин.исходное значение × Наклон
6. Подставим указанные выше параметры в **y = kx + b** и получим операцию, выполняемую командой. y = kx + b = D = k S_1 + b = наклон × S_1 + смещение = наклон × S_1 + мин. достижимое значение – мин. исходное значение × наклон = наклон × (S_1 – мин. исходное

значение) + мин. достижимое значение = (S_1 – мин. исходное значение) × (макс. достижимое значение – мин. достижимое значение) ÷ (макс. исходное значение – мин. исходное значение) + мин. достижимое значение.

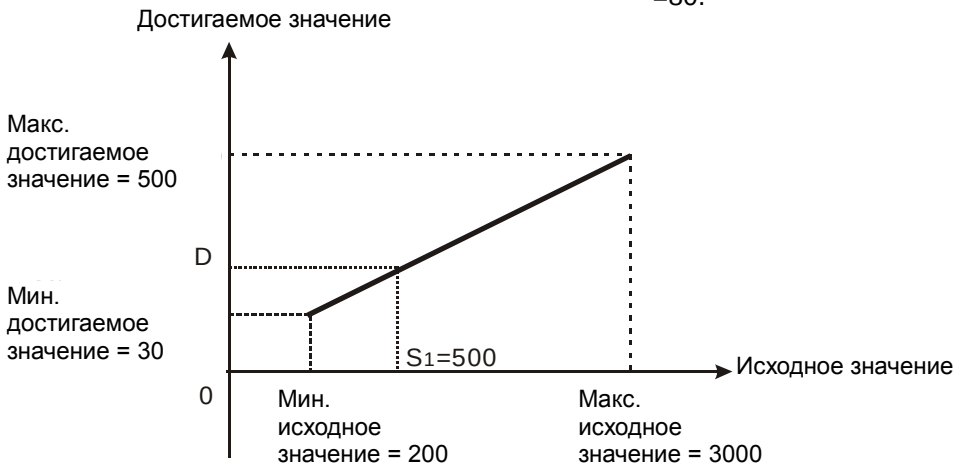
7. Если S_1 > макс. исходное значение, S_1 определяется как макс. исходное значение. Если S_1 < мин. исходное значение, S_1 определяется как мин. исходное значение. Если исходные значения и параметры заданы, получим:



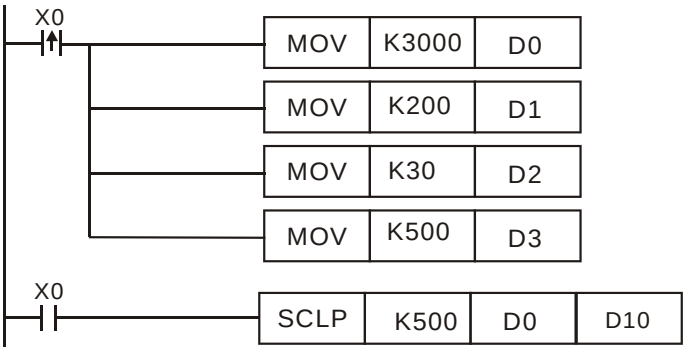
Пример программы 1:



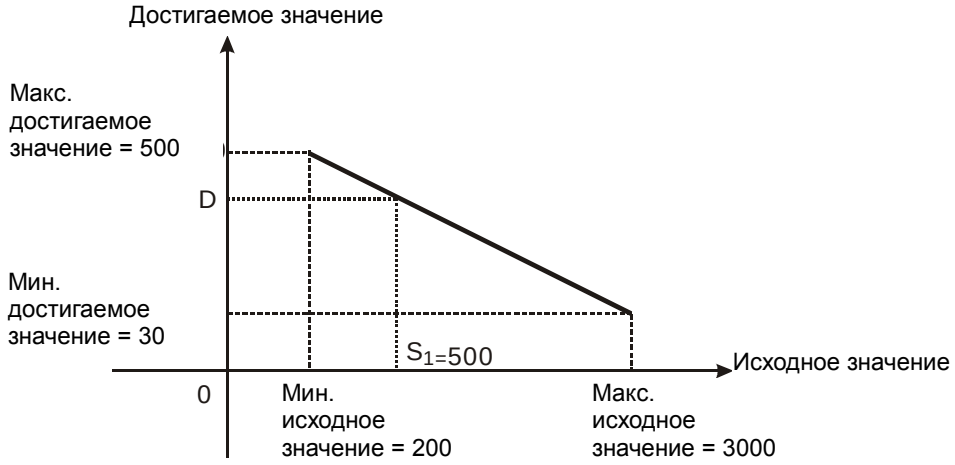
1. Примем исходное значение S_1 = 500, макс. исходное значение D0 = 3000, мин. исходное значение D1 = 200, макс. достижимое значение D2 = 500 и мин. достижимое значение D3 = 30. Когда X0 вкл., SCLP рассчитывает пропорциональное значение и сохраняет его в D10.
2. D10 = [(500 – 200) × (500 – 30)] ÷ (3000 – 200) + 30 = 80.35. Округляем результат до целого D10 =80.



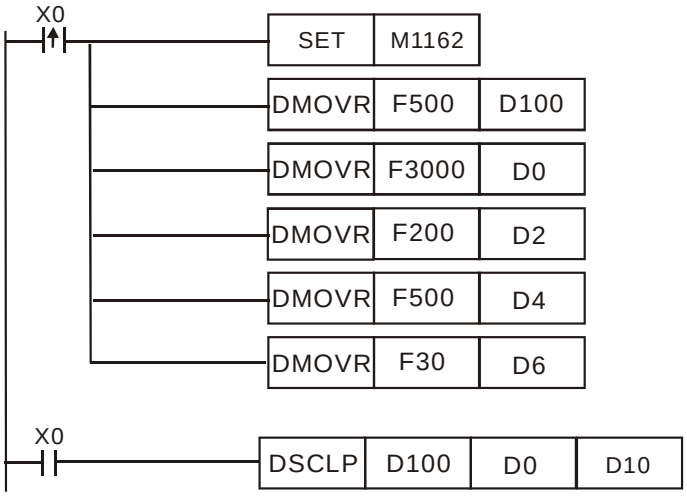
Пример программы 2:



1. Примем исходное значение S_1 = 500, макс. исходное значение D0 = 3000, мин. исходное значение D1 = 200, макс. достижимое значение D2 = 30 и мин. достижимое значение D3 = 500. Когда X0 вкл., команда SCLP рассчитывает пропорциональное значение и сохраняет его в D10.
2. Уравнение: D10 = [(500 – 200) × (30 – 500)] ÷(3000 – 200) + 500 = 449.64. Округляем результат до целого D10 = 450.



Пример программы 3:



1. Примем исходное значение **S_1**, D100 = F500, макс. исходное значение D0 = F3000, мин. исходное значение D2 = F200, макс. достижимое значение D4 = F500, и мин. достижимое значение D6 = F30. Когда X0 включен, M1162 настраивает команду на выполнение операции с числами с плавающей запятой. Команда DSCLP рассчитывает пропорциональное значение и сохраняет его в D10.
2. Уравнение: D10 = [(F500 – F200) × (F500 – F30)] ÷ (F3000 – F200) + F30 = F80.35. Округляем результат до целого D10 = F80.

Заметки:

1. Диапазон **S_1** для 16-битной команды: макс. исходное значение ≥ **S_1** ≥ мин. исходное значение; -32,768 ~ 32,767. При выходе значения за пределы диапазона, для расчета будут применяться соответствующие предельные значения.
2. Диапазон **S_1** для 32-битной команды: макс. исходное значение ≥ **S_1** ≥ мин. исходное значение; -2,147,483,648 ~ 2,147,483,647. При выходе значения за пределы диапазона, для расчета будут применяться соответствующие предельные значения.
3. Диапазон для числа с плавающей запятой **S_1** для 32-битной команды: макс. исходное значение ≥ **S_1** ≥ мин. исходное значение; допустимые значения для 32-битных чисел с плавающей запятой. При выходе значения за пределы диапазона, для расчета будут применяться соответствующие предельные значения.
4. Для уравнения пропорциональноти: максимальное значение источника должно быть больше минимального, но максимальное достижимое не должно быть больше минимального.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
205	CMPT	P	(S1) (S2) (n) (D)	Таблица сравнения				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			CMPT: 9 шагов
S_2											*	*	*			CMPTP: 9 шагов
n					*	*							*			
D								*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Источник 1 **S_2**: Источник 2 **n**: Длина данных (**n** = 1~16) **D**: Результат

Описание:

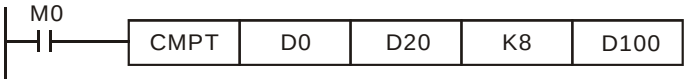
1. **S_1** и **S_2** могут быть словными T/C/D, для C (только 16-битные) применимы (C0~C199).
2. Значения 16 старших битов в **n** для 32-битной команды являются недопустимыми.
3. Диапазон операнда **n**: 1~16. ПЛК использует верхнее/нижнее значение при выходе за пределы диапазона.
4. Данные в операнде **D** сохраняются в 16-битном или 32-битном формате. Когда длина данных меньше 16 (32), нулевые биты фиксируются как 0, т.е. если **n** = K8, биты 0~7 будут участвовать в сравнении, а биты 8~15 (8-31) все будут равны 0.
5. 32-битная команда доступна для ПЛК DVP-ES2/EX2 версии 3.0, DVP-SS2 версии 2.8, DVP-SA2 версии 2.6, DVP-SX2 версии 2.4 и DVP-SE.
6. Значения 8 старших битов **n** показывают условие сравнения:

Значение	K0	K1	K2	K3	K4
Условие сравнения	**S1** = **S2**	**S1** < **S2**	**S1** ≤ **S2**	**S1** > **S2**	**S1** ≥ **S2**

7. Пример настройки **n**: если значение для 16-битной команды равно H0108, 8 фрагментов данных сравниваются с другими восемью частями данных по условию сравнения «больше». Если значение для 32-битной команды равно H00000320, 32 фрагмента данных сравниваются с другими 32 частями данных по условию сравнения «меньше».
8. Если значение параметра сравнения выходит за пределы допустимого диапазона или версия прошивки ПЛК не поддерживает сравнение (см. п.5), по умолчанию сравнение дает результат «равно».
9. Значение сравнения для 16-битной команды является 16-битным, для 32-битной команды – 32-битным значением (M1162 выключен) или значением с плавающей запятой (M1162 включен).
10. Если результат сравнения удовлетворяет условию, соответствующий бит равен 1, если не удовлетворяет, соответствующий бит равен 0.

Пример программы:

Когда M0 включен, сравниваются 16-битные значения в D0~D7 с D20~D27 и результат сохраняется в D100.



- Содержимое D0~D7:

Номер	D0	D1	D2	D3	D4	D5	D6	D7
Значение	K10	K20	K30	K40	K50	K60	K70	K80

- Содержимое D20~D27:

Номер	D20	D21	D22	D23	D24	D25	D26	D27
Значение	K12	K20	K33	K44	K50	K66	K70	K88

- При сравнении результат равен 1 при совпадении битов и 0 при несовпадении:

	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8~15
D100	0	1	0	0	1	0	1	0	0…0
	H0052 (K82)								

API	**Команда**	**Операнды**	**Функция**	**Контроллеры**			
206	ASDRW	S_1 S_2 S	Чтение/запись для сервопривода ASDA	ES2/EX2	SS2	SA2/SE	SX2

Тип / **Операнд**	**Биты**				**Слова**											**Шаги программы**
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ASDRW: 7 шагов
S_1					*	*							*			
S_2					*	*							*			
S													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Адрес сервопривода (K0~K254) **S_2**: Функциональный код **S**: Регистр читаемых/записываемых данных

Описание:

1. Команда ASDRW поддерживает порты COM2 (RS-485) и COM3 (RS-485)

2. S_1: номер сервопривода. Диапазон: K0~K254. K0 связь со всеми устройствами, т.е. ПЛК не будет получать данных обратной связи
3. Функциональный код S_2 и Регистр читаемых/записываемых данных**S**.

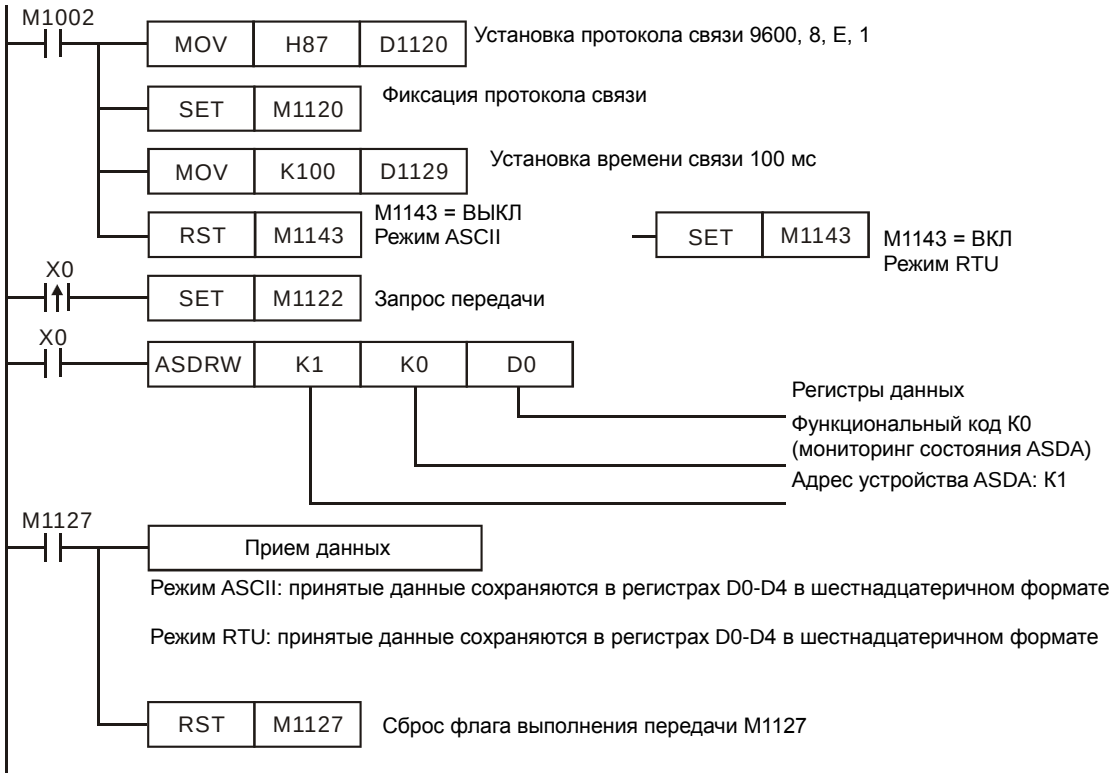
Специально для ASDA типов A, AB, A+, B				
Код	Функция	Параметр	Адрес связи	Данные чтения/записи (настройки)
K0(H0)	Индикация состояния	P0-04 ~ P0-08	0004H ~ 0008H	**S**+0 ~ **S**+4: См. описание ASDA.
K1(H1)	Регистр блока читаемых данных	P0-09 ~ P0-16	0009H ~ 0010H	**S**+0 ~ **S**+7: См. описание ASDA. Тип B не поддерживается.
K2(H2)	Регистр блока записываемых данных	P0-09 ~ P0-16	0009H ~ 0010H	**S**+0 ~ **S**+7: См. описание ASDA. Тип B не поддерживается.
K3(H3)	JOG-операция	P4-05	0405H	**S**: Диапазон: 1~3000, 4999, 4998, 5000
K4(H4)	Сервопривод ВКЛ/ВЫКЛ	P2-30	021EH	**S**: K1 = ВКЛ, другие = ВЫКЛ
K5(H5)	Управление скоростью (3 установки)	P1-09 ~ P1-11	0109H ~ 010BH	**S**+0 ~ **S**+2: Диапазон: -5000~+5000
K6(H6)	Управление моментом (3 установки)	P1-12 ~ P1-14	010CH ~ 010EH	**S**+0 ~ **S**+2: Диапазон: -300~+300

Только для типа A2				
Код	Функция	Параметр	Адрес связи	Данные чтения/записи (настройки)
K16(H10)	Индикация состояния (чтение)	P0-09 ~ P0-13	0012H ~ 001BH	**S**+0 ~ **S**+9: См. описание ASDA-A2
K17(H11)	Выбор индикации состояния (запись)	P0-17 ~ P0-21	0022H ~ 002BH	**S**+0 ~ **S**+9: См. описание ASDA-A2
K18(H12)	Отображение параметров (запись)	P0-25 ~ P0-32	0032H ~ 0041H	**S**+0 ~ **S**+15: См. описание ASDA-A2
K19(H13)	JOG-операция	P4-05	040AH	**S**: Диапазон: 1~5000, 4999, 4998, 0
K20(H14)	Сервопривод ВКЛ/ВЫКЛ	P2-30	023CH	**S**: K1 = ВКЛ, другие = ВЫКЛ
K21(H15)	Управление скоростью (3 установки)	P1-09 ~ P1-11	0112H ~ 0117H	**S**+0 ~ **S**+5: Диапазон: -60000~+60000
K22(H16)	Управление моментом (3 установки)	P1-12 ~ P1-14	0118H ~ 011DH	**S**+0 ~ **S**+5: Диапазон: -300~+300
K23(H17)	Регистр блока читаемых/записываемых данных (для отображаемых параметров)	P0-35 ~ P0-42	0046H~ 0055H	**S**+0 ~ **S**+15: См. описание ASDA-A2

4. Флаги M и специальные регистры D см. описание API 80 команды RS.

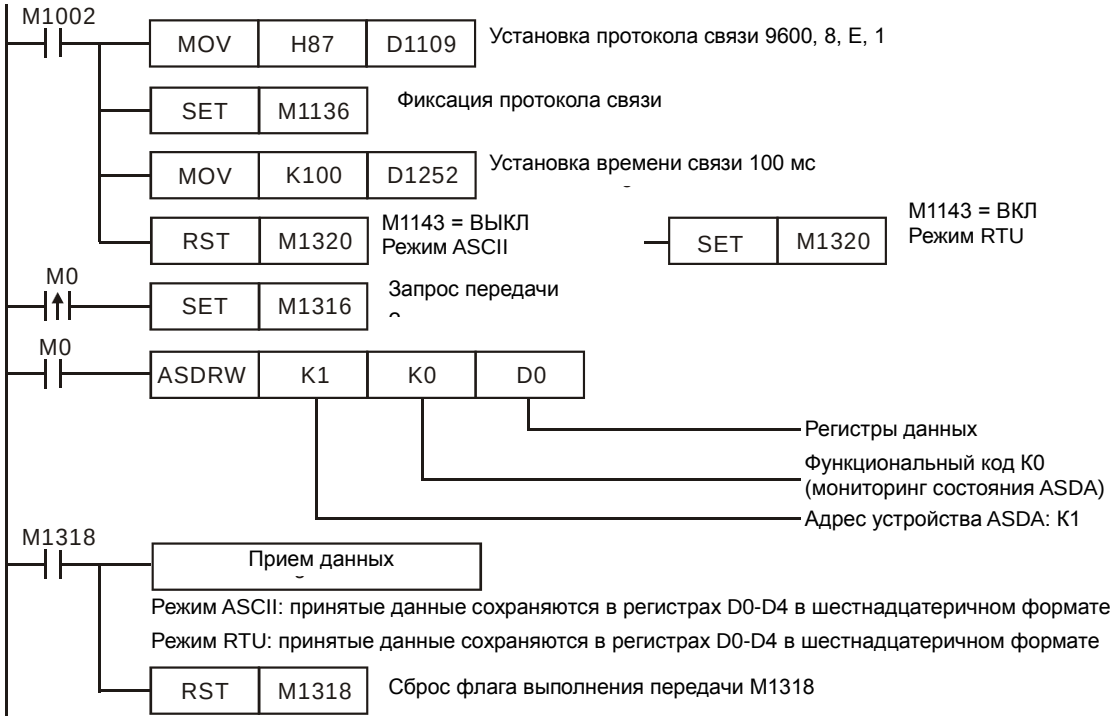
Пример программы 1: COM2 (RS-485)

1. Когда X0 включен, ПЛК направит команду для связи по COM2 для чтения состояния сервопривода.
2. При получении ПЛК данных обратной связи от ASDA, включится M1127, и считываемые данные будут сохраняться в D0~D4.



Пример программы 2: COM3(RS-485)

1. Когда X0 включен, ПЛК направит команду для связи по COM3 для чтения состояния сервопривода.
2. При получении ПЛК данных обратной связи от ASDA, включится M1318, и считываемые данные будут сохраняться в D0~D4.



Заметки:

Флаги и специальные регистры D для COM2/COM3 :

	COM2	COM3	Функция
Протокол настроек	M1120	M1136	Сохранение настроек связи
	M1143	M1320	ASCII/RTU выбор режима
	D1120	D1109	Протокол связи
	D1121	D1255	Адрес связи ПЛК
Отправка запроса	M1122	M1316	Отправка запроса
	D1129	D1252	Настройка паузы связи (мс)
Прием завершен	M1127	M1318	Прием данных завершен
Ошибки	-	M1319	Ошибка приема данных
	-	D1253	Код ошибки связи
	M1129	-	Настройка паузы связи (мс)
	M1140	-	COM2 (RS-485) MODRD/MODWR/MODRW ошибка приема данных
	M1141	-	MODRD/MODWR/MODRW параметры ошибки (Код исключения в данных) Код исключения хранится в D1130
	D1130	-	COM2 (RS-485) Код ошибки (код исключения) возвращаемый по связи Modbus

API	Команда	Операнды	Функция	Контроллеры
207	CSFO	S S1 D	Определение частоты сигнала на входе и масштабирование ее на выходе	ES2/EX2 SS2 SA2/SE SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	CSFO: 7 шагов
S	*															
S_1													*			
D													*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S: Источник входного сигнала (доступны только X0~X3) **S_1**: Параметры сигнала на входе

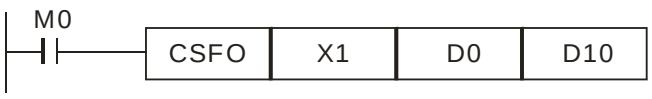
D: Параметры сигнала на выходе

Описание:

1. Когда **S** является X0, ПЛК использует только X0 в качестве входа и связанный с ним высокоскоростной импульсный выход Y0, в этом случае Y1 является нормальным выходом. Когда **S** является X1, ПЛК использует X0 (фаза A) и X1 (фаза B) как входы и связанные с ними выходы: Y0 (Pulse) / Y1 (Dir). Когда **S** является X2, ПЛК использует X2 как вход и связанный с ним высокоскоростной импульсный выход Y2, в этом случае Y3 является нормальным выходом. Когда **S** является X3, ПЛК использует X2 (фаза A) и X3 (фаза B) как входы и связанные с ними выходы: Y2 (Pulse) / Y3 (Dir).
2. Выполнение команды CSFO требует аппаратной реализации высокоскоростного счетчика и функции высокоскоростного импульсного выхода. Поэтому, когда в цикле программы выполняется данная команда с высокоскоростным счетчиком, а на входах (X0, X1) или (X2, X3) запускается команда DCNT, или на высокоскоростных импульсных выходах (Y0, Y1) или (Y2, Y3) запускаются другие высокоскоростные команды, команда CSFO работать не будет.
3. Если **S** является X1 / X3 2-фазными 2 входами, счетный режим устанавливается как четверная частота.
4. Во время работы импульсных выходов Y0 или Y2 специальные регистры (D1031, D1330 / D1337, D1336), сохраняющие текущее количество выходных импульсов, будут обновляться во время работы команды CSFO в цикле программы.
5. $\mathbf{S_1}$ занимает 4 последовательных 16-битных регистров. $\mathbf{S_1}$ +0 определяет время дискретизации, т.е. когда $\mathbf{S_1}$ +0 является K1, ПЛК определяет частоту на входе каждый раз после выдачи импульса. Допустимый диапазон для $\mathbf{S_1}$ +0 на 1 фазе 1 выхода: K1~K100, а для 2 фаз и 2 выходов: K2~K100. При выходе за пределы диапазона ПЛК будет оперировать минимальным или максимальным допустимым значением. Время дискретизации может быть изменено во время работы, измененные значения вступят в силу до запуска цикла программы с данной командой. $\mathbf{S_1}$+1 указывает последнюю вычисленную ПЛК частоту на входе (только чтение). Шаг: 1 Гц. Диапазон: ±10 кГц. $\mathbf{S_1}$+2 и $\mathbf{S_1}$+3 показывают накопленное число импульсов для 32-битных данных (только чтение).
6. $\mathbf{S_1}$ +0 определяет время дискретизации. Заданное значение времени дискретизации рекомендуется увеличить при увеличении частоты на входе для повышения точности определения скорости. Например, зададим $\mathbf{S_1}$ +0 как K1 для диапазона скоростей 1 Гц~1 кГц, K10 для диапазона 10 Гц~10 кГц, K100 для 100 Гц~10 кГц. Для однофазного входа максимальная частота составит 10 кГц; для двухфазных 2 входов – 2 кГц.
7. **D** занимает 3 последовательных 16-битных регистра. **D** +0 определяет выходное пропорциональное значение. Допустимый диапазон: K1 (1%) ~ K10000 (10000%). При выходе за пределы диапазона, ПЛК будет оперировать минимальным или максимальным допустимым значением, как заданным. Выходная пропорция может быть изменена во время работы, измененные значения вступят в силу до запуска цикла программы с данной командой. **D**+2 и **D**+1 показывают выходную скорость для 32-битных данных. Шаг: 1 Гц. Диапазон: ±100 кГц.
8. Вычисленная ПЛК частота на входе будет увеличиваться согласно выходной пропорции **D**+0, затем ПЛК будет генерировать текущую выходную частоту. ПЛК округляет вниз значение, т.е. если вычисленный результат меньше 1 Гц, ПЛК выведет частоту 0 Гц. Например, входная скорость: 10 Гц, Выходная пропорция: K5 (5%), результат: 10 x 0.05 = 0.5 Гц. Выходная частота будет 0 Гц; если изменить пропорцию на K15 (15%), результат будет 10 x 0.15 = 1.5 Гц. В этом случае, выходная частота 1 Гц.

Пример программы:

1. Если D0 равно K2 и D10 равно K100:
   Когда определенная частота на (X0, X1) увеличится на 10 Гц (D1 = K10), (Y0, Y1) будет выдавать импульсы +10 Гц (D12, D11 = K10); при определенной частоте -10 Гц (D1 = K-10), (Y0, Y1) выдает импульсы -10 Гц (D12, D11 = K-10)
2. Если D0 равно K2 и D10 равно K1000:
   Когда определенная частота на (X0, X1) увеличится на 10 Гц (D1 = K10), (Y0, Y1) будет выдавать импульсы +100 Гц (D12, D11 = K100); при определенной частоте -100 Гц (D1 = K-100), (Y0, Y1) выдает импульсы -100 Гц (D12, D11 = K-100)
3. Если D0 равно K10 и D10 равно K10:
   Когда определенная частота на (X0, X1) увеличится на 10 Гц (D1 = K10), (Y0, Y1) будет выдавать импульсы +1 Гц (D12, D11 = K1); при определенной частоте -10 Гц (D1 = K-10), (Y0, Y1) выдает импульсы -1 Гц (D12, D11 = K-1)



API	Команда		Операнды	Функция	Контроллеры			
215~217	D	LD#	S_1 S_2	Логические операции контактного типа	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	LD#: 5 шагов DLD#: 9 шагов
S_1					*	*	*	*	*	*	*	*	*	*	*	
S_2					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Исходные данные 1 **S_2**: Исходные данные 2

Описание:

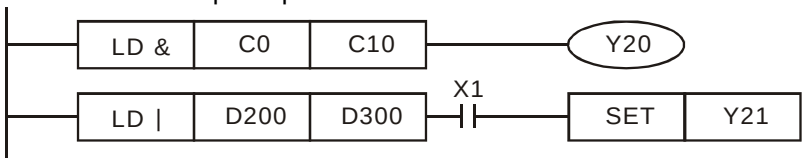
1. Эта инструкция располагается в левой части строки команды и проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0” , то выполняется правая часть команды. Если результат равен “= 0”, то происходит переход к следующей строке.

API	16-битная команда	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
215	LD&	**D**LD&	S_1 & S_2≠0	S_1 & S_2=0
216	LD|	**D**LD|	S_1 | S_2≠0	S_1 | S_2=0
217	LD^	**D**LD^	S_1 ^ S_2≠0	S_1 ^ S_2=0

2. Операции: & : логическое И, | : логическое ИЛИ, ^ : логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
3. При использовании с командой 32-битных счетчиков (C200 ~ C254), необходимо применение 32-битных команд (DLD#). При применении 16-битной команды (LD#) возникнет ошибка и будет мигать индикатор ERROR на панели MPU.

Пример программы:

1. Когда результат операции логическое И между C0 и C10 ≠ 0, Y20 включается.
2. Когда результат операции логическое ИЛИ между D200 и D300 ≠ 0 и X1 включен, Y21 включен и оба зафиксированы.



API	Команда		Операнды	Функция	Контроллеры			
218~220	D	AND#	S_1 S_2	Последовательные логические операции контактного типа	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	AND#: 5 шагов DAND#: 9 шагов
S_1					*	*	*	*	*	*	*	*	*	*	*	
S_2					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Исходные данные 1 **S_2**: Исходные данные 2

Описание:

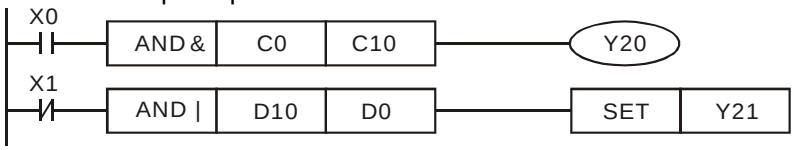
1. Эта инструкция располагается в левой части строки команды и проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0” , то выполняется правая часть команды. Если результат равен “= 0”, то происходит переход к следующей строке.

API	16-битная команда	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
218	AND&	**D**AND&	S_1 & S_2≠0	S_1 & S_2=0
219	AND|	**D**AND|	S_1 | S_2≠0	S_1 | S_2=0
220	AND^	**D**AND^	S_1 ^ S_2≠0	S_1 ^ S_2=0

2. Операция: & : логическое И, | : логическое ИЛИ, ^ : логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
3. При использовании с командой 32-битных счетчиков (C200 ~ C254), необходимо применение 32-битных команд (DAND#). При применении 16-битной команды (AND#) возникнет ошибка и будет мигать индикатор ERROR на панели MPU.

Пример программы:

1. Когда X0 включен и результат операции логическое И между C0 и C10 ≠ 0, Y20 включен
2. Когда X1 выключен и результат операции логическое ИЛИ между D10 и D0 ≠ 0, Y21 включен и зафиксирован



API	Команда	Операнды	Функция	Контроллеры
221~223	OR#	S_1 S_2	Параллельные логические операции контактного типа	ES2/EX2 | SS2 | SA2/SE | SX2

Операнд \ Тип	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	OR#: 5 шагов DOR#: 9 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Исходные данные 1 **S_2**: Исходные данные 2

Описание:

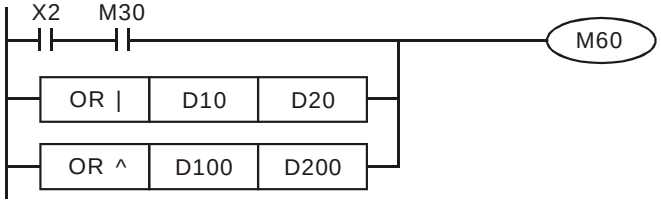
1. Эта инструкция проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0”, инструкция включает OR-контакт.

API	16-битная команда	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
221	OR&	**D**OR&	S_1 & S_2≠0	S_1 & S_2=0
222	OR|	**D**OR|	S_1 | S_2≠0	S_1 | S_2=0
223	OR^	**D**OR^	S_1 ^ S_2≠0	S_1 ^ S_2=0

2. Операция: & : логическое И, | : логическое ИЛИ, ^ : логическое ИСКЛЮЧАЮЩЕЕ ИЛИ
3. При использовании с командой 32-битных счетчиков (C200 ~ C254), необходимо применение 32-битных команд (DOR#). При применении 16-битной команды (OR#) возникнет ошибка и будет мигать индикатор ERROR на панели MPU.

Пример программы:

M60 включен, когда включены оба X2 и M30 или, 1: результат операции логическое ИЛИ между D10 и D20 ≠ 0, 2: результат операции логическое ИСКЛЮЧАЮЩЕЕ ИЛИ между D100 и D200 ≠ 0.



API	Команда	Операнды	Функция	Контроллеры
224~230	D LD※	S_1 S_2	Операции сравнения контактного типа	ES2/EX2 | SS2 | SA2/SE | SX2

Операнд \ Тип	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	LD※: 5 шагов DLD※: 9 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:

S_1: Исходные данные 1 **S_2**: Исходные данные 2

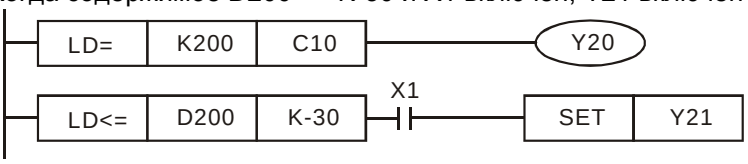
Описание:

1. Эта инструкция располагается в левой части строки команды и проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0” , то выполняется правая часть команды. Если результат равен “= 0”, то происходит переход к следующей строке.

API	16-битная команда	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
224	LD=	DLD=	$S_1 = S_2$	$S_1 \neq S_2$
225	LD>	DLD>	$S_1 > S_2$	$S_1 \leqq S_2$
226	LD<	DLD<	$S_1 < S_2$	$S_1 \geqq S_2$
228	LD<>	DLD<>	$S_1 \neq S_2$	$S_1 = S_2$
229	LD<=	DLD<=	$S_1 \leqq S_2$	$S_1 > S_2$
230	LD>=	DLD>=	$S_1 \geqq S_2$	$S_1 < S_2$

2. Когда самый старший бит MSB (16-битная команда: b15, 32-битная команда: b31) S_1 и S_2 равен 1, значение сравнения будет рассматриваться как отрицательное.
3. При использовании с командой 32-битных счетчиков (C200 ~ C254), необходимо применение 32-битных команд (DLD※). При применении 16-битной команды (LD※) возникнет ошибка и будет мигать индикатор ERROR на панели MPU.

Пример программы:
1. Когда содержимое C10 = K200, Y20 включен.
2. Когда содержимое D200 => K-30 и X1 включен, Y21 включен и зафиксирован.



API	Команда			Операнды	Функция
232~238	D	AND※		S_1 S_2	Последовательное сравнение контактного типа

Контроллеры			
ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	AND※: 5 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	DAND※: 9 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

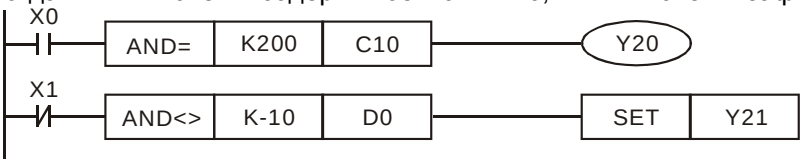
Операнды:
S_1: Исходные данные 1 S_2: Исходные данные 2

Описание:
1. Эта инструкция располагается в левой части строки команды и проводит логические операции между содержимым S_1 и S_2. Если результат “≠ 0” , то выполняется правая часть команды. Если результат равен “= 0”, то происходит переход к следующей строке.

API	16-битная команда	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
232	AND=	DAND=	$S_1 = S_2$	$S_1 \neq S_2$
233	AND>	DAND>	$S_1 > S_2$	$S_1 \leqq S_2$
234	AND<	DAND<	$S_1 < S_2$	$S_1 \geqq S_2$
236	AND<>	DAND<>	$S_1 \neq S_2$	$S_1 = S_2$
237	AND<=	DAND<=	$S_1 \leqq S_2$	$S_1 > S_2$
238	AND>=	DAND>=	$S_1 \geqq S_2$	$S_1 < S_2$

2. Когда самый старший бит MSB (16-битная команда: b15, 32-битная команда: b31) S_1 и S_2 равен 1, значение сравнения будет рассматриваться как отрицательное.
3. При использовании с командой 32-битных счетчиков (C200 ~ C254), необходимо применение 32-битных команд (DAND※). При применении 16-битной команды (AND※) возникнет ошибка и будет мигать индикатор ERROR на панели MPU.

Пример программы:
1. Когда X0 включен и содержимое C10 = K200, Y20 включен
2. Когда X1 выключен и содержимое D0 ≠ K-10, Y21 включен и зафиксирован.



API	Команда		Операнды	Функция	Контроллеры			
240~246	D	OR※	S1 S2	Параллельное сравнение контактного типа	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*	*	*	OR※: 5 шагов
S_2					*	*	*	*	*	*	*	*	*	*	*	DOR※: 9 шагов

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Исходные данные 1 **S_2**: Исходные данные 2

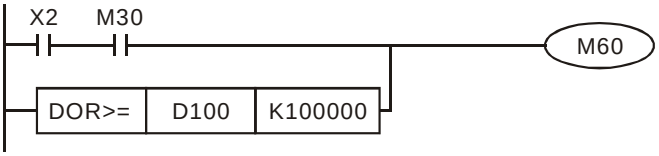
Описание:

1. Эта инструкция проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0”, инструкция включает OR-контакт.

API	16-битная команда	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
240	OR=	DOR=	$S_1=S_2$	$S_1\neq S_2$
241	OR>	DOR>	$S_1>S_2$	$S_1\leqq S_2$
242	OR<	DOR<	$S_1<S_2$	$S_1\geqq S_2$
244	OR<>	DOR<>	$S_1\neq S_2$	$S_1=S_2$
245	OR<=	DOR<=	$S_1\leqq S_2$	$S_1>S_2$
246	OR>=	DOR>=	$S_1\geqq S_2$	$S_1<S_2$

2. Когда самый старший бит MSB (16-битная команда: b15, 32-битная команда: b31) **S_1** и **S_2** равен 1, значение сравнения будет рассматриваться как отрицательное.
3. При использовании с командой 32-битных счетчиков (C200 ~ C254), необходимо применение 32-битных команд (DOR※). При применении 16-битной команды (OR※) возникнет ошибка и будет мигать индикатор ERROR на панели MPU.

Пример программы:



M60 включен, когда включены оба X2 и M30 или когда содержимое 32-битного регистра D100 (D101) ≥ K100 000.

API	Команда	Операнды	Функция	Контроллеры			
258	ATMR	S1 S2	Таймер контактного типа	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*					ATMR: 5 шагов
S_2					*	*							*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

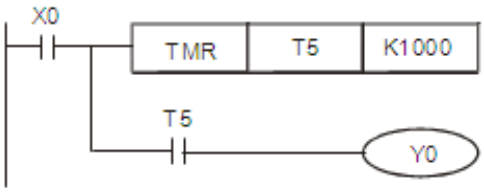
Операнды:
S_1: Номер таймера (T0~T255) **S_2**: Заданное значение (K0~K32767, D0~D9999)

Описание:

1. Поддерживается моделями ПЛК серий DVP-ES2/EX2 версии 3.20, DVP-SS2 версии 3.00, DVP-SA2 версии 2.60, DVP-SE версии 1.20, DVP-SX2 версии 2.40 и выше.
2. После включения команды таймер начинает отсчет времени. Когда таймер достигает заданного значения, Н/О контакт включается, а Н/З контакт выключается.

Пример программы:
Когда Н/О контакт X0 включен, таймер T5 начинает отсчитывать время. Если значение таймера больше или равно K1000, Н/О контакт Y0 включается.

Релейно-контактная схема применения команды TMR:



Релейно-контактная схема применения команды ATMR:



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/ SE	SX2
266	D	BOUT	D n	Переключение заданного бита в слове				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BOUT: 5 шагов DBOUT: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

D: Выход **n**: Номер бита (K0~K15 для 16-битной команды; K0~K31 для 32-битной команды)

Описание:

1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Команда BOUT переключает бит, указанный в операнде **n,** в слове на выходе**.**

Состояние катушек и соответствующих контактов:

Оценка результата	Команда BOUT		
	Катушка реле	Соответствующие контакты выхода	
		Н/О контакт	Н/З контакт
ЛОЖЬ	ВЫКЛ	Выкл.	Вкл.
ИСТИНА	ВКЛ	Вкл.	Выкл.

Пример программы:



Команда:		Операция:
LDI	X0	Загрузка Н/З контакта X0
AND	X1	Последовательный Н/О контакт X1.
BOUT	**K4Y0 D0**	**Когда D0 = k1, переключает выход Y1** **Когда D0 = k2 - выход Y2**

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/ SE	SX2
267	D	BSET	D n	Переключение заданного бита в слове с фиксацией				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BSET: 5 шагов DBSET: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2	ES2/EX2	SS2	SA2/ SE	SX2

Операнды:

D: Выход **n**: Операнд, указывающий номер бита

Описание:

1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.

3. Команда BSET переключает и фиксирует бит, указанный операндом **n**. Для сброса зафиксированного состояния включения используйте команду BRST.

Пример программы:



Команда: Операция:
LDI X0 Загрузка Н/З контакта X0
AND X1 Последовательный Н/О контакт X1.
BSET K4Y0 D0 D0 = k1, Y1 переключается и фиксируется. Если D0 = k2, Y2 перекл. и фиксируется.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
268	D	BRST	(D) (n)	Сброс заданного бита в слове				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
D								*	*	*	*	*	*			BRST: 5 шагов DBRST: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
D: Выход **n**: Операнд, указывающий номер сбрасываемого бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BRST сбрасывает (выключает) выход, указанный операндом **n**.

Пример программы:



Команда: Операция:
LDI X0 Загрузка Н/О контакта X0
BRST K4Y0 D0 Если D0 = k1, Y1 сбрасывается. Если D0 = k2, тоY2

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
269	D	BLD	(S) (n)	Установка Н/О контакта с состоянием по заданному биту				

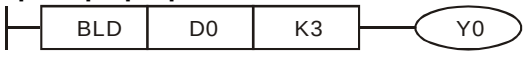
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BLD: 5 шагов DBLD: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **n**: Номер бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BLD используется для создания Н/О контакта, состояние которого определяется заданным битом **n** в источнике **S**, т.е. если бит, заданный **n,** включен, Н/О контакт замыкается и наоборот.

Пример программы:



Команда: Операция:
BLD D0 K3 Загрузка Н/О контакта с состоянием согласно биту 3 в D0
OUT Y0 Катушка Y0

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
270	D	BLDI	(S) (n)	Установка H/З контакта с состоянием по заданному биту				

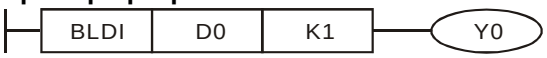
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BLDI: 5 шагов DBLDI: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **n**: Номер бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BLDI используется для создания H/З контакта, состояние которого определяется заданным битом **n** в источнике **S**, т.е. если бит, заданный **n,** включен, H/З контакт размыкается и наоборот.

Пример программы:



Команда:			Операция:
BLDI	**D0**	**K1**	**Загрузка H/З контакта с состоянием согласно биту 1 в D0**
OUT	Y0		Катушка Y0

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
271	D	BAND	(S) (n)	Последовательное подключение H/O контакта с состоянием по заданному биту				

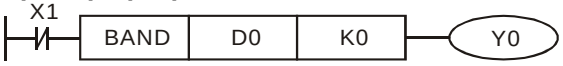
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BAND: 5 шагов DBAND: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **n**: Номер бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BAND используется для последовательного подключения H/O контакта, состояние которого определяется заданным битом **n** в источнике **S**, т.е. если бит, заданный **n,** включен, H/O замыкается и наоборот.

Пример программы:



Команда:			Операция:
LDI	X1		Загрузка H/З контакта X1
BAND	**D0**	**K0**	**Последовательное подключение H/O контакта с состоянием согласно биту 0 в D0**
OUT	Y0		Катушка Y0

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
272	D	BANI	S n	Последовательное подключение Н/З с состоянием по заданному биту				

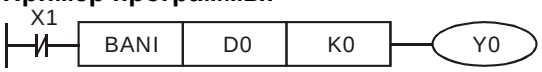
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BANI: 5 шагов DBANI: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **n**: Номер бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BANI используется для последовательного подключения Н/З контакта, состояние которого определяется заданным битом **n** в источнике **S**, т.е. если бит, заданный **n,** включен, Н/З размыкается и наоборот.

Пример программы:



Команда:			Операция:
LDI	X1		Загрузка Н/З контакта X1
BANI	**D0**	**K0**	**Последовательное подключение Н/З контакта с состоянием согласно биту 0 в D0**
OUT	Y0		Катушка Y0

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2/SE	SX2
273	D	BOR	S n	Параллельное подключение Н/О контакта с состоянием по заданному биту				

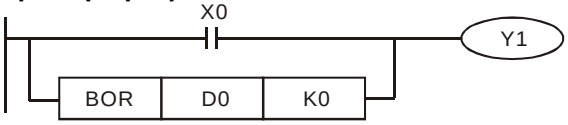
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BOR: 5 шагов DBOR: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **n**: Номер бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BOR используется для параллельного подключения Н/О контакта, состояние которого определяется заданным битом **n** в источнике **S**, т.е. если бит, заданный **n,** включен, Н/О замыкается и наоборот.

Пример программы:



Команда:			Операция:
LD	X0		Загрузка Н/О контакта X0
BOR	**D0**	**K0**	**Параллельное подключение Н/О контакта с состоянием согласно биту 0 в D0**
OUT	Y1		Катушка Y1

API	Команда		Операнды	Функция	Контроллеры			
274	D	BORI	(S) (n)	Параллельное подключение Н/З контакта с состоянием по заданному биту	ES2/EX2	SS2	SA2/SE	SX2

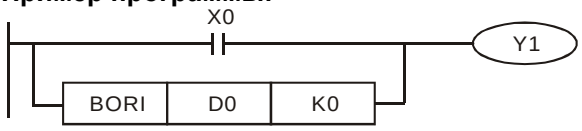
Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S								*	*	*	*	*	*			BORI: 5 шагов DBORI: 9 шагов
n					*	*	*	*	*	*	*	*	*	*	*	

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S: Источник **n**: Номер бита

Описание:
1. Для ES2/EX2: функция поддерживается версией V1.20 или выше.
2. Диапазон **n**: K0~K15 для 16-битной команды; K0~K31 для 32-битной команды.
3. Команда BORI используется для параллельного подключения Н/З контакта, состояние которого определяется заданным битом **n** в источнике **S**, т.е. если бит, заданный **n,** включен, Н/З размыкается и наоборот.

Пример программы:



Команда:			Операция:
LD	X0		Загрузка Н/О контакта X0
BORI	**D0**	**K0**	**Параллельное подключ. Н/З контакта с состоянием по биту 0 в D0**
OUT	Y1		Катушка Y1

API	Команда	Операнды	Функция	Контроллеры			
275~280	FLD※	(S1) (S2)	Операции сравнения контактного типа с плавающей запятой LD※	ES2/EX2	SS2	SA2/SE	SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FLD※: 9 шагов
S_2											*	*	*			

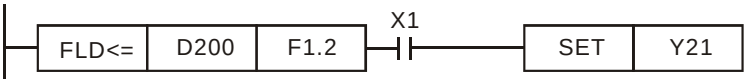
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Источник 1 S_2: Источник 2

Описание:
1. Эта инструкция располагается в левой части строки команды и проводит логические операции между содержимым S_1 и S_2. Если результат “≠ 0” , то выполняется правая часть команды. Если результат равен “= 0”, то происходит переход к следующей строке.
2. Пользователь может указать значения с плавающей запятой непосредственно в операндах S_1 и S_2 (например, F1.2) или сохранить значение с плавающей запятой в регистрах D для дальнейших операций.

API	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
275	FLD=	$S_1 = S_2$	$S_1 \neq S_2$
276	FLD>	$S_1 > S_2$	$S_1 \leqq S_2$
277	FLD<	$S_1 < S_2$	$S_1 \geqq S_2$
278	FLD<>	$S_1 \neq S_2$	$S_1 = S_2$
279	FLD<=	$S_1 \leqq S_2$	$S_1 > S_2$
280	FLD>=	$S_1 \geqq S_2$	$S_1 < S_2$

Пример программы:



Когда содержимое в D200(D201) ≤ F1.2 и X1 вкл, Y21 включается и фиксируется.

API	Команда	Операнды	Функция	Контроллеры
281~286	FAND※	(S1) (S2)	Операции сравнения контактного типа (последовательное соединение) с плавающей запятой AND※	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FAND※: 9 шагов
S_2											*	*	*			

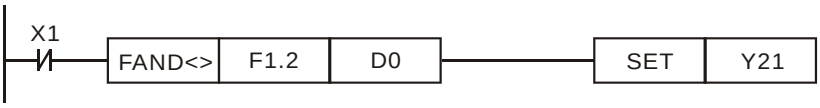
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Источник 1 **S_2**: Источник 2

Описание:
1. Эта инструкция располагается в левой части строки команды и проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0” , то выполняется правая часть команды. Если результат равен “= 0”, то происходит переход к следующей строке.
2. Пользователь может указать значения с плавающей запятой непосредственно в операндах **S_1** и **S_2** (например, F1.2) или сохранить значение с плавающей запятой в регистрах D для дальнейших операций.

API	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
281	FAND=	$S_1=S_2$	$S_1\neq S_2$
282	FAND>	$S_1>S_2$	$S_1\leqq S_2$
283	FAND<	$S_1<S_2$	$S_1\geqq S_2$
284	FAND<>	$S_1\neq S_2$	$S_1=S_2$
285	FAND<=	$S_1\leqq S_2$	$S_1>S_2$
286	FAND>=	$S_1\geqq S_2$	$S_1<S_2$

Пример программы:



Когда X1 выключен, и содержимое D100(D101) ≠ F1.2, Y21 включается и фиксируется.

API	Команда	Операнды	Функция	Контроллеры
287~292	FOR※	(S1) (S2)	Операции сравнения контактного типа (параллельное соединение) с плавающей запятой OR※	ES2/EX2 | SS2 | SA2/SE | SX2

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1											*	*	*			FOR※: 9 шагов
S_2											*	*	*			

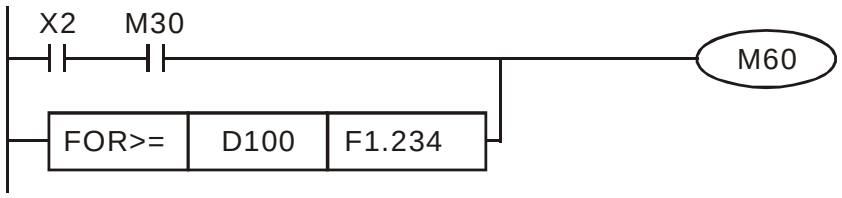
ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2	ES2/EX2	SS2	SA2/SE	SX2

Операнды:
S_1: Источник 1 **S_2**: Источник 2

Описание:
1. Эта инструкция проводит логические операции между содержимым **S_1** и **S_2**. Если результат “≠ 0”, инструкция включает OR-контакт.
2. Пользователь может указать значения с плавающей запятой непосредственно в операндах **S_1** и **S_2** (например, F1.2) или сохранить значение с плавающей запятой в регистрах D для дальнейших операций.

API	32-битная команда	Контакт замкнут, если	Контакт разомкнут, если
287	FOR=	$S_1 = S_2$	$S_1 \neq S_2$
288	FOR>	$S_1 > S_2$	$S_1 \leqq S_2$
289	FOR<	$S_1 < S_2$	$S_1 \geqq S_2$
290	FOR<>	$S_1 \neq S_2$	$S_1 = S_2$
291	FOR<=	$S_1 \leqq S_2$	$S_1 > S_2$
292	FOR>=	$S_1 \geqq S_2$	$S_1 < S_2$

Пример программы:



Когда X2 и M30 оба включены и содержимое в D100(D101) ≥ F1.234, M60 включен.

API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2 SE	SX2
296~301	D	LDZ※	S_1 S_2	Сравнение контактного типа абсолютных значений LDZ※				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			LDZ※: 7 шагов
S_2					*	*	*	*	*	*	*	*	*			DLDZ※: 13 шагов
S_3					*	*	*	*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2

Операнды:

S_1: Источник данных 1 S_2: Источник данных 2 S_3: Источник данных 3

Описание:

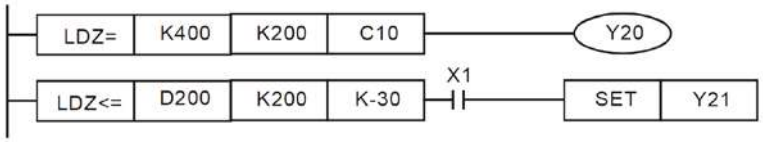
1. Поддерживается моделями ПЛК серий DVP-ES2/EX2 версии 3.20, DVP-SS2 версии 3.00, DVP-SA2 версии 2.60, DVP-SE версии 1.20, DVP-SX2 версии 2.40 и выше.
2. Абсолютное значение разницы между S_1 и S_2 сравнивается с абсолютным значением S_3. Например, для команды LDZ=: если результат сравнения разницы S_1 и S_2 равен S_3, условие команды выполняется, если не равен, то не выполняется.
3. Инструкция ставится слева.

API	16-битная команда	32-битная команда	Результат сравнения	
			ВКЛ	ВЫКЛ
296	LDZ>	DLDZ>	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert \leq \lvert S_3 \rvert$
297	LDZ>=	DLDZ>=	$\lvert S_1 - S_2 \rvert \geq \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$
298	LDZ<	DLDZ<	$\lvert S_1 - S_2 \rvert < \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert \geq \lvert S_3 \rvert$
299	LDZ<=	DLDZ<=	$\lvert S_1 - S_2 \rvert \leq \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert > \lvert S_3 \rvert$
300	LDZ=	DLDZ=	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$
301	LDZ<>	DLDZ<>	$\lvert S_1 - S_2 \rvert \neq \lvert S_3 \rvert$	$\lvert S_1 - S_2 \rvert = \lvert S_3 \rvert$

4. Если значимые биты S_1, S_2 и S_3 = 1, значения S_1, S_2 и S_3 будут отрицательными.
5. 32-битный счетчик (C200~) должен использоваться с 32-битной командой DLDZ※.

При использовании16-битной команды LDZ※ в программе возникает ошибка и мигает индикатор ERROR.

Пример программы:



1. Если значение C10 равно K200 или K-200, Y20 включается.
2. Если значение D200 меньше или равно K230 и больше или равно K170, а также включен X1, Y21 включается и фиксируется.

				Контроллеры			
302~307	D ANDZ※	S1 S2 S3	Сравнение контактного типа абсолютных значений ANDZ※	ES2/EX2	SS2	SA2 SE	SX2

Тип / Операнд	Биты				Слова										Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	
S_1					*	*	*	*	*	*	*	*	*			ANDZ※: 7 шагов
S_2					*	*	*	*	*	*	*	*	*			DANDZ※: 13 шагов
S_3					*	*	*	*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2

Операнды:

S_1: Источник данных 1 **S_2**: Источник данных 2 **S_3**: Источник данных 3

Описание:

1. Поддерживается моделями ПЛК серий DVP-ES2/EX2 версии 3.20, DVP-SS2 версии 3.00, DVP-SA2 версии 2.60, DVP-SE версии 1.20, DVP-SX2 версии 2.40 и выше.
2. Абсолютное значение разницы между **S_1** и **S_2** сравнивается с абсолютным значением **S_3**. Например, для команды ANDZ=: если результат сравнения разницы **S_1** и **S_2** равен **S_3**, условие команды выполняется, если не равен, то не выполняется.
3. Инструкция ANDZ※ используется для последовательного подключения.

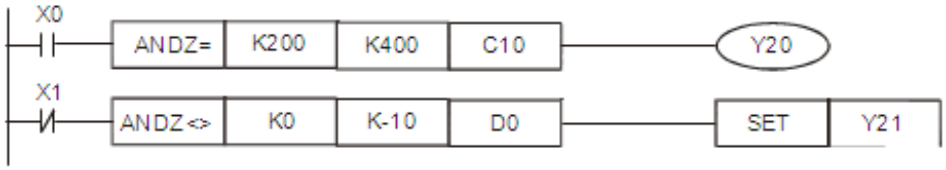
API	16-битная команда	32-битная команда	Результат сравнения	
			ВКЛ	ВЫКЛ
302	ANDZ >	**D**ANDZ >	$\|S_1 - S_2\| > \|S_3\|$	$\|S_1 - S_2\| \leqq \|S_3\|$
303	ANDZ > =	**D**ANDZ > =	$\|S_1 - S_2\| \geqq \|S_3\|$	$\|S_1 - S_2\| < \|S_3\|$
304	ANDZ <	**D**ANDZ <	$\|S_1 - S_2\| < \|S_3\|$	$\|S_1 - S_2\| \geqq \|S_3\|$
305	ANDZ < =	**D**ANDZ < =	$\|S_1 - S_2\| \leqq \|S_3\|$	$\|S_1 - S_2\| > \|S_3\|$
306	ANDZ =	**D**ANDZ =	$\|S_1 - S_2\| = \|S_3\|$	$\|S_1 - S_2\| \neq \|S_3\|$
307	ANDZ < >	**D**ANDZ < >	$\|S_1 - S_2\| \neq \|S_3\|$	$\|S_1 - S_2\| = \|S_3\|$

4. Если значимые биты **S_1**, **S_2** и **S_3** равны 1, значения **S_1**, **S_2** и **S_3** будут негативными.
5. 32-битный счетчик (C200~) должен использоваться с 32-битной командой DANDZ※.

При использовании16-битной команды ANDZ※ в программе возникает ошибка и мигает индикатор ERROR.

Пример программы:

1. При включенном X0, если значение C10 равно K200 или K-200, Y20 включается.
2. При выключенном X1, если значение D0 не равно K10 или K-10, Y21 включается и фиксируется.



API	Команда		Операнды	Функция	Контроллеры			
					ES2/EX2	SS2	SA2 SE	SX2
308~313	D	ORZ※	(S1) (S2) (S3)	Сравнение контактного типа абсолютных значений ORZ※				

Тип / Операнд	Биты				Слова											Шаги программы
	X	Y	M	S	K	H	KnX	KnY	KnM	KnS	T	C	D	E	F	ORZ※: 7 шагов
S_1					*	*	*	*	*	*	*	*	*			DORZ※: 13 шагов
S_2					*	*	*	*	*	*	*	*	*			
S_3					*	*	*	*	*	*	*	*	*			

ИМПУЛЬС				16-бит				32-бит			
ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2	ES2/EX2	SS2	SA2 SE	SX2

Операнды:

S_1: Источник данных 1 **S_2**: Источник данных 2 **S_3**: Источник данных 3

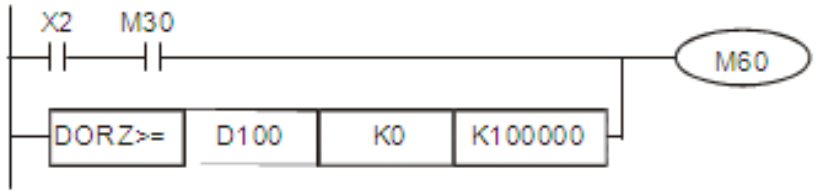
Описание:

1. Поддерживается моделями ПЛК серий DVP-ES2/EX2 версии 3.20, DVP-SS2 версии 3.00, DVP-SA2 версии 2.60, DVP-SE версии 1.20, DVP-SX2 версии 2.40 и выше.
2. 2. Абсолютное значение разницы между **S_1** и **S_2** сравнивается с абсолютным значением **S_3**. Например, для команды ORZ=: если результат сравнения разницы **S_1** и **S_2** равен **S_3**, условие команды выполняется, если не равен, не выполняется.
3. Команда ORZ※ используется для параллельного подключения.

API	16-битная команда	32-битная команда	Результат сравнения	
			ВКЛ	ВЫКЛ
308	ORZ >	**D**ORZ >	$\|S_1 - S_2\| > \|S_3\|$	$\|S_1 - S_2\| \leqq \|S_3\|$
309	ORZ > =	**D**ORZ > =	$\|S_1 - S_2\| \geqq \|S_3\|$	$\|S_1 - S_2\| < \|S_3\|$
310	ORZ <	**D**ORZ <	$\|S_1 - S_2\| < \|S_3\|$	$\|S_1 - S_2\| \geqq \|S_3\|$
311	ORZ < =	**D**ORZ < =	$\|S_1 - S_2\| \leqq \|S_3\|$	$\|S_1 - S_2\| > \|S_3\|$
312	ORZ =	**D**ORZ =	$\|S_1 - S_2\| = \|S_3\|$	$\|S_1 - S_2\| \neq \|S_3\|$
313	ORZ < >	**D**ORZ < >	$\|S_1 - S_2\| \neq \|S_3\|$	$\|S_1 - S_2\| = \|S_3\|$

4. Если значения значимых битов **S_1**, **S_2** и **S_3** равны 1, значения **S_1**, **S_2** и **S_3** будут негативными.
5. 32-битный счетчик (C200~) должен использоваться с 32-битной командой DORZ※.

При использовании16-битной команды ORZ※ в программе возникает ошибка и мигает индикатор ERROR.



Пример программы:
При включенных X2 и M30 или, если значение 32-битного регистра (D101, D100) больше или равно K100000, или меньше или равно K-100000, M60 включается.

Параметры коммуникации

В данной главе рассказывается о назначении и функционале и порядке работы коммуникационных портов ПЛК.

4.1 Коммуникационные порты

DVP-ES2/EX2/SA2/SE/SX2 имеет 3 коммуникационных порта (COM1~COM3), а DVP-SS2 - 2 порта (COM1~COM2). COM порты последних моделей поддерживают коммуникационный формат DELTA Q-link для связи с панелями оператора (HMI), позволяющий увеличить скорость связи.

COM1: Коммуникационный порт RS-232. COM1 может использоваться в качестве ведущего (Master) или ведомого (Slave), а также является основным COM портом для программирования ПЛК. (кроме DVP-SE).
COM2: Коммуникационный порт RS-485. COM2 может использоваться в качестве ведущего (Master) или ведомого (Slave).
COM3 (ES2/EX2/SA2/SE): Коммуникационный порт RS-485. COM3 может использоваться в качестве ведущего (Master) или ведомого (Slave) (для DVP-ES2-C, COM3 - порт CANopen)
COM3 (SX2): Переходник от USB порта до RS-232 порта. COM3 используется только в Slave-режиме.
Все вышеуказанные порты поддерживают протокол связи Modbus в форматах ASCII или RTU.
USB (COM1) (SE): Коммуникационный порт USB, используется только в Slave-режиме. Режим и формат связи не может быть изменен.

Спецификация портов:

Параметр \ COM порт	RS-232 (COM1)	RS-485 (COM2)	RS-485 (COM3)	RS-485 (SX2 COM3)
Скорость (бит/с)	110~115200	110~921000		110~115200
Размер данных	7~8 бит			
Четность	Четное / нечетное / нет контроля четности			
Размер стоп-бита	1~2 бит			
Установочные регистры	D1036	D1120	D1109	
Сохранение формата связи	M1138	M1120	M1136	
Режим ASCII	Доступно в Master/Slave-режимах			Доступно в Slave-режиме
Режим RTU				
Выбор режима ASCII/RTU	M1139	M1143	M1320	
Адрес связи в Slave-режиме	D1121		D1255	
Размер доступных данных (ASCII)	100 регистров			
Размер доступных данных (RTU)	100 регистров			

Настройки по умолчанию для всех COM портов:

- Modbus ASCII
- 7 бит данных
- 1 стоп-бит
- Четность
- Скорость: 9600 бит/с

4.2 Протокол связи в ASCII-режиме

Структура данных: 9600 бит/с (скорость), 7 (бит данных), Четный (четность), 1 (стартовый бит), 1 (стоп-бит)

Имя области	Содержимое	Описание
Стартовый бит	STX	Стартовый бит ':' (3AH)
Коммуникационный адрес	ADR 1	Адрес содержит 2 ASCII кода
	ADR 0	
Код команды	CMD 1	Код команды содержит 2 ASCII кода
	CMD 0	
Данные	DATA (0)	Данные содержат 2n ASCII кодов, n≤205
	DATA (1)	
	…….	
	DATA (n-1)	
Контрольная сумма LRC	LRC CHK 1	Контрольная сумма LRC содержит 2 ASCII кода
	LRC CHK 0	
Стоп-бит	END1	Стоп-бит содержит 2 ASCII кода END1 = CR (0DH), END0 = LF (0AH)
	END0	

Таблица соответствия шестнадцатеричных значений и ASCII кодов:

ASCII	“0“	“1“	“2“	“3“	“4“	“5“	“6“	“7“
Hex	30H	31H	32H	33H	34H	35H	36H	37H
ASCII	“8“	“9“	“A“	“B“	“C“	“D“	“E“	“F“
Hex	38H	39H	41H	42H	43H	44H	45H	46H

4.2.1 ADR (Коммуникационный адрес)

Допустимые коммуникационные адреса находятся в диапазоне 0~254. Адрес, равный 0, служит для трансляции всем ПЛК. ПЛК в этом случае не будут отвечать ведущему устройству. Например, ASCII код для адреса 16 (десятичный формат) (16 десятичный = 10 шестнадцатеричный)
(ADR 1, ADR 0)=’1’,’0’⇨’1’=31H, ‘0’ = 30H

4.2.2 CMD (Код команды) и DATA (символы данных)

Содержимое доступных данных зависит от командного кода.
Доступные настройки командного кода:

Код CMD(Hex)	Описание	Операнд
01 (01 H)	Чтение состояния контакта	S, Y, M, T, C
02 (02 H)	Чтение состояния контакта	S, X, Y, M,T, C
03 (03 H)	Чтение содержимого регистра	T, C, D
05 (05 H)	Состояние ВКЛ/ВЫКЛ одного контакта	S, Y, M, T, C
06 (06 H)	Установка значения одного регистра	T, C, D
15 (0F H)	Состояние ВКЛ/ВЫКЛ нескольких контактов	S, Y, M, T, C
16 (10 H)	Установка значения нескольких регистров	T, C, D
17 (11 H)	Получение информации о Slave-устройстве	нет
23 (17 H)	Одновременное чтение/запись данных при опросе в EASY PLC LINK	нет

Пример: Чтение T20~T27 (адреса: H0614~H61B) от Slave-устройства с ID#01(номер устройства)
ПК→ПЛК “: 01 03 06 14 00 08 DA CR LF”
Отправленное сообщение:

Имя области	ASCII	Hex
STX (стартовый символ)	:	3A
Адрес Slave-устройства	01	30 31
Код команды	03	30 33
Начальный адрес (старший байт)	06	30 36
Начальный адрес (младший байт)	14	31 34
Число точек (старший байт)	00	30 30
Число точек (младший байт)	08	30 38
Контрольная сумма LRC	DA	44 41
END (конечный символ)	CR LF	0D 0A

ПЛК→ПК “: 01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 C8 CR LF”
Ответное сообщение:

Имя области	ASCII	Hex
STX (стартовый символ)	:	3A
Адрес Slave-устройства	01	30 31
Код команды	03	30 33
Число байт	10	31 30
Данные (старший байт) (T20)	00	30 30
Данные (младший байт) (T20)	01	30 31
Данные (старший байт) (T21)	00	30 30
Данные (младший байт) (T21)	02	30 32
Данные (старший байт) (T22)	00	30 30
Данные (младший байт) (T22)	03	30 33
Данные (старший байт) (T23)	00	30 30
Данные (младший байт) (T23)	04	30 34
Данные (старший байт) (T24)	00	30 30
Данные (младший байт) (T24)	05	30 35
Данные (старший байт) (T25)	00	30 30
Данные (младший байт) (T25)	06	30 36
Данные (старший байт) (T26)	00	30 30
Данные (младший байт) (T26)	07	30 37
Данные (старший байт) (T27)	00	30 30
Данные (младший байт) (T27)	08	30 38
Контрольная сумма LRC	C8	43 38
END (конечный символ)	CR LF	0D 0A

4.2.3 LRC CHK (Контрольная сумма)

LRC (продольная проверка избыточности) рассчитывается путем суммирования

шестнадцатеричных значений всех данных, начиная с ADR1, и применением 2-х дополнений отрицаний суммы.

Пример: Чтение содержимого регистров с адресом 0401H. 01H+03H+04H+01H+00+01H=0AH. 2 дополнение отрицания 0AH: F6H

Имя области	ASCII	Hex
STX (стартовый символ)	:	3A
Адрес Slave-устройства	**01**	30 31
Код команды	**03**	30 33
Начальный адрес (старший байт)	**04**	30 34
Начальный адрес (младший байт)	**01**	30 31
Число точек (старший байт)	**00**	30 30
Число точек (младший байт)	**01**	30 31
Контрольная сумма LRC	F6	46 36
END (конечный символ)	CR LF	0D 0A

Ответ исключения:
ПЛК должен получать нормальный ответ от управляющего устройства. Но в ряде случаев ответ отсутствует, или ответ содержит ошибку (т.н. ответ исключения):

1. ПЛК не получил нормального ответа из-за ошибки коммуникации; в этом случае для управляющего устройства будет превышено допустимое время ожидания.
2. ПЛК принимает сообщение без ошибки связи, но не может его обработать, в этом случае управляющему устройству возвращается ответ исключения. В нем наиболее значимый бит имеет значение 1, а код ошибки объясняет условие возникновения исключительного ответа.

Пример ответа исключения с командным кодом 01H и кодом ошибки 02H:
Отправленное сообщение:

Имя области	ASCII	Hex
STX (стартовый символ)	:	3A
Адрес Slave-устройства	01	30 31
Код команды	01	30 31
Начальный адрес (старший байт)	04	30 34
Начальный адрес (младший байт)	00	30 30
Число точек (старший байт)	00	30 30
Число точек (младший байт)	10	31 30
Контрольная сумма LRC	EA	45 41
END (конечный символ)	CR LF	0D 0A

Ответное сообщение:

Имя области	ASCII	Hex
STX (стартовый символ)	:	3A
Адрес Slave-устройства	01	30 31
Функция	81	38 31
Код исключительного ответа	02	30 32
Контрольная сумма LRC	7C	37 43
END (конечный символ)	CR LF	0D 0A

Код исключения	Описание
01	Недопустимый Код команды: Код полученной ПЛК команды является для него недопустимым.
02	Недопустимый адрес: Адрес устройства, полученный в командном сообщении, является недопустимым для ПЛК.
03	Недопустимые данные: Данные, полученные в командном сообщении, являются недопустимыми для ПЛК.
07	1. Ошибка контрольной суммы - Некорректное значение контрольной суммы 2. Недопустимое командное сообщение - Команда слишком короткая или слишком длинная.

4.3 Протокол связи в RTU-режиме

Структура данных
9600 бит/с (скорость), 8 (бит данных), Четный (четность), 1 (стартовый бит), 1 (стоп-бит)

START	Нет входных данных ≥ 10 мс
Коммуникационный адрес	Коммуникационный адрес: 8-битовый двоичный адрес
Код команды	Код команды: 8-битовый двоичный адрес
Данные (n-1)	Содержимое данных: n × 8-bit BIN data, n≦202
…….	
Данные 0	
CRC CHK (младший байт)	Контрольная сумма CRC: 16-битная контрольная сумма CRC содержит 2 8-битных двоичных кода
CRC CHK (старший байт)	
END	Нет входных данных ≥ 10 мс

4.3.1 Адрес (Коммуникационный адрес)

Допустимые коммуникационные адреса находятся в диапазоне 0~254. Адрес, равный 0,

служит для трансляции всем ПЛК. ПЛК в этом случае не будут отвечать ведущему устройству.

Например, коммуникационный адрес 10 (шестнадцатеричный формат) будет 16 (десятичный формат) (16 десятичный = 10 шестнадцатеричный)

4.3.2 CMD (Код команды) и DATA (символы данных)

Содержимое доступных данных зависит от командного кода. См. раздел 4.2.2 данной главы.

Пример: чтение последовательных 8 слов по адресу 0614H~H61B (T20~T27) от Slave-ПЛК с ID#1.

ПК→ПЛК " 01 03 06 14 00 08 04 80"

Отправленное сообщение:

Имя области	Пример (Hex)
START	Нет входных данных ⩾ 10 мс
Адрес Slave-устройства	01
Код команды	03
Стартовый адрес	06
	14
Число точек	00
	08
CRC CHK (младший байт)	04
CRC CHK (старший байт)	80
END	Нет входных данных ⩾ 10 мс

ПЛК→ПК " 01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 72 98"

Ответное сообщение:

Имя области	Пример (Hex)
START	Нет входных данных ⩾ 10 мс
Адрес Slave-устройства	01
Код команды	03
Число байт	10
Данные (старший байт) (T20)	00
Данные (младший байт) (T20)	01
Данные (старший байт) (T21)	00
Данные (младший байт) (T21)	02
Данные (старший байт) (T22)	00
Данные (младший байт) (T22)	03
Данные (старший байт) (T23)	00
Данные (младший байт) (T23)	04
Данные (старший байт) (T24)	00
Данные (младший байт) (T24)	05
Данные (старший байт) (T25)	00
Данные (младший байт) (T25)	06
Данные (старший байт) (T26)	00
Данные (младший байт) (T26)	07
Данные (старший байт) (T27)	00
Данные (младший байт) (T27)	08
CRC CHK (младший байт)	72
CRC CHK (старший байт)	98
END	Нет входных данных ⩾ 10 мс

4.3.3 CRC CHK (контрольная сумма)

Контрольная сумма CRC начиная от "адреса Slave-устройства" и до "последнего данного" Подсчет CRC:

Шаг 1: Задание 16-битного регистра (CRC регистр) = FFFFH.

Шаг 2: Запуск команды XOR для первого 8-битного сообщения (адрес) и младших 8 бит регистра CRC. Сохранение результата в регистре CRC.

Шаг 3: Правый сдвиг побитно регистра CRC и присвоение значения "0" старшему биту.

Шаг 4: Проверка младшего бита (бит 0) смещенного значения. Если он равен 0, повторение сдвига по шагу 3 для нового значения в регистре CRC; если бит 0 не равен 0, запуск XOR для A001H и смещенного значения, сохранение результата в регистре CRC.

Шаг 5: Повтор шагов 3 – 4 для всех 8 бит.

Шаг 6: Повтор шагов 2 – 5 для обработки всех сообщений. Окончательное значение в регистре CRC и есть контрольная сумма CRC.

Пример подсчета контрольной суммы CRC с помощью языка программирования C:

```
unsigned char* data      ←// index of the command message
unsigned char length     ←// length of the command message
unsigned int crc_chk(unsigned char* data, unsigned char length)
{
   int j;
   unsigned int reg_crc=0Xffff;
   while(length--)
   {
      reg_crc ^= *data++;
      for (j=0;j<8;j++)
      {
         If (reg_crc & 0x01) reg_crc=(reg_crc>>1) ^ 0Xa001; /* LSB(b0)=1 */
         else reg_crc=reg_crc >>1;
      }
   }
   return reg_crc;       // the value that sent back to the CRC register finally
}
```

Исключительный ответ:

ПЛК должен получать нормальный ответ от управляющего устройства. Но в ряде случаев ответ отсутствует, или ответ содержит ошибку (т.н. ответ исключения):

3. ПЛК не получил нормального ответа из-за ошибки коммуникации; в этом случае для управляющего устройства будет превышено допустимое время ожидания.
4. ПЛК принимает сообщение без ошибки связи, но не может его обработать, в этом случае управляющему устройству возвращается ответ исключения. В нем наиболее значимый бит имеет значение 1, а код ошибки объясняет условие возникновения исключительного ответа.

Пример ответа исключения с командным кодом 01H и кодом ошибки 02H:

Отправленное сообщение:

Имя области	Пример (Hex)
START	Нет входных данных ⩾ 10 мс
Адрес Slave-устройства	01
Код команды	01
Стартовый адрес	04
	00
Число точек	00
	10
CRC CHK (младший байт)	3C
CRC CHK (старший байт)	F6
END	Нет входных данных ⩾ 10 мс

Ответное сообщение:

Имя области	Пример (Hex)
START	Нет входных данных ⩾ 10 мс
Адрес Slave-устройства	01
Функция	81
Код исключительного ответа	02
CRC CHK (младший байт)	C1
CRC CHK (старший байт)	91
END	Нет входных данных ⩾ 10 мс

4.4 Адреса внутренних устройств ПЛК

Устрой-ство	Диапазон	Рабочий диапазон			Адрес MODBUS	Адрес
		ES2/EX2	SS2	SA2/SE SX2		
S	000~255	000~1023	000~1023		000001~000256	0000~00FF
S	256~511				000257~000512	0100~01FF
S	512~767				000513~000768	0200~02FF
S	768~1023				000769~001024	0300~03FF
X	000~377 (восьмер.)	000~377	000~377		101025~101280	0400~04FF
Y	000~377 (восьмер.)	000~377	000~377		001281~001536	0500~05FF
T	000~255 бит	000~255	000~255		001537~001792	0600~06FF
	000~255 слов	000~255	000~255		401537~401792	0600~06FF
M	000~255	0000	0000~4095		002049~003584	0800~08FF

Устройство	Диапазон	Рабочий диапазон			Адрес MODBUS	Адрес
		ES2/EX2	SS2	SA2/SE SX2		
M	256~511	~ 4095				0900~09FF
M	512~767					0A00~0AFF
M	768~1023					0B00~0BFF
M	1024~1279					0C00~0CFF
M	1280~1535					0D00~0DFF
M	1536~1791				045057~047616	B000~B0FF
M	1792~2047					B100~B1FF
M	2048~2303					B200~B2FF
M	2304~2559					B300~B3FF
M	2560~2815					B400~B4FF
M	2816~3071					B500~B5FF
M	3072~3327					B600~B6FF
M	3328~3583					B700~B7FF
M	3584~3839					B800~B8FF
M	3840~4095					B900~B9FF
C	000~199 (16-бит)	000~199	000~199		003585~003784	0E00~0EC7
		000~199	000~199		403585~403784	0E00~0EC7
	200~255 (32-бит)	200~255	200~255		003785~003840	0EC8~0EFF
		200~255	200~255		401793~401903 (действителен нечетный адрес)	0700~076F
D	000~255	0000 ~ 9999	0000 ~ 4999	0000 ~ 9999	404097~405376	1000~10FF
D	256~511					1100~11FF
D	512~767					1200~12FF
D	768~1023					1300~13FF
D	1024~1279					1400~14FF
D	1280~1535				405377~408192	1500~15FF
D	1536~1791					1600~16FF
D	1792~2047					1700~17FF
D	2048~2303					1800~18FF
D	2304~2559					1900~19FF
D	2560~2815					1A00~1AFF
D	2816~3071					1B00~1BFF
D	3072~3327					1C00~1CFF
D	3328~3583					1D00~1DFF
D	3584~3839					1E00~1EFF
D	3840~4095					1F00~1FFF
D	4096~4351				436865~440960	9000~90FF
D	4352~4999					9100~91FF
D	4608~4863					9200~92FF
D	4864~5119					9300~93FF
D	5120~5375		Не определено			9400~94FF
D	5376~5631					9500~95FF
D	5632~5887					9600~96FF
D	5888~6143					9700~97FF
D	6144~6399					9800~98FF
D	6400~6655					9900~99FF
D	6656~6911					9A00~9AFF
D	6912~7167					9B00~9BFF
D	7168~7423					9C00~9CFF
D	7424~7679					9D00~9DFF
D	7680~7935					9E00~9EFF
D	7936~8191					9F00~9FFF
D	8192~8447				440961~442768	A000~A0FF
D	8448~8703					A100~A1FF
D	8704~8959					A200~A2FF
D	8960~9215					A300~A3FF
D	9216~9471					A400~A4FF
D	9472~9727					A500~A5FF
D	9728~9983					A600~A6FF
D	9984~9999					A700~A70F
D	10000~11999	Применимо для DVP-SE			442767~444768	A710~AEDF

4.5 Код команды

4.5.1 Код команды: 01, Чтение состояния контактов (за исключением входов X)

Число точек (макс) = 255 (десятичн.) = FF (шестнадцатеричн.)

Пример: Чтение состояния контактов T20~T56 от Slave-устройства с ID#1
ПК→ПЛК ":01 01 06 14 00 25 BF CR LF"
Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	01
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	14
Число точек (старший байт)	00
Число точек (младший байт)	25
Контрольная сумма LRC	BF
ETX 1	0D (Hex)
ETX 0	0A (Hex)

Примем число точек в отправленном сообщении как **n** (десятичн.), величина **n**/8 - это **M** и остаток – **N.** Когда **N** = 0, число байт в ответном сообщении – **M**; если N≠0, число байт **M+1**.

ПЛК→ПК ":01 01 05 CD 6B B2 0E 1B D6 CR LF"
Ответное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	01
Число байт	05
Данные (Катушка T27…T20)	CD
Данные (Катушка T35…T38)	6B
Данные (Катушка T43…T36)	B2
Данные (Катушка T51…T44)	0E
Данные (Катушка T56…T52)	1B
Ошибка контрольной суммы (LRC)	E6
END 1	0D (Hex)
END 0	0A (Hex)

4.5.2 Код команды: 02, Чтение состояния контактов (включая входы X)

Пример: чтение состояния контактов Y024~Y070 от Slave-устройства с ID#01
ПК→ПЛК ": 01 02 05 14 00 25 BF CR LF"
Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	02
Начальный адрес (старший байт)	05
Начальный адрес (младший байт)	14
Число точек (старший байт)	00
Число точек (младший байт)	25
Контрольная сумма LRC	BF
END 1	0D (Hex)
END 0	0A (Hex)

Примем число точек в отправленном сообщении как **n** (десятичн.), величина **n**/8 - это **M** и остаток – **N.** Когда **N =** 0, число байт в ответном сообщении – **M**; если N≠0, число байт **M+1**.

ПЛК→ПК ": 01 01 05 CD 6B B2 0E 1B E5 CR LF"
Ответное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	02
Число байт	05
Данные (Катушка Y033…Y024)	CD
Данные (Катушка Y043…Y034)	6B
Данные (Катушка Y053…Y044)	B2
Данные (Катушка Y063…Y054)	0E
Данные (Катушка Y070…Y064)	1B
Ошибка контрольной суммы (LRC)	E5
END 1	0D (Hex)
END 0	0A (Hex)

4.5.3 Код команды: 03, Чтение содержимого регистров (T, C, D)

Пример: Чтение состояния T20~T27 от Slave-устройства с ID#01
ПК→ПЛК ": 01 03 06 14 00 08 DA CR LF"

Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	03
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	14
Число точек (старший байт)	00
Число точек (младший байт)	08
Контрольная сумма LRC	DA
END 1	0D (Hex)
END 0	0A (Hex)

ПЛК→ПК ":01 03 10 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 08 B8 CR LF"

Ответное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	03
Число байт	10
Данные (старший байт) (T20)	00
Данные (младший байт) (T20)	01
Данные (старший байт) (T21)	00
Данные (младший байт) (T21)	02
Данные (старший байт) (T22)	00
Данные (младший байт) (T22)	03
Данные (старший байт) (T23)	00
Данные (младший байт) (T23)	04
Данные (старший байт) (T24)	00
Данные (младший байт) (T24)	05
Данные (старший байт) (T25)	00
Данные (младший байт) (T25)	06
Данные (старший байт) (T26)	00
Данные (младший байт) (T26)	07
Данные (старший байт) (T27)	00
Данные (младший байт) (T27)	08
Контрольная сумма LRC	C8
END 1	0D (Hex)
END 0	0A (Hex)

4.5.4 Код команды: 05, Состояние ВКЛ/ВЫКЛ одного контакта

Данные состояния FF00 (шестнадцатеричн.) показывают ВКЛЮЧЕНИЕ контакта. Данные состояния 0000 (шестнадцатеричн.) показывают ВЫКЛЮЧЕНИЕ контакта. Когда MMNN = 0xFF00, катушка включается, MMNN = 0x0000 – выключается. Другие значения данных состояния некорректны.

Пример: ВКЛЮЧЕНИЕ катушки Y0

ПК→ПЛК ": 01 05 05 00 FF 00 F6 CR LF"

Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	05
Адрес катушки (старший байт)	05
Адрес катушки (младший байт)	00
Данные состояния (старший байт)	FF
Данные состояния (младший байт)	00
Ошибка контрольной суммы (LRC)	F6
END 1	0D (Hex)
END 0	0A (Hex)

ПЛК→ПК ": 01 05 05 00 FF 00 F6 CR LF"
Ответное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	05
Адрес катушки (старший байт)	05
Адрес катушки (младший байт)	00
Данные питания (старший байт)	FF
Данные питания (младший байт)	00
Ошибка контрольной суммы (LRC)	F6
END 1	0D (Hex)
END 0	0A (Hex)

4.5.5 Код команды: 06, Установка значения одного регистра

Пример: Установка значения регистра T0: 12 34 (шестнадцатеричн.)
ПК→ПЛК ": 01 06 06 00 12 34 AD CR LF"
Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	06
Адрес регистра (старший байт)	06
Адрес регистра (младший байт)	00
Значение регистра (старший байт)	12
Значение регистра (младший байт)	34
Ошибка контрольной суммы (LRC)	AD
END 1	0D (Hex)
END 0	0A (Hex)

ПЛК→ПК ": 01 06 06 00 12 34 AD CR LF"
Ответное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	06
Адрес регистра T0 (старший байт)	06
Адрес регистра T0 (младший байт)	00
Значение регистра (старший байт)	12
Значение регистра (младший байт)	34
Ошибка контрольной суммы (LRC)	AD
END 1	0D (Hex)
END 0	0A (Hex)

4.5.6 Код команды: 15, Состояние ВКЛ/ВЫКЛ нескольких контактов

Максимальное число контактов/катушек: 255
Пример: Установим состояние катушек Y007…Y000 = 1100 1101, Y011…Y010 = 01.
ПК→ПЛК ": 01 0F 05 00 00 0A 02 CD 01 11 CR LF"
Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	0F
Адрес катушки (старший байт)	05
Адрес катушки (младший байт)	00
Количе5ство катушек (старший байт)	00
Количе5ство катушек (младший байт)	0A
Число байт	02
Данные состояния (старший байт)	CD
Данные состояния (младший байт)	01
Ошибка контрольной суммы (LRC)	11
END 1	0D (Hex)
END 0	0A (Hex)

ПЛК→ПК ": 01 0F 05 00 00 0A E1 CR LF"

Ответное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	0F
Адрес регистра T0 (старший байт)	05
Адрес регистра T0 (младший байт)	00
Значение регистра (старший байт)	00
Значение регистра (младший байт)	0A
Ошибка контрольной суммы (LRC)	E1
END 1	0D (Hex)
END 0	0A (Hex)

4.5.7 Код команды: 16, Установка значений нескольких регистров

Пример: Установка значений регистров T0 - 00 0A , T1 - 01 02 .

ПК→ПЛК ": 01 10 06 00 00 02 04 00 0A 01 02 D6 CR LF"

Отправленное сообщение:

Имя области	ASCII
STX	:
Адрес Slave-устройства	01
Код команды	10
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	00
Число регистров (старший байт)	00
Число регистров (младший байт)	02
Число байт	04
Данные (старший байт)	00
Данные (младший байт)	0A
Данные (старший байт)	01
Данные (младший байт)	02
Ошибка контрольной суммы (LRC)	D6
END 1	0D(Hex)
END 0	0A(Hex)

ПЛК→ПК ": 01 10 06 00 00 02 E7 CR LF"

Ответное сообщение:

Имя области	ASCII
STX	3A
Адрес Slave-устройства	01
Код команды	10
Начальный адрес (старший байт)	06
Начальный адрес (младший байт)	00
Число регистров (старший байт)	00
Число регистров (младший байт)	02
Ошибка контрольной суммы (LRC)	E7
END 1	0D (Hex)
END 0	0A (Hex)

Последовательная функциональная диаграмма (SFC). Команды пошагового управления

В данной главе рассматриваются принципы пошагового (в режиме SFC) программирования.

5.1 Команды пошагового управления [STL], [RET]

Команда	Операнды	Функция	Шаги программы	Контроллеры			
				ES2/EX2	SS2	SA2	SX2
STL	S0~S1023	Запуск пошагового управления	1				

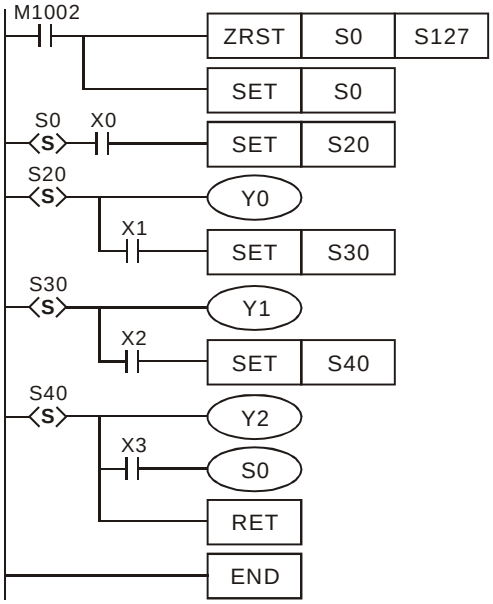
Описание:
STL оперирует шаговыми операндами S**n**. Когда команда STL запускается в программе, основная программа перейдет в пошаговое (лестничная диаграмма) управление. S0 ~ S9 применяются командой STL в качестве начальных шаговых операндов (точек). Номера шаговых операндов (точек) не могут повторяться.

Команда	Операнды	Функция	Шаги программы	Контроллеры			
				ES2/EX2	SS2	SA2	SX2
RET	Нет	Завершение пошагового управления	1				

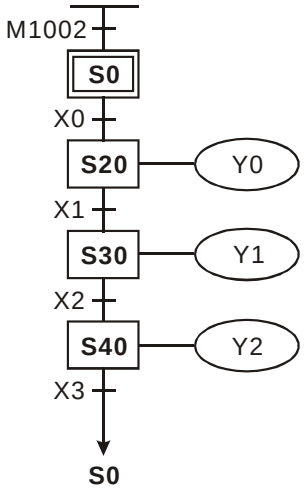
Описание:
Команда RET указывает завершение программы пошагового управления, начинающейся с S0 ~ S9, т.е. после выполнения RET продолжается выполнение основной программы. Могут применяться максимум 10 начальных шаговых операндов (S0 ~ S9) и для каждого необходимо применение команды RET для завершения STL программы. С помощью пошаговых команд (лестничной диаграммы) STL/RET, SFC пошагово реализует процесс управления.

Пример программы:

Лестничная диаграмма:



SFC:

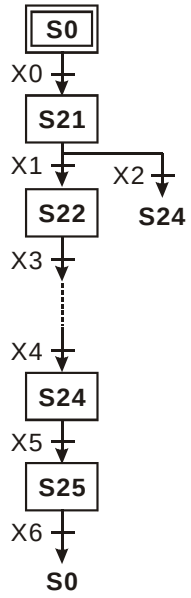


5.2 Последовательная функциональная диаграмма (SFC)

Автоматическое управление представляет собой сочетание процессов электронного и механического управления. Последовательность процесса автоматического управления можно

разделить на несколько шагов. Каждый шаг представляет собой набор действий и для завершения шага необходимо выполнение определенных условий. Выполнение этих условий служит непременным требованием для перехода к следующему шагу (шаговый переход), при этом действия предыдущего шага сбрасываются. Эта последовательность определяет принцип разработки последовательных функциональных диаграмм (SFC).

Особенности:

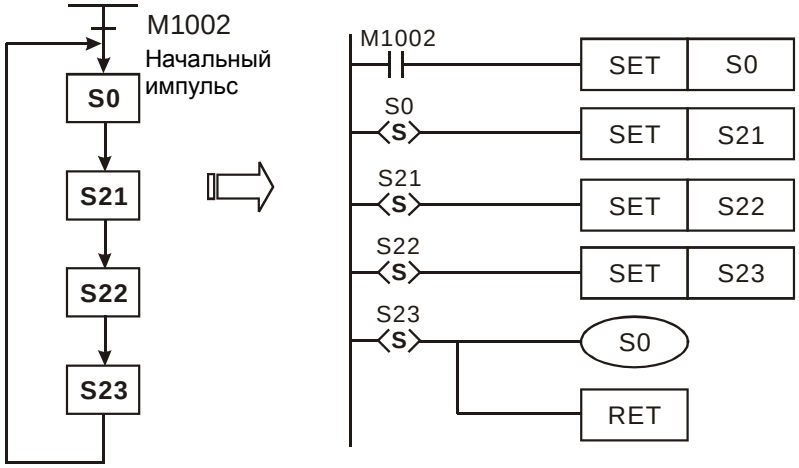
1. Простая последовательная взаимосвязь между выходами не является общей логикой STL. В схеме шаговой функции может присутствовать функционирование параллельно нескольких выходов и осуществляться различные процессы блокировки. 2. SFC функционирует как блок-схема. STL-операция работает с внутренними шаговыми реле S, которые также определяют каждое состояние SFC. Когда текущий шаг выполнен, происходит переход к следующему шагу, формируя тем самым непрерывный процесс управления. 3. Цикл процесса показан на диаграмме SFC. С начального шага S0 осуществляется переход к основному шагу S21 путем выполнения условия X0. С S21 возможен переход на S22 или S24 в зависимости от условий X1 и X2. Процесс заканчивается на шаге S25 и весь процесс заканчивается, когда с S25 осуществляется возврат к S0 путем выполнения условия X6.	SFC: 

Описание символов SFC в ПО Ladder Editor (WPLSoft)	
	Режим лестничной (релейно-контактной) диаграммы. Символ помещается перед лестничной диаграммой STL , обозначая запуск STL-программы.
	Инициализация шагов (стартовые шаговые операнды) в SFC. Доступны S0 ~ S9.
	Шаг общего назначения. Доступны S10 ~ S1023.
	Шаговый переход. Применяется для перехода в заданную точку (не соседнюю). (Переход вверх/вниз на не соседний уровень, возврат к начальному шагу, переход на другой уровень и т.п.)
	Условие перехода между отдельными шагами программы.
	Запуск альтернативного выбора (расхождения). Осуществляет переход на ту или иную ветку, в зависимости от условий перехода.
	Конец альтернативного выбора. Осуществляет процесс схождения на основную программу после окончания процесса альтернативного выбора.
	Параллельное расхождение. Запускает процесс одновременного выполнения параллельных процессов.
	Схождение после параллельного расхождения на основную программу.

5.3 Работа STL программы.

Лестничная диаграмма (STL) – метод программирования для написания программ, работающих аналогично SFC. STL позволяет программировать ПЛК простым и понятным методом блок-схемы. Последовательность шагов SFC легко представляется с помощью лестничной диаграммы (см. ниже).

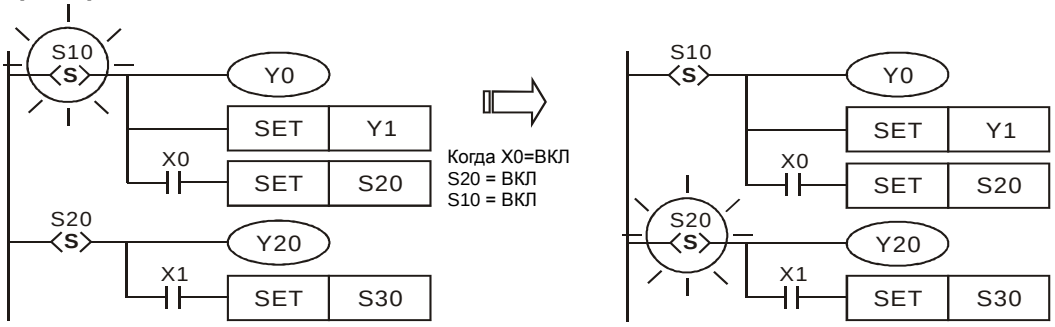
STL программа стартует командой STL и заканчивается командой RET. Операнды S**n** в STL являются шаговыми. Когда команда STL появляется в программе, начинается выполнение пошаговых операций согласно лестничной диаграмме. Команда RET показывает завершение пошаговой программы, стартовавшей с S0 ~ S9 и данная команда применяется на каждом этапе завершения STL программы. Отсутствие команды RET в конце каждой пошаговой операции приведет к ошибке, что отобразит WPLSoft.



Выполнение шага STL программы:

STL программа состоит из многих шагов. Для осуществление каждого шага необходимо выполнить следующие 3 действия: Установка состояния выхода, назначение условия перехода, определение номера следующего управляемого шага

Пример:



Описание:

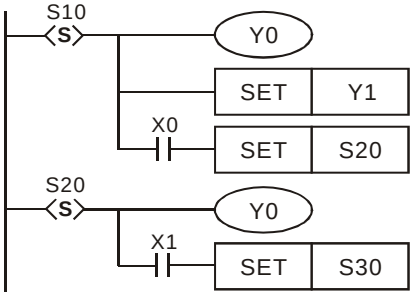
Когда S10 включен, Y0 и Y1 тоже включаются. Когда X0 включен, S20 и Y20 тоже включаются. Когда S10 выключен, Y0 также выключен, но Y1 останется включенным (команда SET применится для Y1, поэтому Y1 будет включен и зафиксирован).

STL переход:

Когда состояние шага S**n** =1, его схема будет активирована. При S**n** = 0, его схема будет неактивна. Интервал между двумя этими состояниями равен времени одного скана.

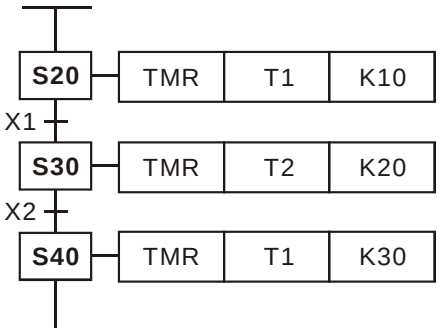
Повторное использование выходов:

1. Одни и те же выходы могут применяться на разных шагах STL программы.
2. См. диаграмму. Выход (Y0) применяется в процессе 2-х шагов. Y0 остается включенным когда S10 переходит к S20.
3. Y0 отключится при переходе от S10 к S20. Когда S20 активен, Y0 включается. В этом случае, Y0 остается включенным при переходе от S10 к S20.
4. Для общих лестничных диаграмм таких повторений использования выходов следует избегать.



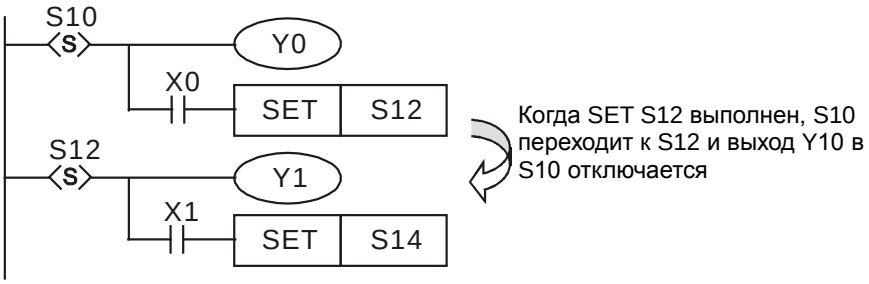
Повторное использование таймера:

См. диаграмму. Один и тот же таймер не может применяться в 2-х один за другим идущих шагах.



Переход к следующему шагу:

Команды SET Sn и OUT Sn применяются для перехода к следующему шагу. Это может быть последовательный переход к следующему шагу (в т.ч., начиная со стартовых шагов S0…S9) или скачкообразный переход (jump), возврат и т.д. Использование команд SET Sn и OUT Sn может быть различно при различных условиях.



SET Sn

Применяется для перехода к следующему шагу в одном процессе. После перехода происходит ее обнуление.

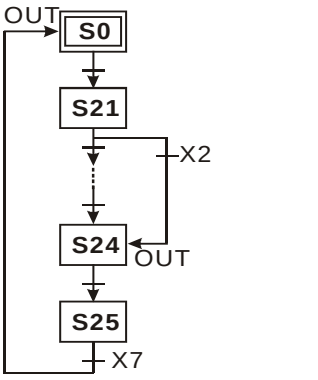
OUT Sn

Применение 1: возврат к начальному шагу той же последовательности, 2: скачкообразный переход (jump) вверх/вниз в той же последовательности (несмежные шаги) 3: переход в другой процесс. После перехода происходит обнуление команды.

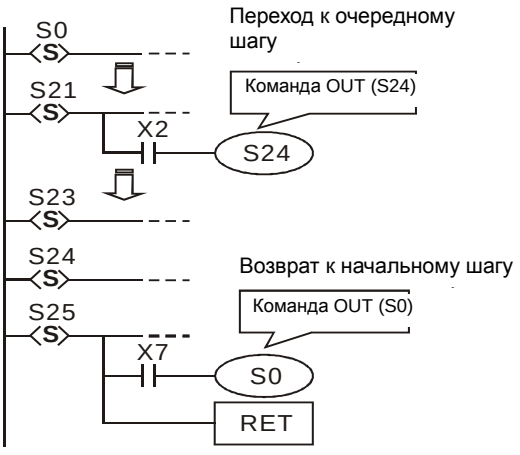
① Возврат к начальному шагу той же последовательности.

②Скачкообразный переход (jump) вверх/вниз в той же последовательности (несмежные шаги).

SFC диаграмма:

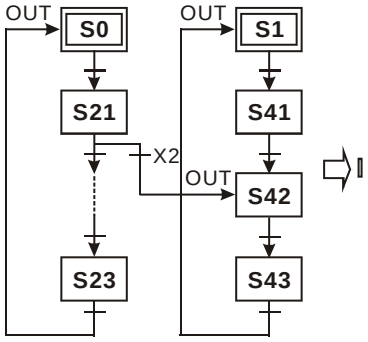


Лестничная диаграмма:

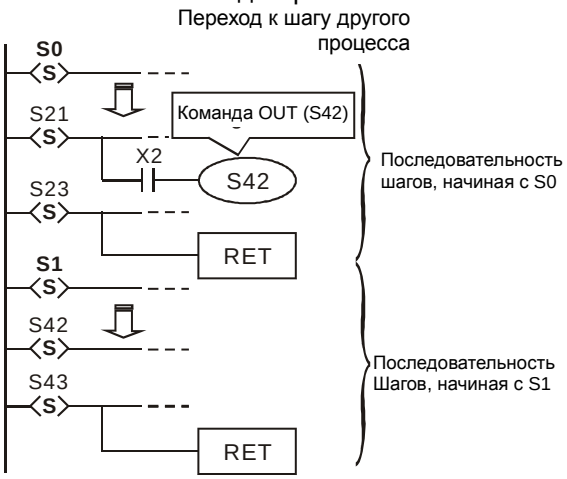


③ переход в другой процесс.

SFC диаграмма:

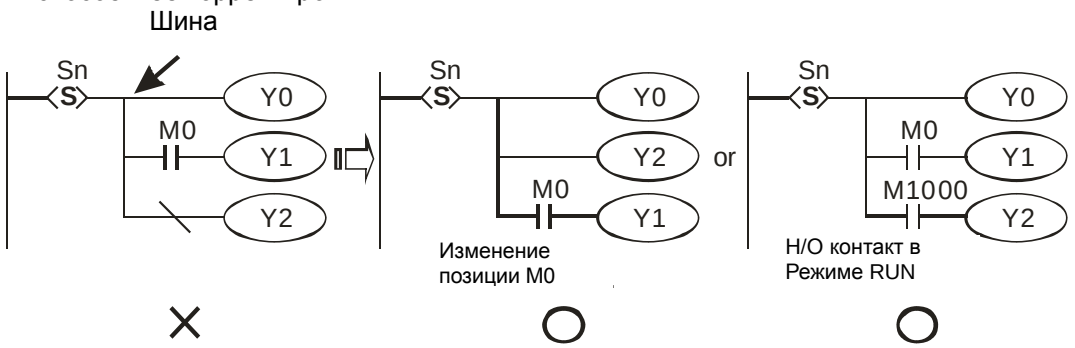


Лестничная диаграмма:



Две разных шаговых последовательности: S0 и S1:
Команда OUT применяется для возврата с шага S23 к начальному шагу S0
Команда OUT применяется для возврата с шага S43 к начальному шагу S1

Корректное соединение выходов:
После команд LD или LDI во второй линии после точки шага S**n**, выходные реле не могут быть подключены к шине непосредственно. На диаграммах показана некорректная схема слева и способы ее корректировки.



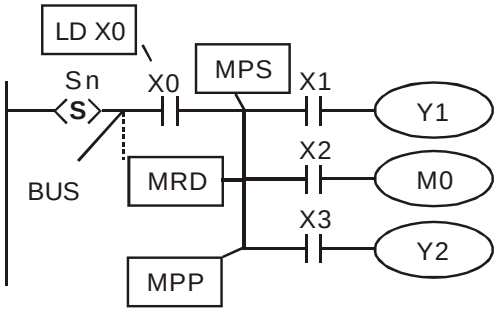
Ограничения применения команд:
Последовательные/параллельные схемы или команды общих лестничных диаграмм также применимы в пошаговых STL диаграммах. Но существует ряд ограничений по их применению. См. таблицу.

Основные команды, применяемые в пошаговом режиме

Основная команда / Шаг		LD/LDI/LDP/LDF AND/ANI/ANDP/ANDF OR/ORI/ORP/ORF INV/OUT/SET/RST	ANB/ORB MPS/MRD/MPP	MC/MCR
Начальный шаг/ Общий шаг		Да	Да	Нет
Разветвление программы	Общие выходы	Да	Да	Нет
	Переход к следующему шагу	Да	Да	Нет

1. Команды MC/MCR в пошаговом режиме не применяются.
2. Команды STL не применяются в подпрограммах и подпрограммах прерывания.
3. Команда CJ может применяться в пошаговом режиме, но ее применение не рекомендуется.
4. Команды позиционирования MPS/MRD/MPP:

Лестничная диаграмма:



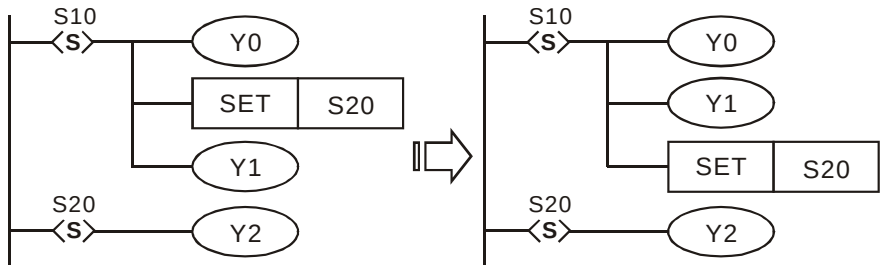
Командный код:

```
STL    Sn
LD     X0
MPS
AND    X1
OUT    Y1
MRD
AND    X2
OUT    M0
MPP
AND    X3
OUT    Y2
```

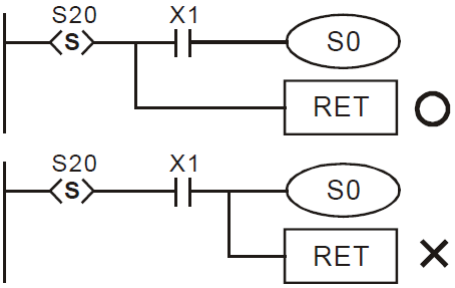
Описание:
Команды MPS/MRD/MPP применимы только после команд LD или LDI.

Заметки:

1. Команды шагового перехода (SET S□ or OUT S□) рекомендуется ставить в конце шага после обработки всех выходов и процессов. Работа ПЛК при этом не изменится. Если в S10 применяется много процессов, рекомендуется применять схему справа (см. диаграммы ниже).



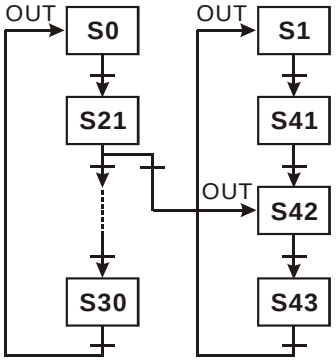
2. Примеры корректного (перед H/O или H/3 контактом)и некорректного применения ко-команды RET.



5.4 Особенности программирования пошагового управления.

1. Каждый шаг SFC должен начинаться с начальных шаговых операндов S0 ~ S9 и завершаться командой RET.
2. Если команда STL не используется, шаговые реле S могут применяться в качестве реле общего назначения.
3. При выполнении STL номер операнда S не должен повторяться.
4. Типы последовательностей:
   Одиночная последовательность: простая последовательность без альтернативных или параллельных ветвей.
   Сложная одиночная последовательность: одиночная последовательность с наличием разветвлений альтернативных или параллельных ветвей.
   Множественная последовательность: выполнение нескольких последовательностей, максимум до 10: S0 ~ S9.
5. Последовательный скачкообразный переход (jump): См. схему с несколькими последовательностями ниже:

   - Две последовательности, S0 и S1. ПЛК сначала обрабатывает S0 ~ S30 и затем S1 ~ S43.
   - Пользователь может назначить шаг перехода в другую последовательность.
   - При выполнении условия ниже S21 то будет осуществлен переход в последовательность S1 на шаг S42 посредством jump-перехода.



6. Ограничения на расходящиеся последовательности:
   a) Максимально 8 разветвлений из одного шагового операнда.
   b) Максимально 16 всех альтернативных или параллельных разветвлений.
   c) Из одной точки последовательности можно совершить переход только в одну точку другой последовательности.
7. Сброс шаговых операндов и отключение выходов
   a) Для сброса шаговых операндов применяется команда ZRST.
   b) Установка флага M1034 = 1 отключает выходы Y.
8. Энергонезависимый пошаговый операнд:
   Состояние (ВКЛ/ВЫКЛ) энергонезависимого шагового операнда будет сохранятся при отключении питания ПЛК. При последующем включении выполнение программы

начнется с точки (шага) прерывания. (См. область энергонезависимых пошаговых операндов).

9. Специальные реле и регистры (См. раздел 5.6):

Устройство	Описание
M1040	Запрещение шаговых переходов.
M1041	Старт шагового перехода. Флаг команды IST.
M1042	Импульсный старт шагового перехода. Флаг команды IST.
M1043	Возвращение в нулевую точку завершено. Флаг команды IST.
M1044	Нахождение в нулевой точке. Флаг команды IST.
M1045	Запрещение сброса всех выходов. Флаг команды IST.
M1046	Выполнение STL-режима. M1046 = ON при начавшем выполнение любом шаге
M1047	Включение мониторинга работы STL-режима
D1040	№ 1-го активного шага
D1041	№ 2-го активного шага
D1042	№ 3-го активного шага
D1043	№ 4-го активного шага
D1044	№ 5-го активного шага
D1045	№ 6-го активного шага
D1046	№ 7-го активного шага
D1047	№ 8-го активного шага

5.5 Типы последовательностей

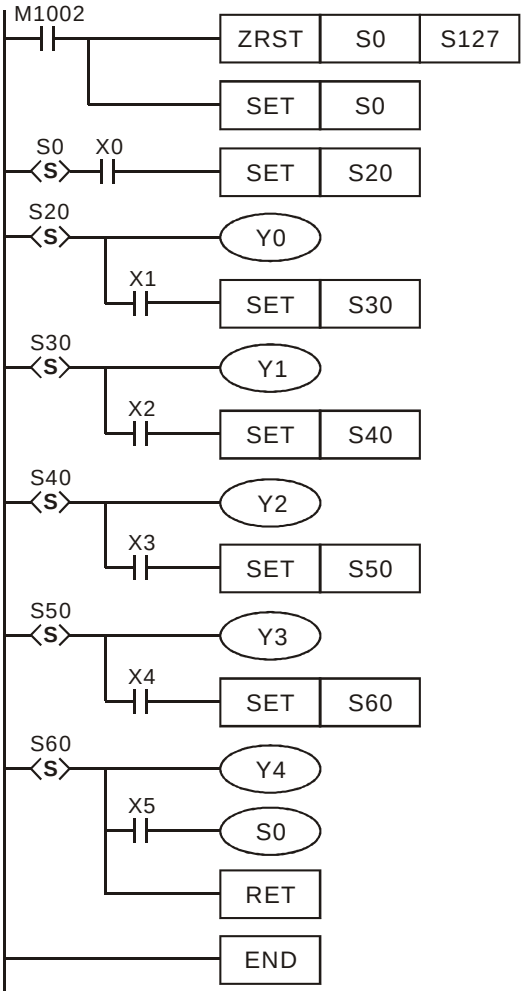
Одиночная последовательность: основной тип.

Первый шаг в лестничной диаграмме, т.н. шаг инициализации, имеет диапазон операндов: S0 ~ S9. Следующие шаги нумеруются как S10 ~ S1023. Когда применяется команда IST, S10 ~ S19 становятся шагами сброса.

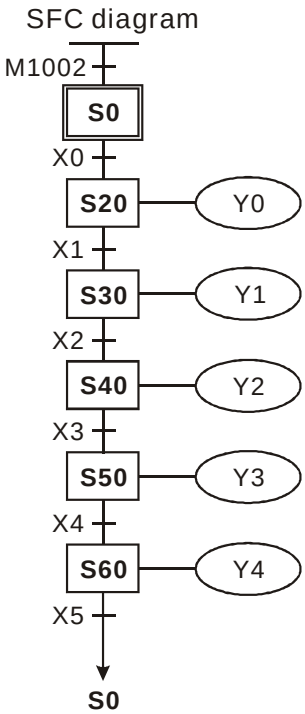
1. Одиночная последовательность без разветвлений

   Шаги обрабатываются последовательно. После отработки всей последовательности происходит возврат к шагу инициализации (начальному шагу).

Лестничная диаграмма:

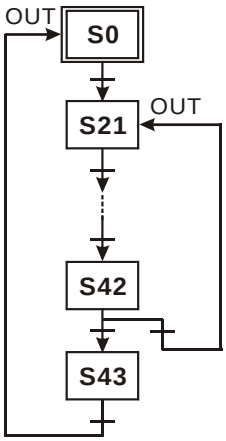


SFC диаграмма:

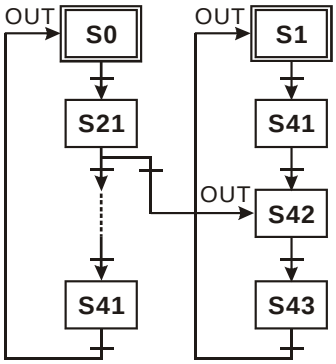


2. Процесс со скачкообразным переходом (jump)

a) В ходе выполнения последовательности осуществляется переход на шаг, находящийся выше по лестничной диаграмме.

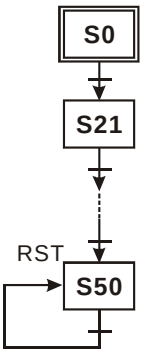


b) Переход осуществляется на шаг, находящийся в другой последовательности.



3. Процесс со сбросом

S50 осуществит сброс себя при условии выполнения процесса и окончания последовательности на данном шаге.

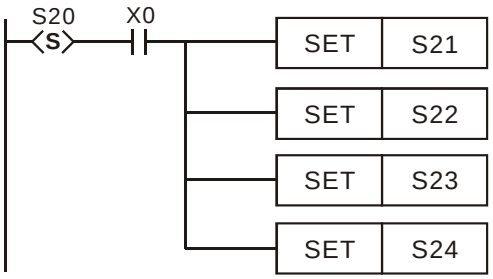


Сложная одиночная последовательность: одиночная последовательность с наличием разветвлений альтернативных или параллельных ветвей.

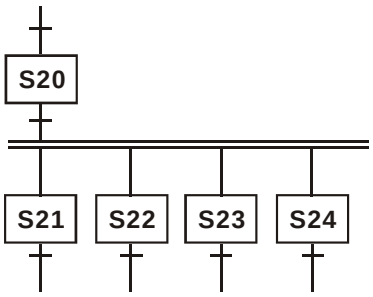
1. Структура с параллельным разветвлением

Шаг может быть передан на несколько ветвей одновременно, после чего происходит одновременная обработка нескольких процессов. Например, когда X0 включен, состояние S20 будет одновременно передано S21, S22, S23 и S24 (одновременное выполнение 4-х процессов).

Лестничная диаграмма:

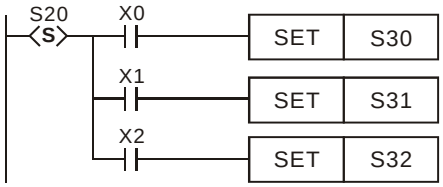


SFC диаграмма:

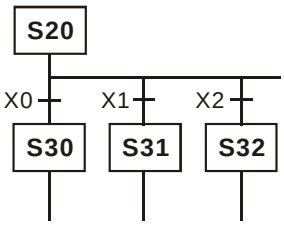


2. Селективное (альтернативное) разветвление
При выполнении заданных условий в текущем шаге происходит выбор продолжения процесса среди нескольких (передача шага только на одну ветвь из нескольких). Например, когда X0 включен, S20 переходит на S30; когда X1 включен, S20 переходит на S31; когда X2 включен, S20 переходит на S32.

Лестничная диаграмма:

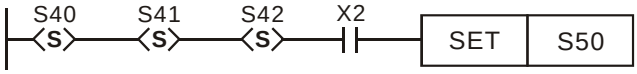


SFC диаграмма:

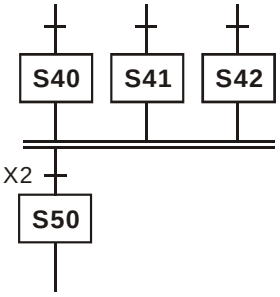


3. Схождение параллельного разветвления
Команда STL формирует структуру параллельного схождения различных процессов к одному шагу. Переход возможен только после полного завершения всех параллельных процессов.

Лестничная диаграмма:

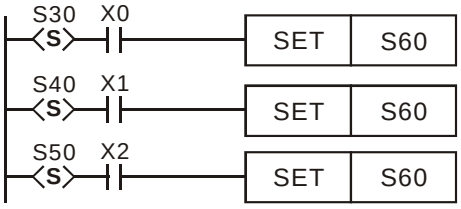


SFC диаграмма:

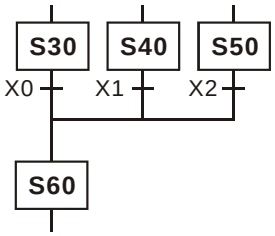


4. Селективное (альтернативное) схождение
Выполнение программы перейдет в S60, когда будет активен один из процессов S30, S40 или S50.

Лестничная диаграмма:

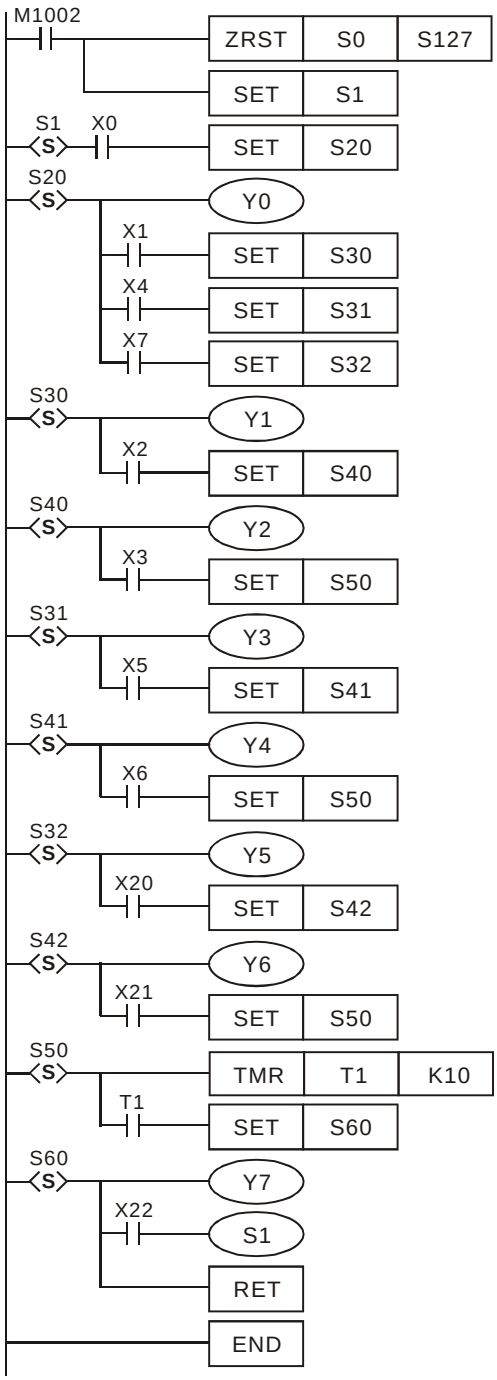


SFC диаграмма:

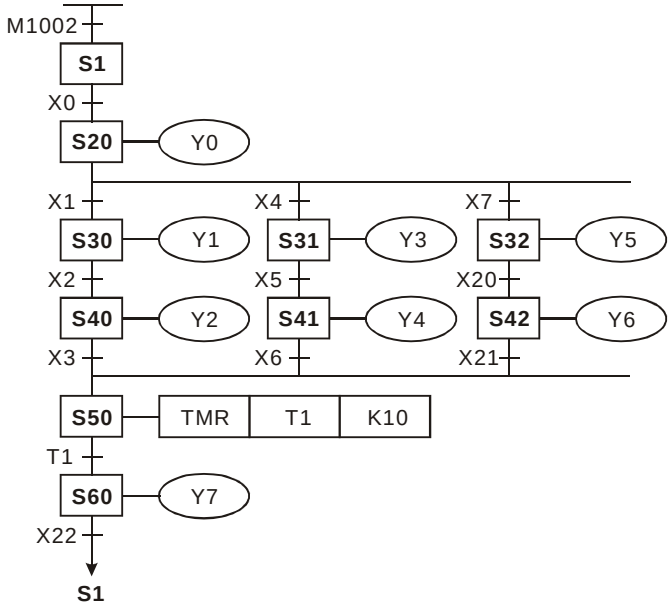


Пример селективного (альтернативного) разветвления и схождения:

Лестничная диаграмма:

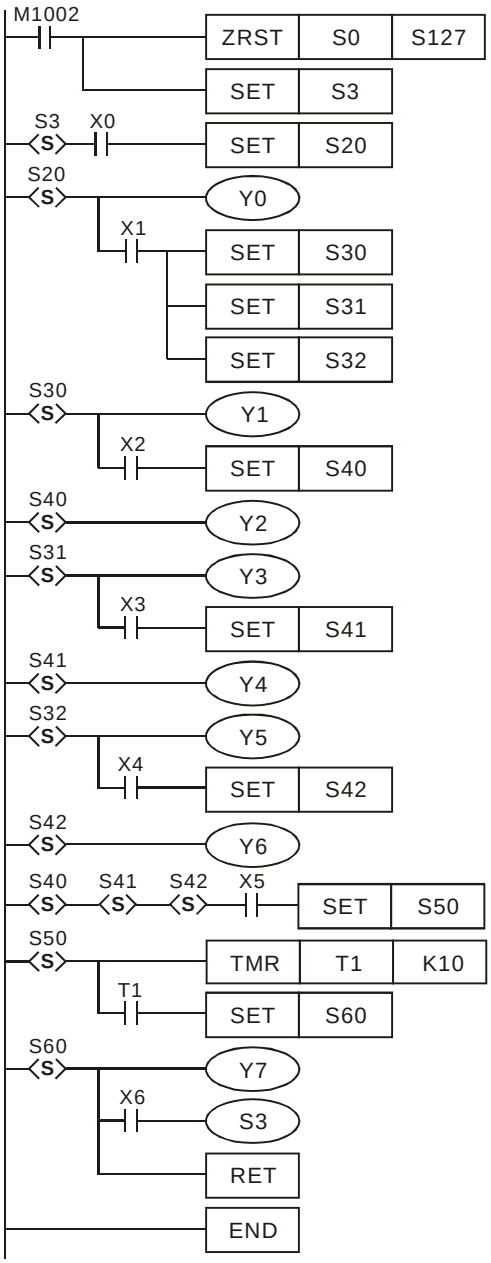


SFC диаграмма:

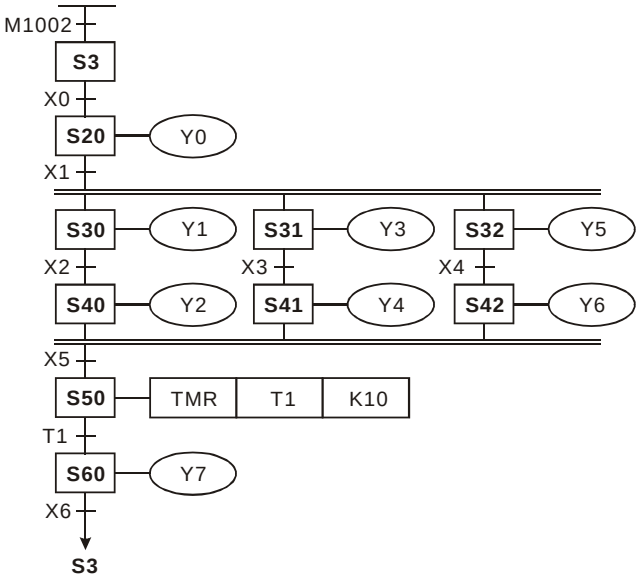


Пример параллельного разветвления и схождения:

Лестничная диаграмма:

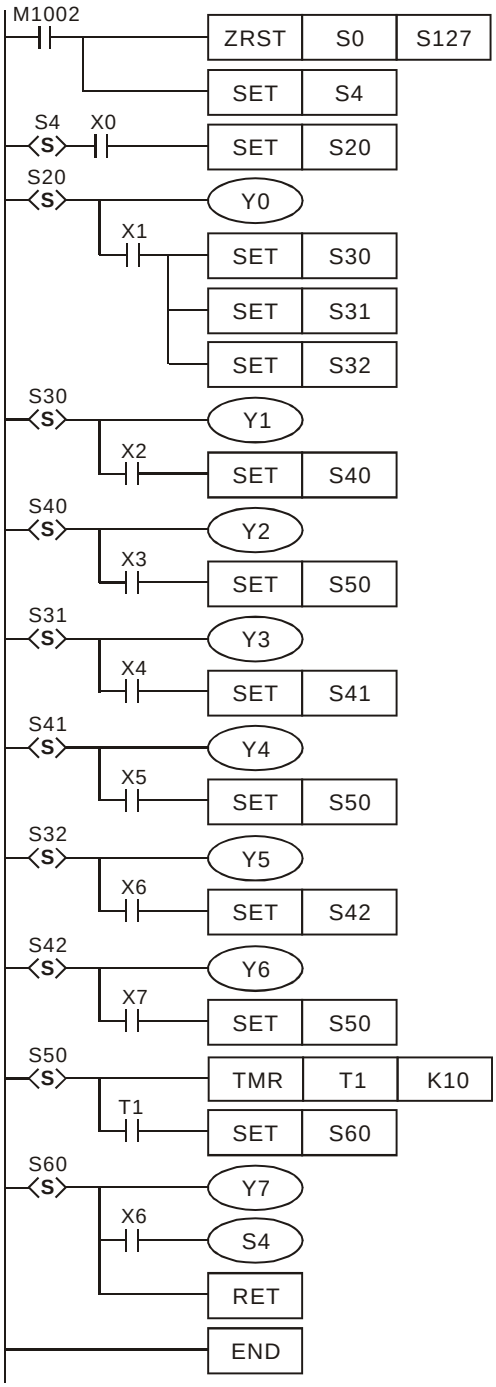


SFC диаграмма:

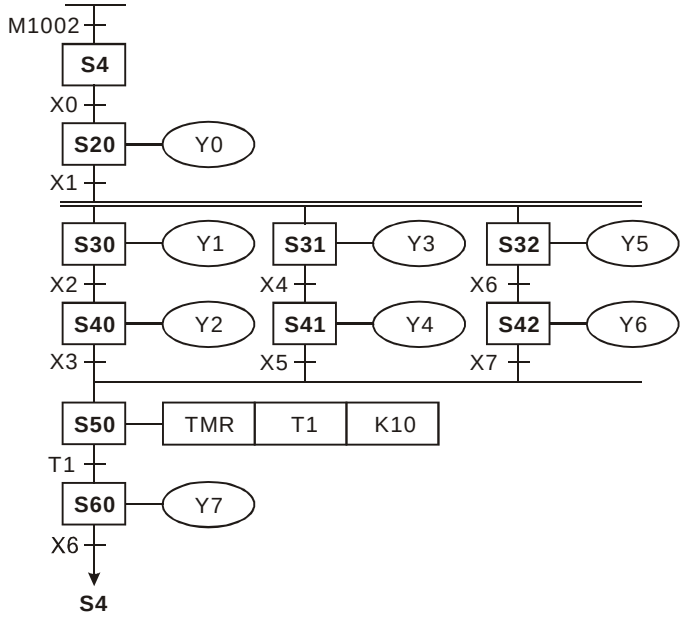


Пример параллельного разветвления и селективного (альтернативного) схождения:

Лестничная диаграмма:

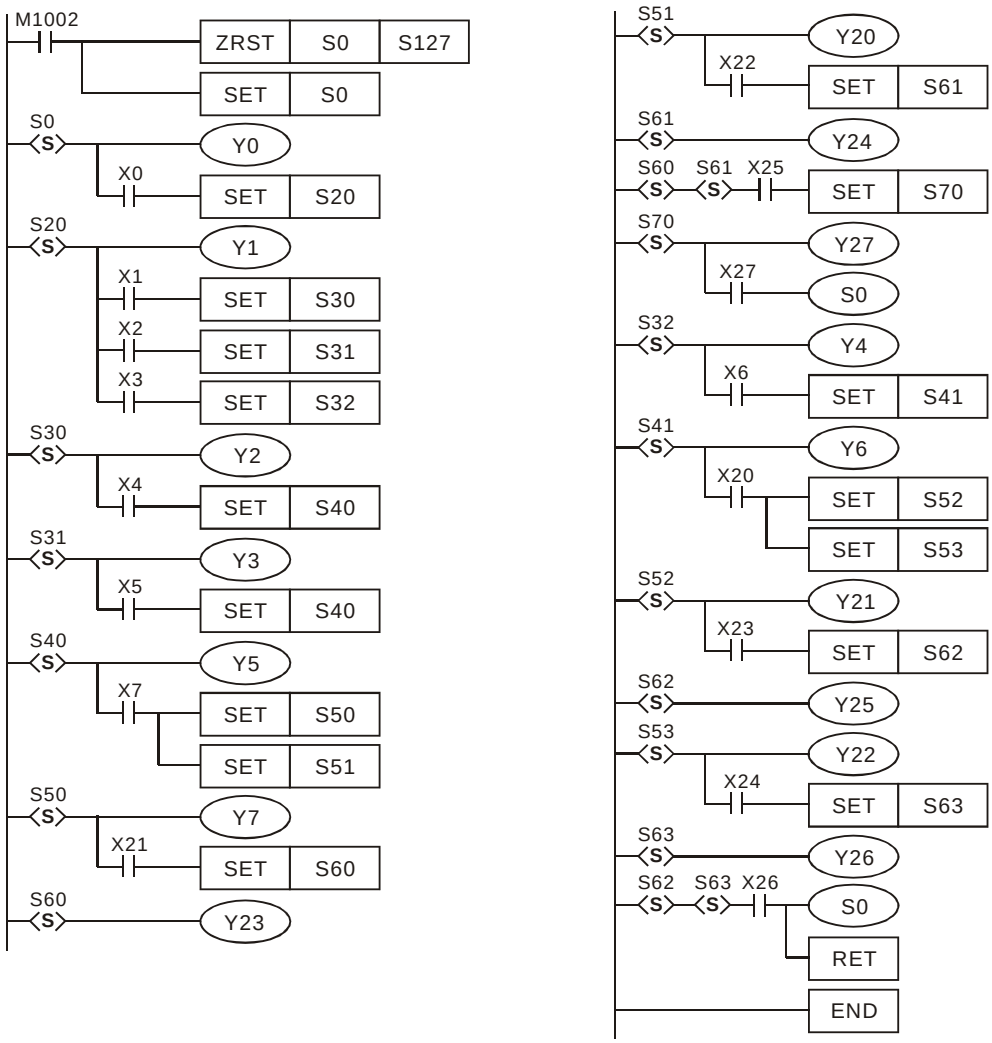


SFC диаграмма:

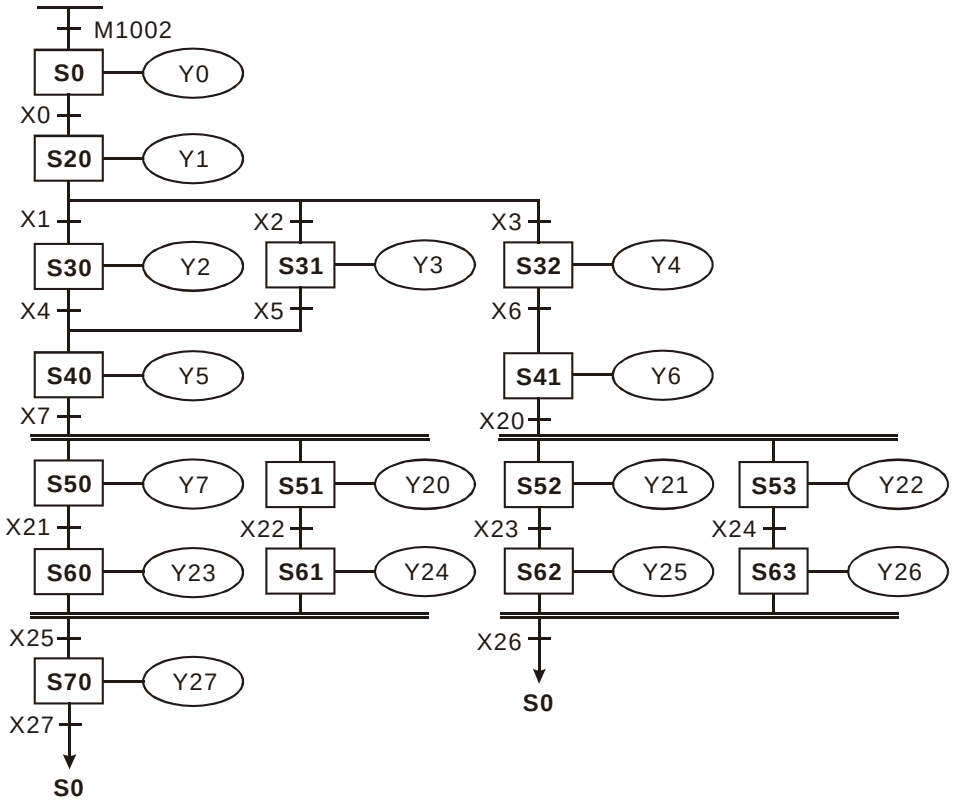


Пример комбинации 1: (Селективное разветвление/схождение и параллельное разветвление/схождение)

Лестничная диаграмма:

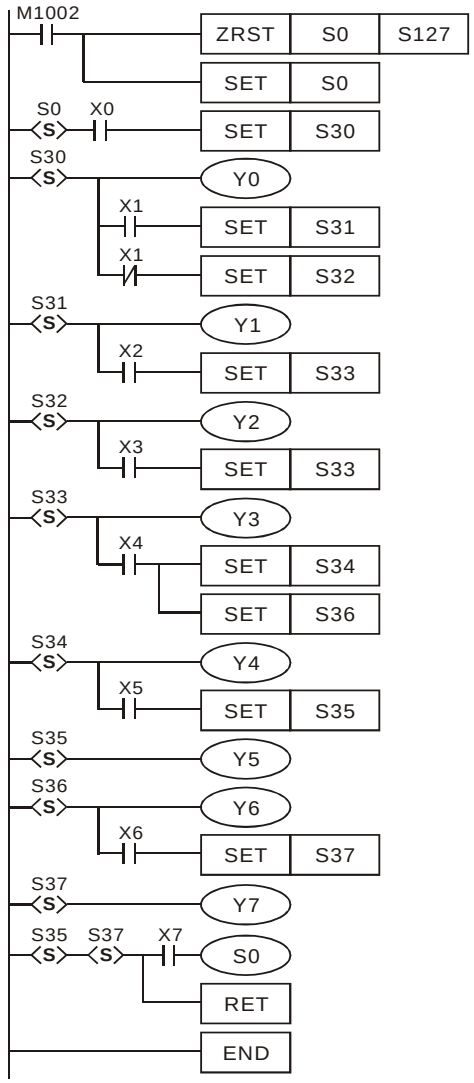


SFC диаграмма:

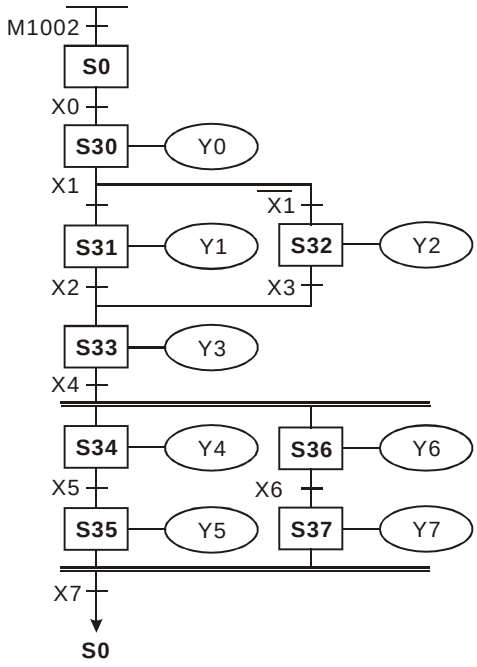


Пример комбинации 2: (Селективное разветвление/схождение и параллельное разветвление/схождение)

Лестничная диаграмма:



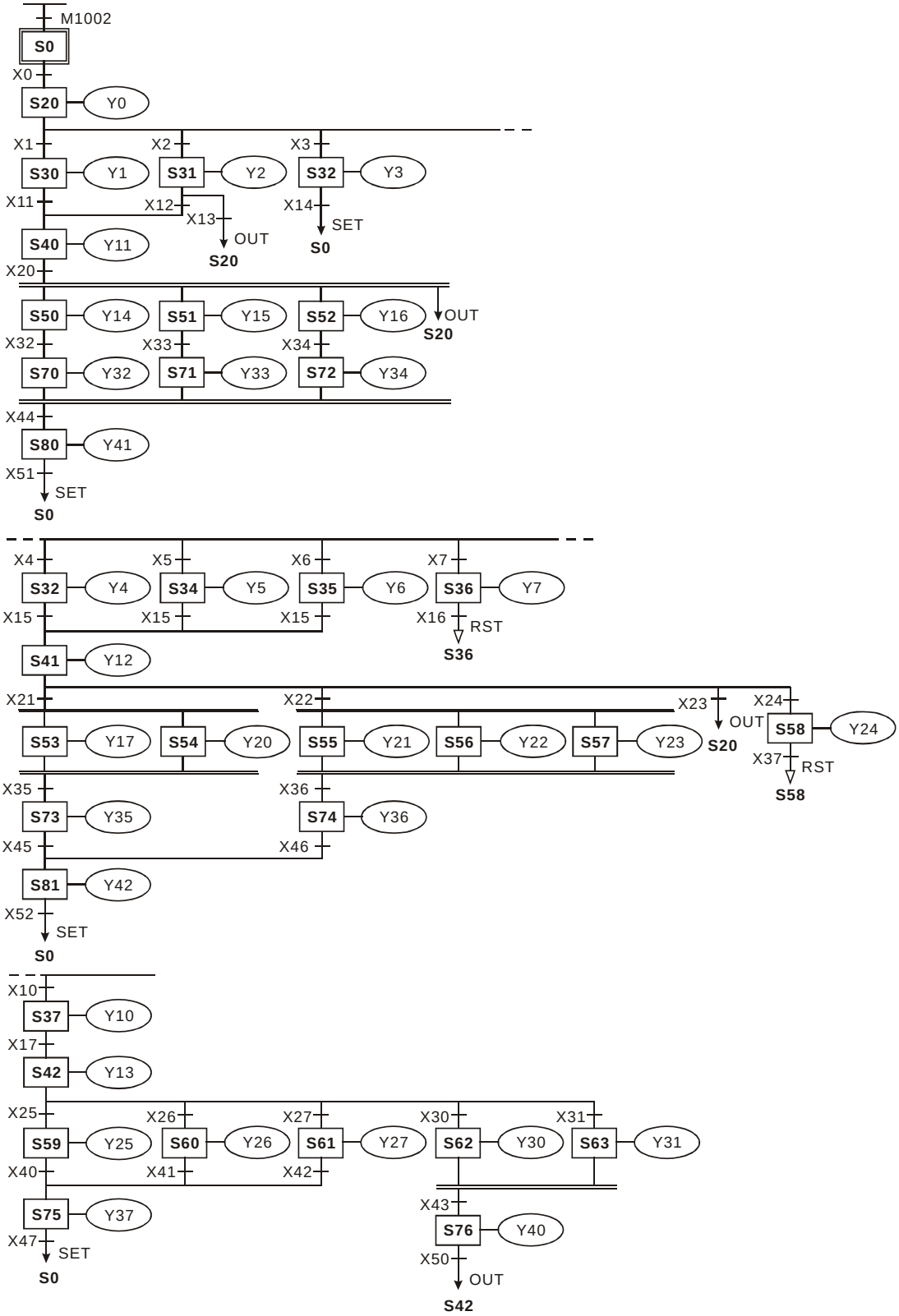
SFC диаграмма:



Ограничения на разветвление:

1. Из одной шаговой точки можно реализовать максимально 8 разветвлений на другие последовательности. На диаграмме ниже показаны 8 разветвлений на шаги S30 ~ S37 после шага S20.
2. Можно применять сходимость в одну шаговую точку максимально с 16 других последовательностей. На диаграмме ниже показаны 4 разветвления от S40, 7 разветвлений с S41, и 5 рразветвлений с S42. Т.е. максимум 16 циклов в последовательности.
3. Пользователь может назначить любой шаг в последовательности для осуществления перехода.

SFC диаграмма:



5.6 Команда IST

Описание команды IST см. в Главе 3 на стр. 184

Поиск и устранение неисправностей

В данной главе содержатся таблицы кодов ошибок и информация по устранению неполадок во время работы ПЛК.

6.1 Общие проблемы и решения

Проблема	Рекомендуемые действия по решению проблемы
Не горят все световые индикаторы	1. Проверьте подключение питания. 2. Проверьте соответствие параметров питания характеристикам ПЛК. 3. Убедитесь в исправности источника питания. 4. Отсоедините питание от других устройств, если питание подается от блока, питающего ПЛК. Если проблема решена, значит замените блок питания на более мощный. 5. Если вышеперечисленные мероприятия не решили проблему, обратитесь к дилеру или дистрибьютору, где был приобретен ПЛК.
Мигает световой индикатор ОШИБКА	1. Причиной может быть неправильная команда, ошибка связи, недопустимая операция или отсутствие команды, ошибка выдается функцией самодиагностики, код ошибки и шаг, где она произошла сохраняются в специальных регистрах. Коды могут быть считаны при помощи ПО WPLSoft. Регистры кодов: Код ошибки: D1004. Шаг, где возникла ошибка: D1137 2. Если нет связи между ПЛК и индикатор мигает быстро, это означает падение мощности от блока питания постоянного тока 24В, проверьте нагрузку на блок питания. 3. Индикатор непрерывно горит, если время выполнения программы превышает заданное время (устанавливается в D1000). Проверьте программу или сторожевой таймер. Если это не помогло, загрузите программу снова и перезапустите ПЛК. Если данная операция также не привела к отключению индикатора ошибки, проверьте наличие посторонних предметов в ПЛК.
Диагностика неисправностей на входах ПЛК	**Световой индикатор входов НЕ ГОРИТ:** 1. Проверьте подключение устройства ввода. 2. Убедитесь в правильной подаче питания на входные клеммы. 3. Если вышеперечисленные мероприятия не решили проблему, обратитесь к дилеру или дистрибьютору, где был приобретен ПЛК. 4. Если питание неправильно подается на входы, проблема, скорее всего, в устройстве ввода или источнике питания. Проверьте их. **Световой индикатор входов ГОРИТ:** 1. Проверьте входные настройки в программе. Если контроль входов отключен, проблема, вероятно, во входной цепи ПЛК, обратитесь к дилеру или дистрибьютору, где был приобретен ПЛК. 2. Если контроль выходов включен, снова проверьте программу. Кроме того проверьте утечку тока на входах, а также проверьте не дублируются ли применение оборудования или выполнение программы при использовании команд управления MC и CJ. 3. Проверьте настройки распределения входов/выходов.

Проблема	Рекомендуемые действия по решению проблемы
Диагностика неисправностей на выходах ПЛК	**Световой индикатор входов ГОРИТ:** 1. Проверьте подключение нагрузки. 2. Убедитесь в правильной подаче питания на нагрузку. 3. Если питание неправильно подается на входы, проблема, скорее всего, в нагрузке. Проверьте устройства нагрузки. 4. Если питание не подается на нагрузку, проблема, скорее всего, в выходных цепях ПЛК, обратитесь к дилеру или дистрибьютору, где был приобретен ПЛК. **Световой индикатор входов НЕ ГОРИТ:** 1. Проверьте выходные настройки в программе. Если контроль выходов включен, вероятно, происходит дублирование выходов. 2. Принудительно активируйте выходы. Если индикатор горит, проверьте входы, если не горит, проблема, вероятно, в выходных цепях ПЛК, обратитесь к дилеру или дистрибьютору, где был приобретен ПЛК.

6.2 Таблица кодов ошибок (шестнадцатеричный формат)

После загрузки программы в ПЛК недопустимое использование операндов или неправильный синтаксис приведут к миганию индикатора ОШИБКА, а флаг M1004 будет включен. В этом случае можно определить причину ошибки, проверив код ошибки в специальном регистре D1004 (в шестнадцатеричном формате). Адрес, где произошла ошибка, будет сохранен в регистре D1137. Если возникла общая ошибка цикла, адрес в регистре D1137 будет недействителен.

Код ошибки	Описание	Действия
0001	Операнд битового устройства S выходит за границы диапазона	Проверьте D1137 (адрес ошибки) Перезапустите команду корректно
0002	Метка P выходит за границы диапазона или дублируется	
0003	Операнд KnSm выходит за границы диапазона	
0102	Указатель прерывания I выходит за границы диапазона или дублируется	
0202	Команда MC выходит за границы диапазона	
0302	Команда MCR выходит за границы диапазона	
0401	Операнд битового устройства X выходит за границы диапазона	
0403	Операнд KnXm выходит за границы диапазона	
0501	Операнд битового устройства Y выходит за границы диапазона	
0503	Операнд KnYm выходит за границы диапазона	
0601	Операнд битового устройства T выходит за границы диапазона	
0604	Операнд словного устройства T выходит за границы диапазона	
0801	Операнд битового устройства M выходит за границы диапазона	
0803	Операнд KnMm выходит за границы диапазона	
0B01	Операнд K, H выходит за границы диапазона	
0D01	Неправильный операнд в команде DECO	
0D02	Неправильный операнд в команде ENCO	
0D03	Неправильный операнд в команде DHSCS	
0D04	Неправильный операнд в команде DHSCR	
0D05	Неправильный операнд в команде PLSY	
0D06	Неправильный операнд в команде PWM	
0D07	Неправильный операнд в команде FROM/TO	
0D08	Неправильный операнд в команде PID	
0D09	Неправильный операнд в команде SPD	
0D0A	Неправильный операнд в команде DHSZ	
0D0B	Неправильный операнд в команде IST	
0E01	Операнд битового устройства C выходит за границы диапазона	
0E04	Операнд словного устройства C выходит за границы диапазона	
0E05	Неправильный операнд CXXX в команде DCNT	
0E18	Ошибка преобразования BCD	
0E19	Деление на «0»	
0E1A	Операнд битового устройства используется вне диапазона (включая индексные регистры E, F)	
0E1B	Отрицательное значение квадратного корня	
0E1C	Ошибка связи FROM/TO	

Код ошибки	Описание	Действия
0F04	Операнд словного устройства D выходит за границы диапазона	Проверьте D1137 (адрес ошибки) Перезапустите команду корректно
0F05	Неправильный операнд DXXX в команде DCNT	
0F06	Неправильный операнд в команде SFTR	
0F07	Неправильный операнд в команде SFTL	
0F08	Неправильный операнд в команде REF	
0F09	Неправильное использование операндов в командах WSFR, WSFL	
0F0A	Время работы команд TTMR, STMR выходит за границы диапазона	
0F0B	Время работы команды SORT выходит за границы диапазона	
0F0C	Время работы команды TKY выходит за границы диапазона	
0F0D	Время работы команды HKY выходит за границы диапазона	
1000	Неправильный операнд в команде ZRST	
10EF	Операнды E и F неправильные или выходят за границы диапазона	
2000	Превышен лимит при использовании (MTR, ARWS, TTMR, PR, HOUR)	
C400	Командный код не распознан	Ошибка в схеме при неправильном использовании команд или их сочетаний Войдите в режим программирования и исправьте ошибки
C401	Ошибка цикла	
C402	Команды LD / LDI непрерывно используется более 9 раз	
C403	Команда MPS непрерывно используется более 9 раз	
C404	FOR-NEXT превышает 6 уровней вложений	
C405	Команды STL / RET используются между FOR и NEXT Команды SRET / IRET используются между FOR и NEXT Команды MC / MCR используются между FOR и NEXT Команды END / FEND используются между FOR и NEXT	
C407	Команда STL непрерывно используется более 9 раз	
C408	Использование MC / MCR в STL или I / P в STL	
C409	Использование STL/RET в подпрограмме или подпрограмме прерывания	
C40A	Использование MC/MCR в подпрограмме прерывания	
C40B	MC / MCR начинается не с N0 или прерывается	
C40C	MC / MCR соответствует разным значениям N	
C40D	Некорректное использование I / P	
C40E	IRET не следует последней команде FEND SRET не следует последней команде FEND	
C40F	Программа ПЛК и данные в параметрах не были инициализированы	
C41B	Недопустимые команды RUN/STOP для модулей расширения	
C41C	Количество точек ввода/вывода модулей расширения превышает допустимое число	
C41D	Число модулей расширения превышает допустимое	
C41F	Некорректная запись данных в памяти	
C430	Инициализация параллельной ошибки интерфейса	
C440	Аппаратная ошибка высокоскоростного счетчика	
C441	Аппаратная ошибка высокоскоростного компаратора	
C442	Аппаратная ошибка импульсного выхода MCU	
C443	Нет ответа от модуля расширения	
C4EE	В программе нет команды END	
C4FF	Недопустимая команда (несуществующая команда)	

6.3 Отображение и фиксация возникающих ошибок

Устройство	Описание	Энергонезависимость	STOP → RUN	RUN → STOP
M1067	Флаг ошибки выполнения программы	нет	сброс	фиксация
M1068	Флаг фиксации ошибки выполнения программы	нет	фиксация	фиксация
D1067	Код ошибки выполнения программы	нет	сброс	фиксация
D1068	Шаг ошибки выполнения программы	нет	фиксация	фиксация

Код ошибки в D1067	Описание
0E18	Ошибка преобразования BCD
0E19	Деление на «0»
0E1A	Значение с плавающей запятой выходит за пределы допустимого диапазона
0E1B	Отрицательное значение квадратного корня

Функции и порядок работы с CANopen

7.1 Введение в CANopen

- Благодаря простоте монтажа оборудования и прокладки линий связи, возможности непосредственной коммуникации оборудования, широких возможностей по настройке и отладке, стабильной связи и низкой стоимости, сети CANopen нашли широкое применение в промышленной автоматике, автомобильной промышленности, производстве медицинского оборудования, строительстве и т.д.
  - CAN порт, использующий основной протокол CANopen DS301, встроен в ПЛК DVP-ES2-C и может использоваться в ведущем (Master) режиме или ведомом (Slave) режиме. Контроллеры серии «S» расширяются модулем DVPCOPM-SL.
- В данной главе рассматриваются функции CANopen. Функции контролируются специальным реле M1349. При включенном флаге M1349 функции CANopen также включаются. В Master-режиме, функции CANopen могут управлять 16 Slave-устройствами.
- Сети CANopen для ПЛК DVP-ES2-C конфигурируются с помощью ПО CANopen Builder. Коммуникационный адрес (т.н. адрес станции) в CANopen и скорость коммуникации устанавливаются этим ПО. ПО для программирования DVP-ES2-C это, в свою очередь, WPLSoft или ISPSoft.
- Для ознакомления с терминами и определениями, описываемыми в данной главе, см. раздел 7.3.

7.1.1 Описание функций CANopen

- **Функции CAN порта в режиме Мастера (Master).**
  - Поддерживается стандартный протокол CANopen DS301 V4.02.
  - Поддерживается сервис NMT (управление сетью).
    - Поддержка управления состояниями NMT.
      Управление состояниями NMT может использоваться для управления Ведомыми станциями в сетях CANopen.
    - Поддержка контроля ошибок NMT.
      Контроль ошибок NMT применяется для обнаружения отключений Ведомых устройств (Slave). Контроль ошибок NMT может быть двух типов – Heartbeat (генерация проверочных сигналов) и Node Guarding (периодический опрос периферийных устройств). ПЛК DVP-ES2-C поддерживает только тип Heartbeat.
  - Поддерживается сервис PDO (объект данных процесса – обмен данными с объектами).
    - Сообщение PDO используется для оперативной передачи входных/выходных данных.
    - Поддержка макс. 128 RxPDO и макс. 390 байт.
    - Поддержка макс. 128 TxPDO и макс. 390 байт.
    - Типы передачи PDO: синхронный и асинхронный режимы.
  - Поддерживается сервис SDO (объект данных сервиса – сервис обмена данными с объектами).
    - Сервис SDO может быть использован для записи, чтения и настройки параметров Slave-устройств.
    - Поддержка стандартного режима передачи SDO.
    - Поддержка автоматической функции SDO. В Slave-устройство может быть записано до 20 блоков данных.
    - Поддержка использования сервиса SDO в лестничных диаграммах ПЛК для обмена данными с Slave-устройствами.
  - Поддерживается считывание аварийных сигналов от Slave-устройств.
    - Сервис может быть использован для чтения ошибок или аварийных сигналов от Slave-устройств.
    - Может быть считано и сохранено до 5 аварийных сигналов.
    - Может использоваться в лестничных диаграммах ПЛК.
  - Поддерживается сервис SYNC (синхронное обслуживание объектов).
    Несколько устройств могут работать синхронно при помощи данного сервиса.
  - Скорость связи CANopen может быть 20 Кбит/с, 50 Кбит/с, 125 Кбит/с, 250 Кбит/с, 500 Кбит/с, 1 Мбит/с.
  - Отображение поддерживаемых типов данных:

Хранение	Тип данных
8-бит	SINT USINT BYTE
16-бит	INT UINT WORD
32-бит	DINT UDINT REAL DWORD
64-бит	LINT ULINT LREAL LWORD

➢ **Функции CAN порта контроллера DVP-ES2-C в Slave-режиме.**

- ✧ Поддерживается стандартный протокол CANopen DS301 V4.02.
- ✧ Поддерживается сервис NMT (управление сетью).
  - ◆ Поддержка управления состояниями NMT.
    Состоянием DVP-ES2-C в сети CANopen управляет Мастер.
  - ◆ Поддержка контроля ошибок NMT.
    ПЛК поддерживает только тип Heartbeat.
- ✧ Поддерживается сервис PDO (объект данных процесса).
  - ◆ Сообщение PDO используется для оперативной передачи входных/выходных данных.
  - ◆ Поддержка макс. 8 RxPDO и 8 RxPDO.
  - ◆ Типы передачи PDO: синхронный и асинхронный режимы.
- ✧ Поддерживается считывание аварийных сигналов от Slave-устройств.
  Если ошибка или авария возникают в DVP-ES2-C, Master-устройство получает аварийное уведомление.

7.1.2 Области отображения входов/выходов

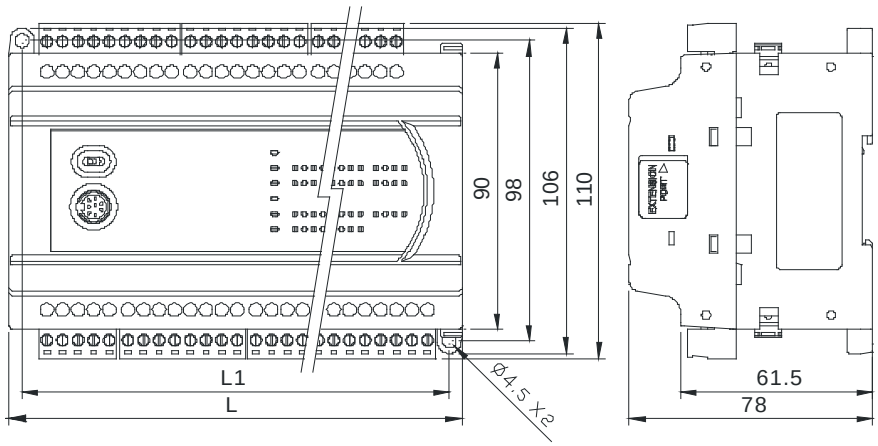
DVP-ES2-C в Master-режиме поддерживает максимум 16 Slave-устройств с адресами в диапазоне 1 – 16. Диапазон регистров выходов: D6250-D6476, входов: D6000-D6226.

Устройство в ПЛК	Область отображения	Длина данных
D6250~D6281	SDO запрос информации, NMT сервисная информация, запрос аварийной информации	64 байт
D6000~D6031	SDO ответная информация, ответная аварийная информация	64 байт
D6282~D6476	RxPDO область отображения	390 байт
D6032~D6226	TxPDO область отображения	390 байт

7.2 Инсталлирование и топология сети

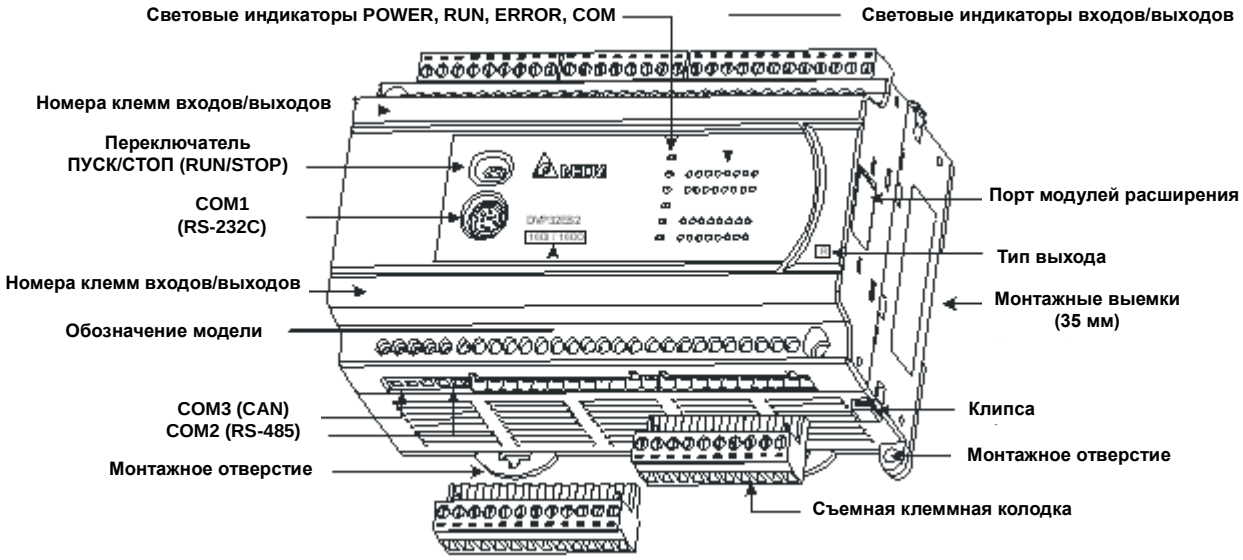
В разделе представлены габариты DVP-ES2-C, интерфейс CAN, топология сети CANopen и дальность связи.

7.2.1 Габариты



Размеры: в миллиметрах

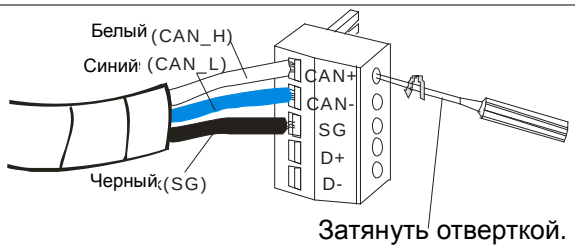
7.2.2 Профиль



7.2.3 Интерфейс CAN и топология сети

➢ Контакты COM3 (интерфейс CAN)

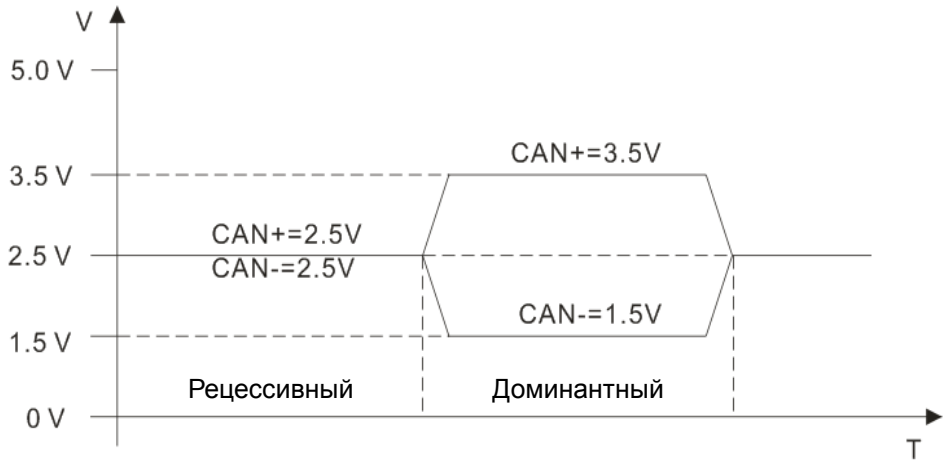
Контакт	Описание
CAN+	CAN-H
CAN-	CAN-L
SG	Заземление



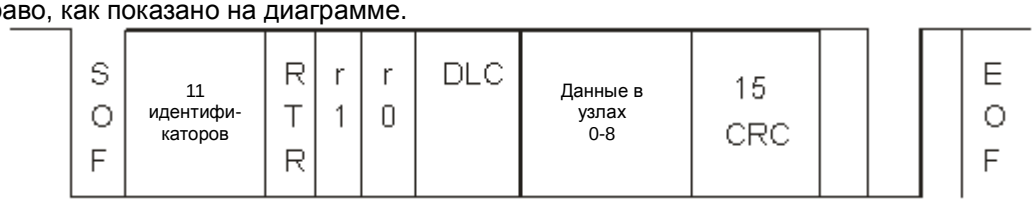
7

➢ **Сигнал CAN и формат фрейма данных**

Сигнал CAN является дифференциальным. Напряжение сигнала это разность между CAN+ и CAN-. Напряжение CAN+ и CAN- используют SG в качестве нулевой точки. Сеть CAN может находиться в двух состояниях. Первое – доминантный уровень, показывающий логический “0”. Второе – рецессивный уровень, показывающий логическую “1”. Уровни сигнала CAN показаны ниже.

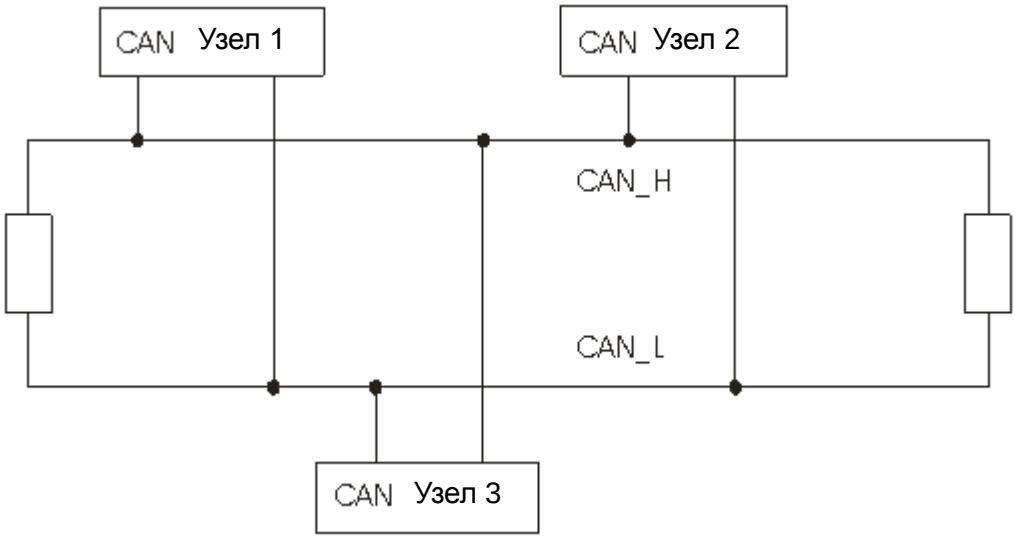


Формат фрейма данных показан ниже. Узлы CAN передают сообщения по сети слева направо, как показано на диаграмме.

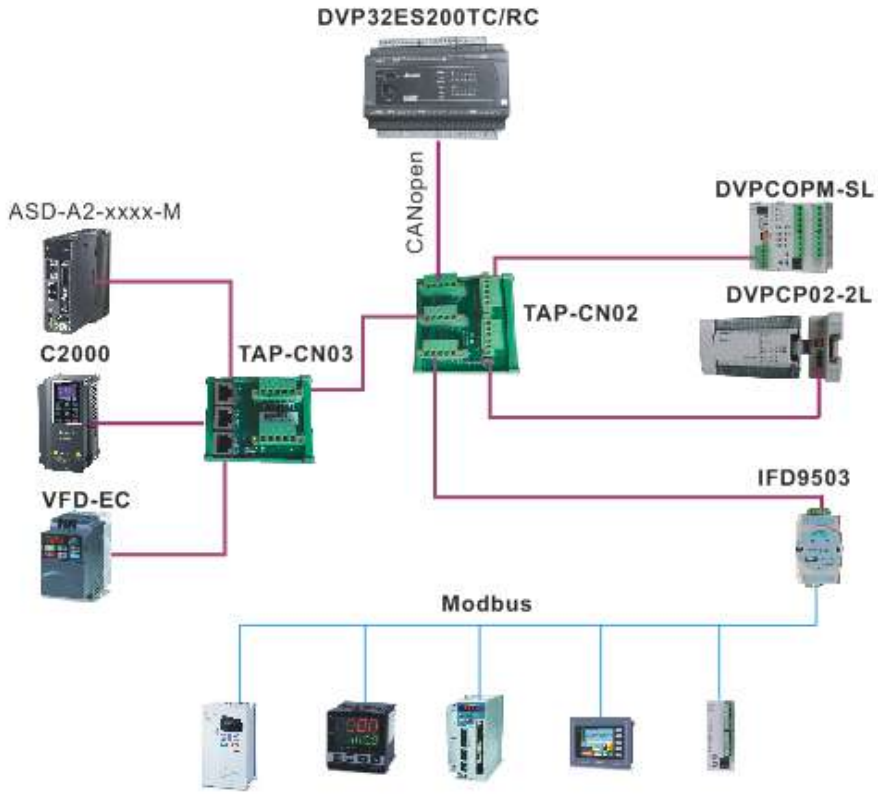


➢ **Конечные точки сети CAN и топологическая структура**

Для обеспечения стабильной связи на оба конца линии устанавливаются согласующие (терминальные) резисторы номиналом 120 ОМ. Топологическая структура сети CAN показана ниже.



➢ Топологическая структура сети **CANopen**



1) Используйте стандартные кабели Delta для создания сетей CANopen. Это «толстый» кабель TAP-CB01 и «тонкие» кабели TAP-CB03/04/05. Сетевые кабели должны прокладываться отдельно от кабелей питания.
2) TAP-TR01. CAN+ и CAN-, располагающиеся на концах сети, используются вместе с резисторами 120 Ом. Используйте стандартные терминальные резисторы Delta TAP-TR01.
3) Ограничения по длине сетей CANopen.
   Расстояние передачи сигнала по сети CANopen зависит от скорости передачи. Соотношение между скоростью и расстоянием передачи показано в таблице.

Скорость передачи (бит/с)	20K	50K	125K	250K	500K	1M
Максимальное расстояние передачи (м)	2500	1000	500	250	100	25

4) Сетевые устройства Delta, применяемые в сетях CANopen:

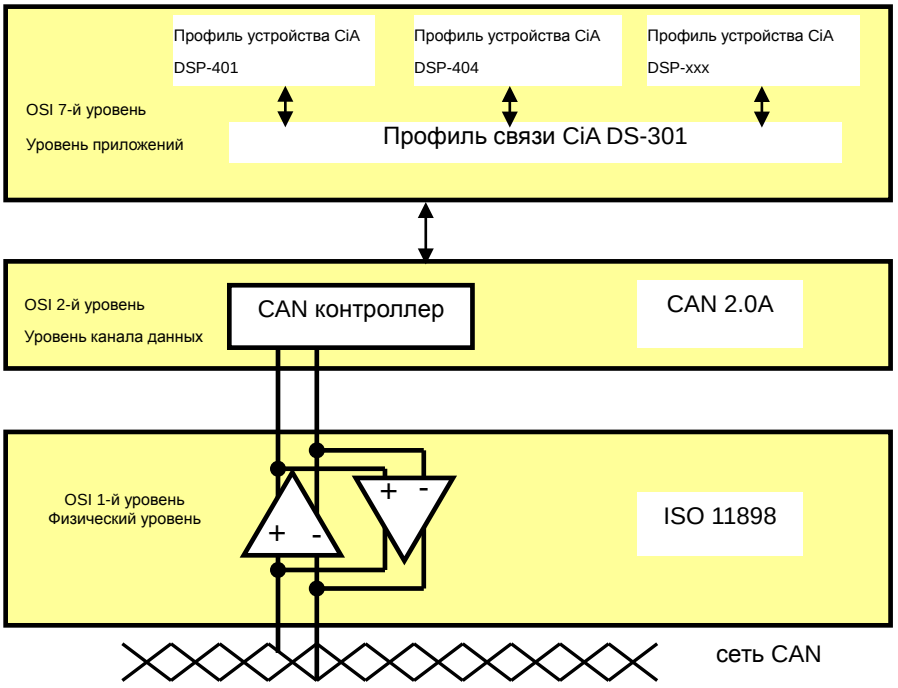
Устройство	Модель	Функция
	DVP32ES200RC DVP32ES200TC	ПЛК DVP-ES2-C со встроенным интерфейсом CAN. В сети может быть как Master, так и Slave-устройством.
	DVPCOPM-SL	DVPCOPM-SL – левосторонний модуль для ПЛК серии «S». Функционирует как Master или Slave в CANopen. ПЛК, подключаемые к DVPCOPM-SL, это DVP-SV, DVP-SV2, DVP-SX2, DVP-SA2, DVP-SE и DVP-EH2-L.
	IFD9503	Преобразователь CANopen в Modbus, подключает устройства (по RS-232 или RS-485), работающие по протоколу Modbus, к сети CANopen (макс. 15 устройств).
	DVPCP02-H2	CANopen Slave-модуль, правосторонний модуль для ПЛК серии EH2. Позволяет подключить такие ПЛК к сети CANopen.
	IFD6503	Инструмент анализа передачи данных по сети CANopen. На концах – интерфейсы CAN и USB. Фиксирует передачу данных, или позволяет узлам передавать данные. Применяется с ПО Netview Builder.
	ASD-A2-xxxx-M сервопривод	Сервопривод со встроенным интерфейсом CAN. Управляет позиционированием, скоростью и моментом.
	C2000/CP2000/C200 преобразователь частоты	ПЧ со встроенной функцией CANopen, управляет позиционированием, скоростью и моментом. Для использования функции CANopen на C2000/CP2000 необходимо поставить карту CMC-COP01. Это карта с интерфейсом CAN. ПЧ C200 обладают встроенным интерфейсом CAN.
	преобразователь частоты серии E + адаптер интерфейса CME-COP01	ПЧ серии E управляет скоростью и моментом. Для использования функции CANopen на ПЧ необходимо поставить адаптер интерфейса CME-COP01

Устройство	Модель	Функция
	TAP-CN01	CANopen сетевая распределительная коробка с резистором120 Ом. Резистор может быть подключен через движковый переключатель.
	TAP-CN02	CANopen сетевая распределительная коробка с резистором120 Ом. Резистор может быть подключен через движковый переключатель.
	TAP-CN03	CANopen сетевая распределительная коробка с резистором120 Ом. Резистор может быть подключен через движковый переключатель.
	TAP-CB03 TAP-CB04 TAP-CB05	Экранированный CANopen-кабель с разъемами RJ45 на обоих концах: TAP-CB03: 0.3 м TAP-CB05：0.5 м TAP-CB04：1 м
	TAP-CB01	CANopen сетевой кабель: TAP-CB01: CANopen магистральный кабель
	TAP-TR01	Резистор 120 Ом с разъемом RJ45.

7.3 Протокол CANopen

7.3.1 Введение в протокол CANopen

Стандарт шины CAN определяет только физический и канальный уровни (см. ISO11898). При практической реализации физический и канальный уровни реализуются в составе аппаратной части. Прикладной уровень Стандарт не определяет. Таким образом, шина CAN сама по себе не является законченной. Ей необходим протокол верхнего уровня (HLP) для определения порядка передачи идентификаторов и байтов данных.
Одним из таких протоколов и является CANopen – открытый протокол верхнего уровня на базе шины CAN, разработанный и поддерживаемый Ассоциацией CiA (CAN-in-Automation). Он разработан на основе протокола CAL (CAN прикладного уровня), используя его инструменты связи и сервисные протоколы.
Протокол CANopen распространяется на Прикладной (верхний) уровень и профиль связи (CiA DS301). Также распространяется на программируемые устройства (CiA 302), кабели и коннекторы (CiA 303-1), единицы СИ и представление префиксов (CiA 303-2).
В модели OSI взаимодействие между стандартом CAN и протоколом CANopen следующее:

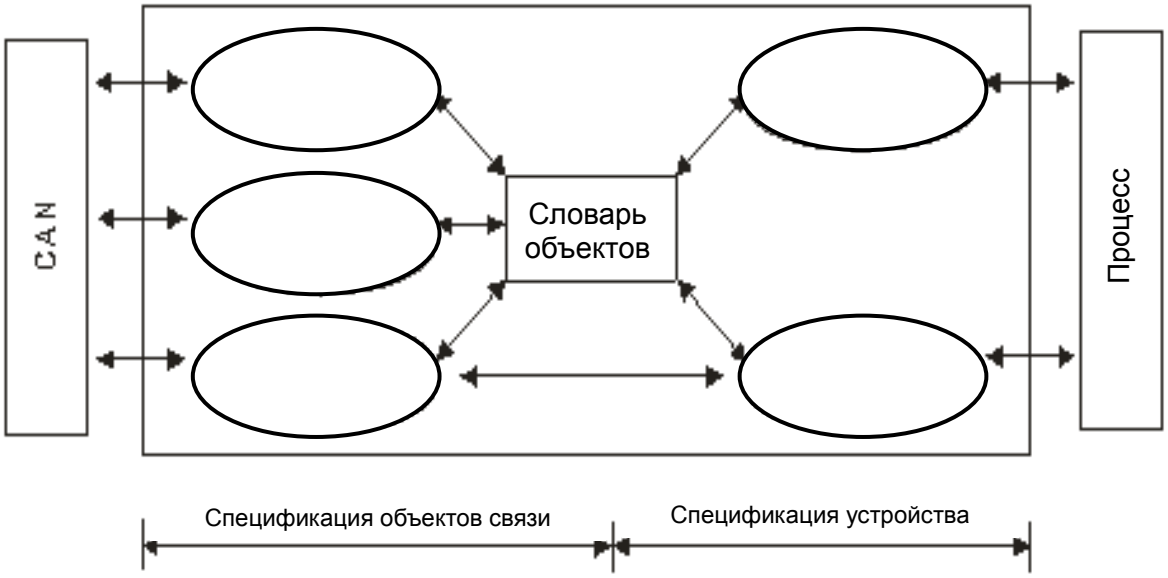


➢ **Словарь объектов**

Для определения стандартных устройств в протоколе CANopen используется объектно-ориентированный подход. Каждое устройство представляется в виде набора коммуникационных объектов (или просто объектов), к которым можно обращаться из сети. Подобный набор называется «Словарь объектов».

Стандартная коммуникационная модель устройства, осуществляющего обмен по протоколу CANopen, представлена на рисунке ниже. Как видно из рисунка, Словарь объектов – это интерфейс между физическим устройством (его внутренними переменными и величинами) и протоколом передачи данных по шине CAN (его стандартной адресацией).

Словарь объектов является ключевой идеей протокола CANopen. Каждый объект имеет 16-ти битный адрес, а каждый элемент внутри объекта определяется 8-ми битным индексом. Любое устройство, работающее по протоколу CANopen, имеет свой словарь объектов, который описывается в так называемом «электронном описании», или EDS файле. Помимо адресов и индексов, данное «электронное описание» содержит информацию о самом устройстве, код производителя, а также сетевые параметры.



7.3.2 Объект связи CANopen

Протокол связи CANopen содержит следующие объекты связи.

➢ **PDO (объект данных процесса)**

- ✧ PDO обеспечивает прямой канал связи с прикладным объектом в устройстве, используется для передачи данных в режиме реального времени, и имеет высокий приоритет. Данный тип сообщения содержит в основном только полезные данные..
- ✧ Существует два типа PDO. В первом случае данные передаются, во втором – при-

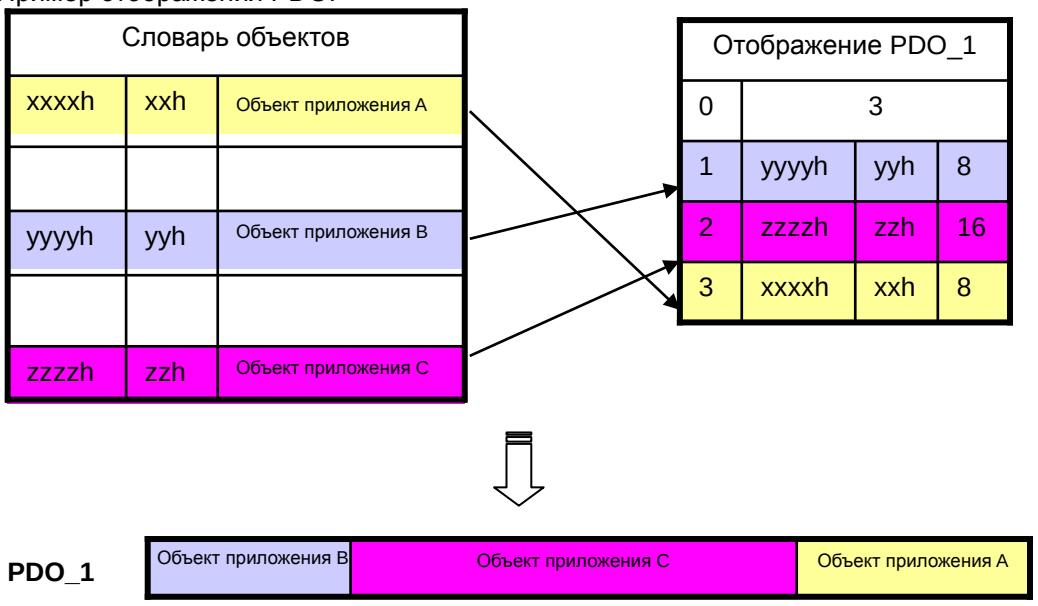
нимаются (Transmit-PDOs (TxPDOs) и Receive-PDOs (RxPDOs)). Устройства, поддерживающие TxPDOs, называются PDO-производителями, а поддерживающие RxPDOs – PDO-потребителями.

- ✧ PDO описывается как “режим производителя/потребителя”. Данные передаются от одного производителя одному или нескольким потребителям. Размер данных от 1 до 8 байт. После передачи данных производителем потребитель не выдает ответного сообщения. Каждый узел автоматически обнаруживает данные и принимает решение, обрабатывать их или нет.
- ✧ Каждый PDO описывается двумя элементами в словаре объектов: параметры связи PDO и параметры отображения PDO:
  Параметры связи PDO: COB-ID, используемый PDO, тип передачи, время задержки, цикл счетчика.
  Параметры отображения PDO: Список объектов в словаре объектов. Также отображается длина блока данных в битах.
- ✧ Режимы передачи PDO: синхронный и асинхронный
  Синхронный: Синхронный периодический и синхронный непериодический
  Асинхронный: PDO передается при изменении данных или по событию.

Режимы передачи.

Тип	Передача PDO				
	Периодическая	Непериодическая	Синхронная	Асинхронная	RTR
0		X	X		
1 – 240	X		X		
254				X	
255				X	

Режим 0: Информация PDO передается только при изменении данных PDO и поступлении синхронизирующего сигнала.
Режим 1~240: Каждый блок информации PDO передается по каждому из 1~240 синхронизирующих сигналов.
Режим 254: Условие отправки сообщения определяется производителем.
Режим 255: PDO передается при изменении данных или по событию. Все данные в PDO должны соответствовать словарю объектов.
Пример отображения PDO:



Формат данных для RxPDO и TxPDO:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Идентификатор объекта	Данные							

➢ **SDO (объект данных сервиса)**

- ✧ SDO используется для построения связи по технологии сервер/клиент между двумя устройствами CANopen. Клиентское устройство может считывать/записывать данные из/в словаря объектов сервера. Режим обращения SDO «клиент/сервер». Каждое устройство CANopen имеет хотя бы одно SDO, создающее канал связи со словарем объектов.

- ✧ Сообщение SDO содержит адрес объекта и индекс элемента, которые используются для поиска в словаре объектов. Таким образом, при помощи SDO можно работать и с составными структурами данных. После того, как SDO-клиент отправляет запрос чтения/записи, SDO-сервер отправляет ответ. И клиент и сервер могут прервать передачу SDO. Запрос и ответ разделяются при помощи различных типов «идентификаторов коммуникационных объектов» (COB-ID).
- ✧ SDO передает данные любой длины. Если длина данных больше 4 байт, то данные передаются сегментами. Последний сегмент содержит флаг завершения.
- ✧ Формат запроса SDO:

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
600 (hex) +Узел-ID	Запрашиваемый код	Адрес объекта		Индекс элемента	Запрашиваемые данные			
		LSB	MSB		бит7-0	бит15-8	бит23-16	бит31-24

Определение запрашиваемого кода в сообщении запроса:

Запрашиваемый код (hex)	Описание
23	Запись 4-байт данных
2B	Запись 2-байт данных
2F	Запись 1-байт данных
40	Чтение данных
80	Остановка функции SDO

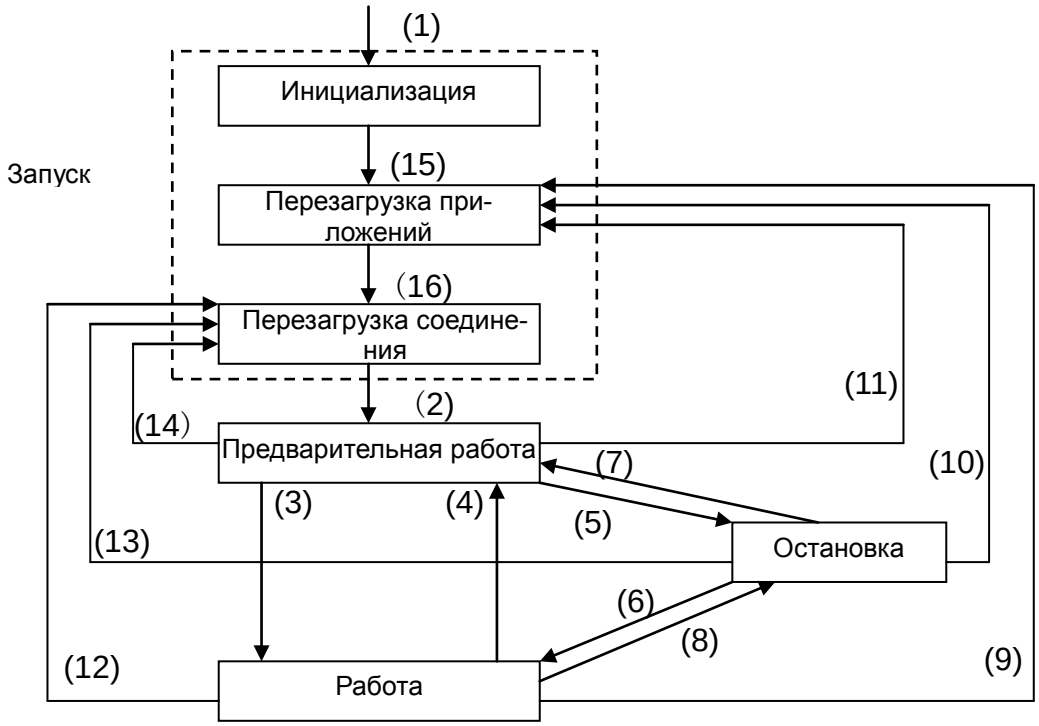
Формат ответного сообщения:

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
580 (hex) +Узел-ID	Ответный код	Адрес объекта		Индекс элемента	Ответные данные			
		LSB	MSB		бит7-0	бит15-8	бит23-16	бит31-24

Определение ответного кода в ответном сообщении:

Запрашиваемый код (hex)	Описание
43	Чтение 4- байт данных
4B	Чтение 2- байт данных
4F	Чтение 1- байт данных
60	Запись1/2/4- байт данных
80	Остановка функции SDO

- ➢ **NMT (объект управления сетью)**
  Управление сетью CANopen соответствует режиму «Мастер/Ведомый». В сети CANopen существует только один NMT-Мастер, остальные узлы (устройства) являются Ведомыми. NMT осуществляет три сервиса: – управление узлами сети, контроль ошибок, управление начальной загрузкой.
  - ✧ **Модуль управления узлами сети**
    Мастер-узел в сети CANopen управляет Ведомыми устройствами посредством команд. Ведомые устройства отрабатывают команды Мастера, не отправляя ответных сообщений. Все узлы CANopen имеют внутренние состояния NMT. Ведомые узлы имеют четыре состояния: инициализация, предварительное рабочее, рабочее и стоп. См. рис. ниже:



(1) При включении питания устройство автоматически входит в режим Инициализации.
(2) После завершения Инициализации, устройство входит в предварительный рабочий режим.
(3)(6) Устройство входит в состояние Работы.
(4)(7) Устройство входит в предварительный рабочий режим.
(5)(8) Устройство входит в состояние Стоп.
(9)(10)(11) Перезагрузка приложений.
(12)(13)(14) Перезагрузка соединения.
(15) После завершения Инициализации, устройство входит в состояние «Перезагрузка приложений».
(16) После выполнения “Перезагрузка приложений” устройство автоматически входит в состояние “Перезагрузка соединения”.

Связь между объектом и его состоянием показана ниже. Сервис объекта связи выполняется только в соответствующем состоянии. Например, SDO выполняется только в рабочем и предварительном рабочем состоянии.

	Запуск	Предв.работа	Работа	Остановка
PDO			X	
SDO		X	X	
SYNC		X	X	
Временная метка		X	X	
EMCY		X	X	
Загрузка	X			
NMT		X	X	X

Формат сообщения о состоянии узла:

COB-ID	Байт 0	Байт 1
0	Спецификатор команды (CS)	Адрес Ведомого (0: широковещательный режим)

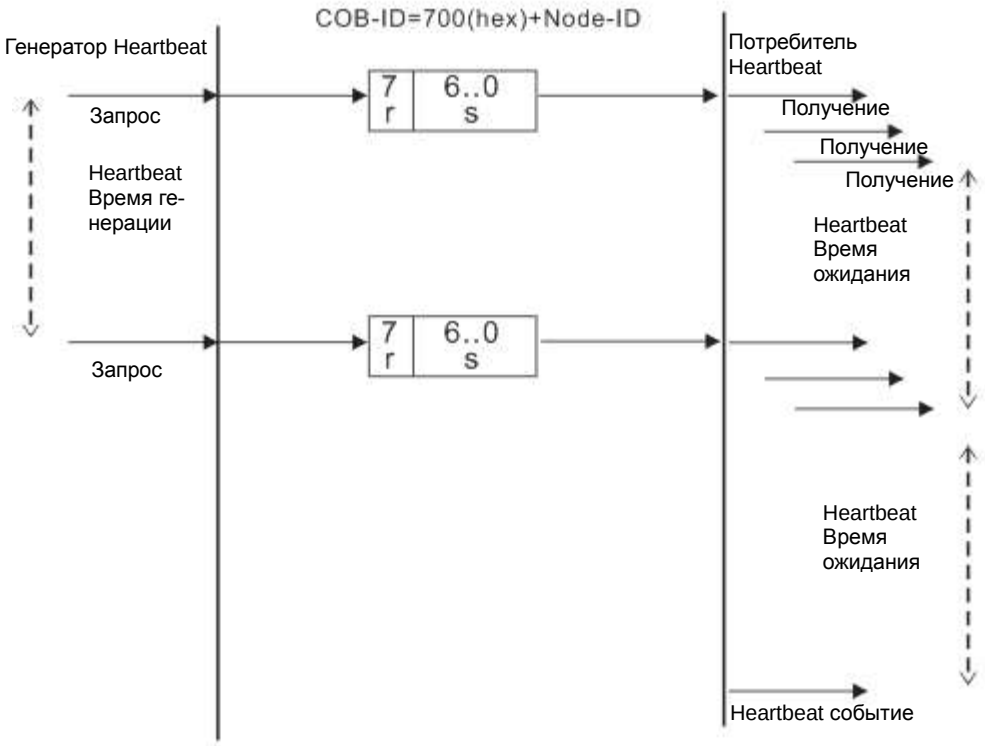
Спецификаторы команд.

Спецификатор (hex)	Функция
01	Инициализация
02	Стоп
80	Вход в предварительное рабочее состояние
81	Перезагрузка приложений
82	Перезагрузка соединения

✧ **Сервис контроля ошибок**

Сервис контроля ошибок применяется для обнаружения отключения узлов в сети. Сервис может быть двух типов: Heartbeat и Node Guarding. ПЛК поддерживает только тип Heartbeat. Например, Master-устройство может обнаруживать отключение Slave-устройств только после запуска Slave-устройствами сервиса Heartbeat.

Принцип Heartbeat показан ниже. Генератор Hearbeat передает сообщение Heartbeat в соответствии с установленным временем. Один или несколько потребителей Heartbeat обнаруживают сообщение, переданное генератором Heartbeat. Если потребитель не получает сигнал генератора в течение заданного времени ожидания, связь по CANopen не осуществляется.



✧ **Сервис загрузок**

После того, как Slave-устройство завершает запуск и входит в предварительное рабочее состояние, происходит передача загрузочного сообщения.

➢ Другие объекты связи **CANopen (SYNC и EMCY)**

✧ **SYNC (объект синхронизации)**

Объект синхронизации – это сообщение, периодически транслируемое Мастером в сеть в широковещательном режиме, и служит для реализации тактового сигнала для всей сети. Ведомые устройства, приняв объект синхронизации, предпринимают какие-либо действия по осуществлению связи в синхронном режиме или не предпринимают в зависимости от своих настроек.

Например, при управлении несколькими приводами, команды будут выполняться не сразу по их получении, а только после появления на шине объекта синхронизации. Таким образом, можно достичь синхронной работы большого количества устройств.

Формат сообщения SYNC:

COB-ID	80 (hex)

✧ **Объекты Аварийных сообщений**

Объекты Аварийных сообщений применяются устройствами CANopen для индикации внутренних ошибок. При возникновении ошибки в устройстве, устройство передает Аварийное сообщение (включая код ошибки) и устройство входит в состояние ошибки. После устранения ошибки устройство посылает Аварийное сообщение с кодом ошибки «0» и входит в нормальное состояние.

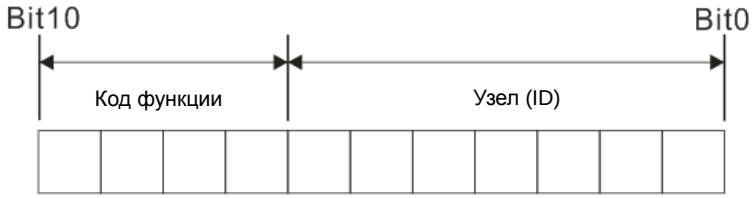
Формат Аварийных сообщений:

COB-ID	**Байт 0**	**Байт 1**	**Байт 2**	**Байт 3**	**Байт 4**	**Байт 5**	**Байт 6**	**Байт 7**
80 (hex) +Узел-ID	Код ошибки		Регистр ошибки	Код ошибки, определяемый производителем				
	LSB	MSB						

Примечание: Значение регистра ошибок привязано к регистру 1001 (hex) словаря объектов. Если значение 0, то ошибок нет. Значение 1 сигнализирует о внешней ошибке. Значение H'80 сообщает о возникновении внутренней ошибки в устройстве.

7.3.3 Стандартный набор подключений

Чтобы уменьшить конфигурационную нагрузку на сеть, CANopen определяет идентификаторы по умолчанию. В стандартном наборе подключений структура 11-битного идентификатора выглядит следующим образом:



Поддерживаемые объекты и соотносимые с ними COB-ID приведены ниже.

- Объекты широковещательного режима связи стандартного набора подключений

Объект	Функциональный код	COB-ID	Регистр в словаре объектов
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
Временная метка	0010	256 (100h)	1012h, 1013h

- Объекты связи стандартного набора подключений

Объект	Функциональный код	COB-ID	Регистр в словаре объектов
Тревожн.	0001	129 (81h)–255 (FFh)	1014h, 1015h
PDO1 (TX)	0011	385 (181h)–511 (1FFh)	1800h
PDO1 (RX)	0100	513 (201h)–639 (27Fh)	1400h
PDO2 (TX)	0101	641 (281h)–767 (2FFh)	1801h
PDO2 (RX)	0110	769 (301h)–895 (37Fh)	1401h
PDO3 (TX)	0111	879 (381h)–1023 (3FFh)	1802h
PDO3 (RX)	1000	1025 (401h)–1151 (47Fh)	1402h
PDO4 (TX)	1001	1153 (481h)–1279 (4FFh)	1803h
PDO4 (RX)	1010	1281 (501h)–1407 (57Fh)	1403h
SDO (TX)	1011	1409 (581h)–1535 (5FFh)	1200h
SDO (RX)	1100	1537 (601h)–1663 (67Fh)	1200h
NMT Контроль ошибок	1110	1793 (701h)–1919 (77Fh)	1016h, 1017h

7.4 Отправка SDO, NMT и чтение Аварийных сообщений из программы контроллера

Для каждого типа объектов связи в программе контроллера выделен определенный диапазон регистров данных. Меняя содержимое регистров, можно формировать каждый раз новый запрос, а прием ответного сообщения осуществляется в свой выделенный диапазон регистров данных.

Ниже в таблицах приведены выделенные диапазоны регистров данных в памяти контроллера под каждый тип объекта связи.

Регистры ПЛК	Тип объекта связи	Длина данных
D6250~D6281	запрос SDO, сервисное сообщение NMT и запрос Аварийного сообщения	64 байта
D6000~D6031	ответное сообщение SDO и ответное Аварийное сообщение	64 байта

1> В каждый момент времени Мастер может отправлять только одно сообщение типа SDO, NMT или Emergency одному и тому же Ведомому.

2> После отправки сообщения необходимо обнулить выделенные регистры ПЛК.

7.4.1 Структура данных запроса SDO

Отправка SDO через лестничную диаграмму позволяют считывать/записывать параметры Ведомого устройства.

➢ Формат данных запроса SDO:

Регистры ПЛК	Запрос		
		Старший байт	Младший байт
D6250	Заголовок сообщения	Идентификатор запроса	Команда (постоянно 01)
D6251		Зарезервировано	Размер
D6252		Тип	Сетевой адрес
D6253	Данные сообщения	Старший байт адреса в сло-	Младший байт адреса в сло-
D6254		Резервный	Индекс
D6255		Данные 1	Данные 0
D6256		Данные 3	Данные 2
D6257 ~ D6281		Зарезервировано	

✧ Команда: фиксировано “01”.
✧ Идентификатор запроса (ReqID). При отправке запроса SDO ему присваивается ReqID для идентификации Мастером. При чтении/записи другого сообщения SDO, его идентификатор должен быть изменен. Таким образом, выбор какое SDO «читать/записывать» определяется указанием “ReqID”. Диапазон ReqID: 00 (hex) ~ FF (hex).
✧ Размер: Длина данных сообщения. Отсчет начинается с D6253 и ведется побайтно. При чтении длина фиксируется как 4, а при записи как 4 плюс байты с типом данных регистров и индексов, максимально 8. Однако при записи, если тип данных регистров и индексов – слово, то величина 6, если байт – 5.
✧ Сетевой адрес (Node ID): адрес узла в сети CANopen, которому предназначено SDO сообщение.
✧ Тип: 01 показывает доступ к чтению; 02 – доступ к записи.

➢ Формат данных ответного сообщения SDO:

Регистры ПЛК	Ответное сообщение		
		Старший байт	Младший байт
D6000	Заголовок сообщения	Идентификатор ответа	Код состояния
D6001		Зарезервировано	Размер
D6002		Тип	Сетевой адрес
D6003	Данные сообщения	Старший байт адреса в сло-	Младший байт адреса в сло-
D6004		Зарезервировано	Индекс
D6005		Данные 1	Данные 0
D6006		Данные 3	Данные 2
D6007 ~ D6031		Зарезервировано	

✧ Код состояния:
Коды состояния в ответном сообщении:

Код состояния	Описание
0	Нет запроса передачи данных
1	Сообщение SDO передано успешно
2	Сообщение SDO сообщение передается
3	Ошибка: превышено время ожидания SDO
4	Ошибка: недопустимый код команды
5	Ошибка: недопустимый размер данных для передачи.
6	Ошибка: недопустимый размер ответных данных.
7	Ошибка: оборудование для передачи сообщения занято.
8	Ошибка: недопустимый тип
9	Ошибка: некорректный адрес узла
0A	Сообщение об ошибке (см. код ошибки для ответного сообщения SDO)
0B~FF	Зарезервировано

✧ Идентификатор ответа (ResID)..(см. описание идентификатора запроса)
✧ Размер: Длина данных сообщения. Макс. 20 байт. Ед. изм.: байт. При записи – 4; при чтении длину данных определяют типы адреса и индекса.
✧ Узел ID: Адрес узла целевого оборудования в сети CANopen.
✧ Тип: В ответном сообщении SDO 43 (hex) означает чтение 4 байт данных; 4B (hex)

означает чтение 2 байт данных; 4F (hex) означает чтение 1 байта данных; 60 (hex) означает запись 1/2/4 байт данных; 80 (hex) означает остановку команды SDO.

Пример 1: Запись 010203E8 (hex) в 2109/0 (адрес/индекс) Ведомого № 3 посредством запроса SDO, и тип данных 2109/0 (адрес/индекс) – двойное слово (32 бит).

- Запрос:

Регистры ПЛК	Запрос (сообщение)		
		Старший байт (Hex)	Младший байт (Hex)
D6250	Заголовок сообщения	Идентификатор запроса =01	Команда =01
D6251		Зарезервировано =0	Размер =8
D6252		Тип =02	Сетевой адрес =03
D6253	Данные сообщения	Адрес, старший байт =21	Адрес, младший байт =09
D6254		Зарезервировано =0	Индекс =0
D6255		Данные 1=03	Данные 0=E8
D6256		Данные 3=01	Данные 2=02

- Ответ:

Регистры ПЛК	Ответное сообщение		
		Старший байт (Hex)	Младший байт (Hex)
D6000	Заголовок сообщения	Идентификатор ответа =01	Команда =01
D6001		Зарезервировано =0	Размер =4
D6002		Тип =60	Сетевой адрес =03
D6003	Данные сообщения	Адрес, старший байт =21	Адрес, младший байт =09
D6004		Зарезервировано =0	Индекс =0
D6005		Данные 1=00	Данные 0=00
D6006		Данные 3=00	Данные 2=00

Пример 2: Чтение значения 2109/0 (адрес/индекс) Ведомого № 3 посредством запроса SDO, и тип данных 2109/0 (адрес/индекс) – двойное слово (32 бит).

- Запрос:

Регистры ПЛК	Запрос (сообщение)		
		Старший байт (Hex)	Младший байт (Hex)
D6250	Заголовок сообщения	Идентификатор запроса =01	Команда =01
D6251		Зарезервировано =0	Размер =4
D6252		Тип =01	Сетевой адрес =03
D6253	Данные сообщения	Адрес, старший байт =21	Адрес, младший байт =09
D6254		Зарезервировано =0	Индекс =0
D6255		Данные 1=00	Данные 0=00
D6256		Данные 3=00	Данные 2=00

- Ответ:

Регистры ПЛК	Ответное сообщение		
		Старший байт (Hex)	Младший байт (Hex)
D6000	Заголовок сообщения	Идентификатор ответа =01	Команда =01
D6001		Зарезервировано =0	Размер =8
D6002		Тип =43	Сетевой адрес =03
D6003	Данные сообщения	Адрес, старший байт =21	Адрес, младший байт =09
D6004		Зарезервировано =0	Индекс =0
D6005		Данные 1=03	Данные 0=E8
D6006		Данные 3=01	Данные 2=02

7.4.2 Структура данных сообщения NMT

Сервис NMT может быть использован для управления сетью и реализации запуска, управления работой, остановки и перезапуска Ведомых станций.

➢ Формат запроса NMT:

Регистры ПЛК	Запрос (сообщение)		
		Старший байт (Hex)	Младший байт (Hex)
D6250	Заголовок сообщения	Идентификатор запроса	Команда (постоянно 01)
D6251		Зарезервировано	Размер (постоянно 04)
D6252		Тип (постоянно 03)	Сетевой адрес
D6253	Данные сообщения	Зарезервировано	Сервисный код NMT
D6254		Зарезервировано	Сетевой адрес

✧ Команда: фиксировано “01”.

✧ Идентификатор запроса. Присваивается при отправлении запроса NMT для идентификации Master-устройством. Перед чтением/записью другого NMT-сообщения, предыдущий идентификатор должен быть изменен. Т.е. чтение/запись NMT определяется изменением идентификатора. Диапазон: 00 (hex) ~ FF (hex).

✧ Сетевой адрес: Адрес узла целевого устройства в сети CANopen.

✧ Сервисные коды NMT:

Сервисный код NMT (Hex)	Функция
01	Запуск удаленного узла
02	Стоп удаленного узла
80	Запуск состояния предварительной работы
81	Перезагрузка приложения
82	Перезагрузка связи

➢ Формат данных ответного сообщения NMT:

Регистры ПЛК	Ответное сообщение		
		Старший байт	Младший байт
D6000	Заголовок сообщения	Идентификатор ответа	Код состояния
D6001		Зарезервировано	Зарезервировано
D6002		Зарезервировано	Сетевой адрес

✧ Когда код состояния = 1, операция NMT завершена успешно. Когда код не равен 1, NMT не выполняется, необходимо проверить корректность данных в запросе NMT.

✧ Сетевой адрес: Адрес узла целевого устройства в сети CANopen.

Пример 1: Остановка Slave № 3 с помощью NMT

◆ Запрос:

Регистры ПЛК	Запрос (сообщение)		
		Старший байт (Hex)	Младший байт (Hex)
D6250	Заголовок сообщения	Идентификатор запроса =01	Команда =01
D6251		Зарезервировано =0	Размер =04
D6252		Тип =03	Сетевой адрес =03
D6253	Данные сообщения	Зарезервировано	Сервисный код NMT =02
D6254		Зарезервировано	Сетевой адрес =03

◆ Ответ:

Регистры ПЛК	Ответное сообщение		
		Старший байт (Hex)	Младший байт (Hex)
D6000	Заголовок сообщения	Идентификатор ответа =01	Код состояния =01
D6001		Зарезервировано =0	Зарезервировано =0
D6002		Зарезервировано =0	Сетевой адрес =03

7.4.3 Структура данных Аварийного запроса

Считывается аварийная информация и ошибки Slave-устройств.

➢ Формат данных Аварийного запроса:

Регистры ПЛК	Запрос (сообщение)		
		Старший байт	Младший байт
D6250	Заголовок сообщения	Идентификатор запроса	Команда (постоянно 01)
D6251		Зарезервировано	Размер (постоянно 00)
D6252		Тип (постоянно 04)	Сетевой адрес
D6253~D6281	Данные со-	Зарезервировано	

- ✧ Команда: фиксировано - “01”.
- ✧ Идентификатор запроса. Присваивается при отправлении Аварийного запроса для идентификации Master- устройством. Перед чтением/записью другого Аварийного сообщения, предыдущий идентификатор должен быть изменен. Т.е. чтение/запись определяется изменением идентификатора. Диапазон: 00 (hex) ~ FF (hex).
- ✧ Сетевой адрес: Адрес узла целевого устройства в сети CANopen.

➢ Формат данных ответного сообщения от Ведомого:

Регистры ПЛК	Ответное сообщение		
		Старший байт (Hex)	Младший байт (Hex)
D6000	Заголовок сообщения	Идентификатор ответа	Код состояния
D6001		Зарезервировано	Фиксированный размер 2A
D6002		Тип (постоянно 04)	Сетевой адрес
D6003	Данные сообщения	Общее количество данных	Количество сохраненных данных
D6004		Данные 1	Данные 0
D6005		Данные 3	Данные 2
D6006		Данные 5	Данные 4
D6007		Данные 7	Данные 6
D6008 ~ D6011		Авария 2	
D6012 ~ D6015		Авария 3	
D6016 ~ D6019		Авария 4	
D6020~ D6023		Авария 5	
D6024~ D6031		Зарезервировано	

- ✧ Команда: фиксировано “01”.
- ✧ Когда код состояния равен 1, чтение Аварийного сообщения выполнено. Если код не равен 1, чтение аварийного сообщения не произведено, проверьте корректность Аварийного запроса.
- ✧ Сетевой адрес: Адрес узла целевого устройства в сети CANopen.
- ✧ Общее количество данных: Общее количество Аварийных сообщений, получаемых Master-устройством CANopen от Slave-устройств.
- ✧ Количество сохраненных данных: Последние Аварийные сообщения, получаемые Мастером от Ведомых (не более 5).
- ✧ Данные в D6004-D6007 содержат данные об Аварии № 1. Каждое Аварийное сообщение состоит из 8 байт.
  Формат тревожных сообщений показан ниже. Данные 0 -7 соответствуют байтам 0 – 7.

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
80（hex）+Узел-ID	Код ошибки Аварийного сообщения		Ошибка регистров хранения	Код ошибки, определяемый производителем				

Пример 1: Чтение Аварийных сообщений от Ведомого № 2, сообщения посылаются последовательно.

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
82（hex）	43	54	20	14	0	0	0	0

COB-ID	Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	Байт 5	Байт 6	Байт 7
82（hex）	42	54	20	15	0	0	0	0

◆ Запрос:

Регистры ПЛК	Запрос (сообщение)		
		Старший байт	Младший байт
D6250	Заголовок сообщения	Идентификатор запроса =01	Команда =01
D6251		Зарезервировано	Размер =0
D6252		Тип =04	Сетевой адрес =02

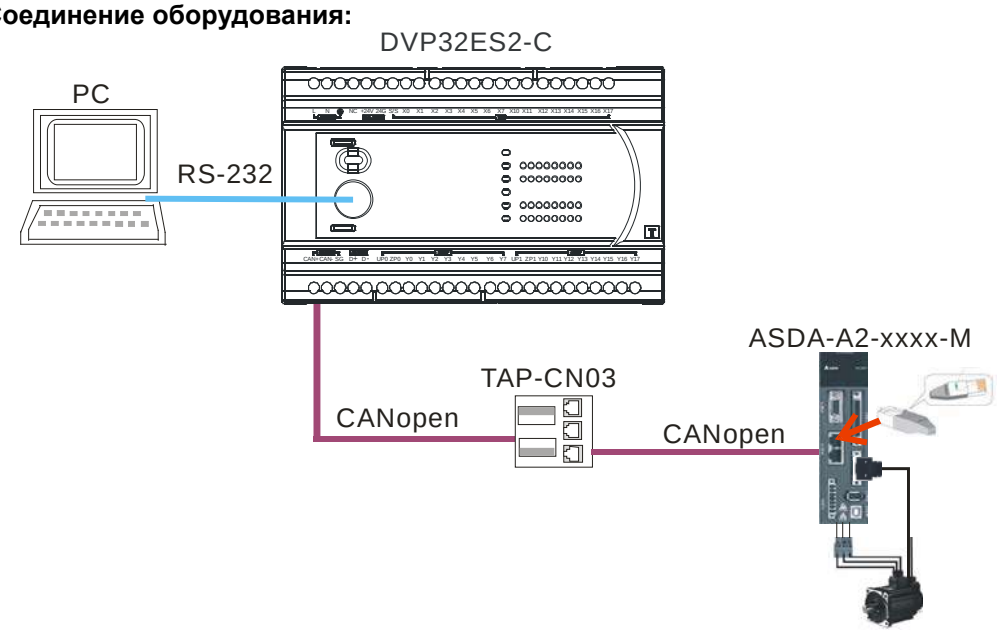
◆ Ответное сообщение

Регистры ПЛК	Ответное сообщение		
		Старший байт	Младший байт
D6000	Заголовок сообщения	Идентификатор ответа =01	Код состояния =01
D6001		Зарезервировано =0	Размер =2A (Hex)
D6002		Тип =04	Сетевой адрес =02
D6003	Данные сообщения	Общее количество данных=1	Количество сохраненных данных=1
D6004		Данные 1=54	Данные 0=42
D6005		Данные 3=15	Данные 2=20
D6006		Данные 5=0	Данные 4=0
D6007		Данные 7=0	Данные 6=0
D6004		Данные 1=54	Данные 0=43
D6005		Данные 3=14	Данные 2=20
D6006		Данные 5=0	Данные 4=0
D6007		Данные 7=0	Данные 6=0

7.4.4 Пример отправки SDO через лестничную диаграмму

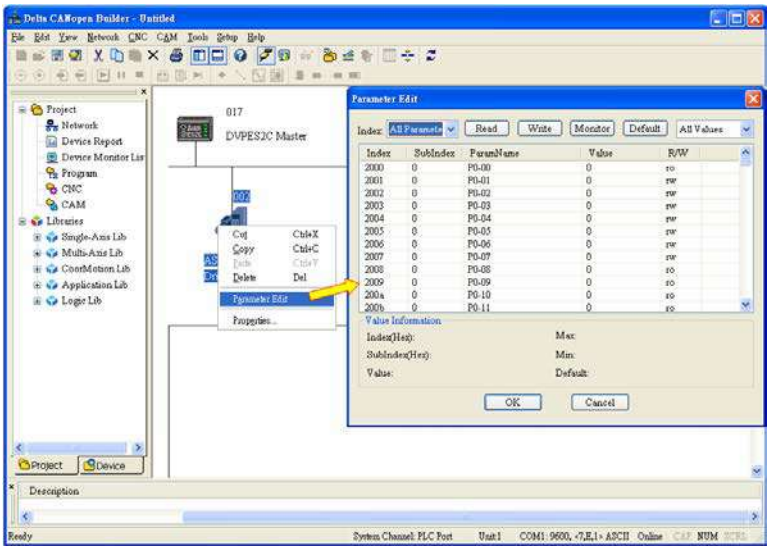
➢ **Задача управления:**
Чтение значения P0-09 сервопривода в цикле через SDO.

➢ **Соединение оборудования:**



➢ **Соответствие параметров Ведомого и Адресам/индексам (словарь объектов)**

Адрес/индекс, относящийся к P0-09 сервопривода, является 2009/0. На интерфейсе ПО конфигурации сети щелкните правой кнопкой мыши на значок сервопривода, выберите пункт «Parameter Edit», в появившемся окне вы увидите список параметров сервопривода и соответствующие им адреса/индексы (index/subindex).

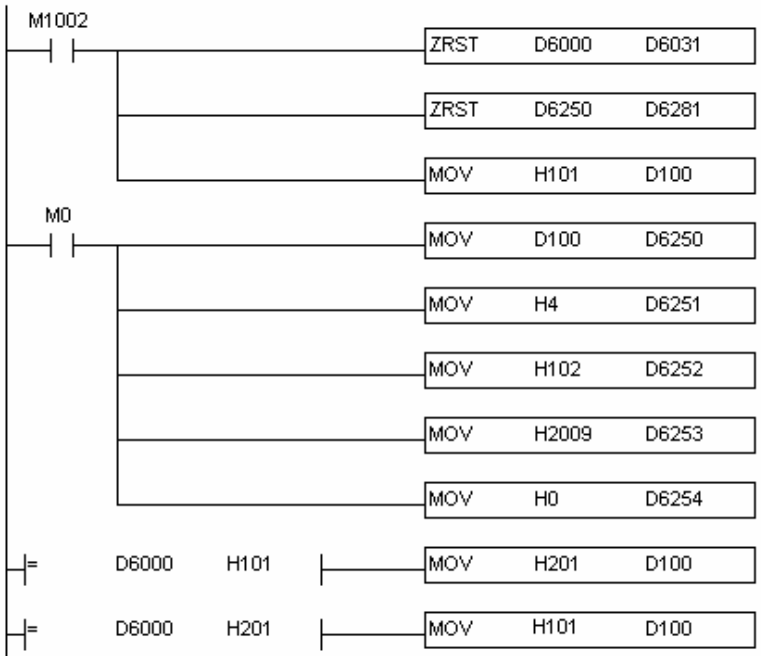


Для получения дополнительной информации о том, как работать с программой конфигурации сети, см. раздел 11.1.1 в файле справки ПО CANopen Builder.

➢ **Описание регистров ПЛК, работающих с запросом:**

Регистры ПЛК		Содержимое (Hex)	Описание	
			Старший байт (Hex)	Младший байт (Hex)
SDO Область отображения сообщений запроса	D6250	0101	Идентификатор запроса = 01	Команда = 01
	D6251	0004	Зарезервировано	Размер = 04
	D6252	0102	Тип = 01	Сетевой адрес = 02
	D6253	2009	Адрес, старший байт = 20	Адрес, младший байт = 09
	D6254	0000	Зарезервировано	Индекс = 00

➢ **Редактирование лестничной диаграммы через WPLsoft**



При включении M0, DVP-ES2-C посылает первый запрос с идентификатором 01. При удачном получении ответа от Ведомого содержимое регистра D6000 должно быть 0101 (hex). В программе проверяется содержимое регистра D6000. Если оно 0101 (hex), то содержимое регистра D6250 меняется на 0201 и наоборот. После этого DVP-ES2-C снова посылает запрос. При помощи такой конструкции можно добиться чтения в реальном масштабе времени. Данные ответного сообщения попадают в регистры D6000~D6005. Значение D6005: 100 (hex) – это считанное значение P0-09.

➢ **Описание регистров ПЛК, работающих с ответным сообщением:**

Регистры ПЛК		Содержимое (Hex)	Описание	
			Старший байт (Hex)	Младший байт (Hex)
SDO Область отображения сообщений ответного сообщения	D6000	0101	Идентификатор ответа = 01	Код состояния = 01
	D6001	0008	Зарезервировано	Размер = 08
	D6002	4302	Тип = 43	Сетевой адрес = 02
	D6003	2009	Адрес, старший байт = 20	Адрес, младший байт = 09
	D6004	0004	Зарезервировано	Индекс = 00
	D6005	0100	Данные 1= 01	Данные 0= 00
	D6006	0100	Данные 3= 00	Данные 2= 00

7.5 Индикаторы и устранение неисправностей

На DVP-ES2-C имеется 6 светодиодных индикаторов. Индикатор питания (POWER) показывает нормальную подачу питания, индикаторы RUN и ERROR показывают состояние выполнения внутренних программ, и индикатор COM3 показывает состояние связи в сети CANopen.

7.5.1 Описание индикаторов

- **Индикатор питания (POWER)**

Индикатор	Описание	Метод устранения
Зеленый индикатор не горит или мигает	Недостаточное напряжение питания	Убедись, что напряжение питания в разрешенном диапазоне
Зеленый индикатор горит постоянно	Напряжение питания в норме	Действий не требуется

- **Индикатор RUN**

Индикатор	Описание	Метод устранения
Зеленый индикатор горит постоянно	ПЛК работает	Действий не требуется
Зеленый индикатор не горит	ПЛК остановлен	Запустите ПЛК переключателем RUN/STOP или ПО WPLSoft

- **Индикатор ERROR**

Индикатор	Описание	Метод устранения
Индикатор не горит	ПЛК работает нормально	Действий не требуется
Красный индикатор мигает	Синтаксические ошибки в программе или операнд (инструкция) находятся вне допустимого диапазона	Проверьте ошибку по значению в специальном регистре D1004; найдите местоположение ошибки по значению в D1137. Описания кодов ошибок в D1004 см. «ПЛК серии ES2. Руководство по Программированию»
Красный индикатор горит постоянно	Превышено допустимое время скана программы	Сократите время выполнения программ ПЛК или используйте команду WTD

- **Индикатор COM3（CANopen）**

Индикатор	Описание	Метод устранения
Зеленый индикатор горит постоянно	ПЛК работает нормально	Действий не требуется
Единичное мигание зеленого индикатора	DVP-ES2-C остановлен	Осуществляется загрузка конфигурации и параметров сети
Зеленый индикатор мигает	DVP-ES2-C в Slave-режиме, предоперационный режим; DVP-ES2-C в Master-режиме, некоторые Slave-устройства не в сети.	1. Проверьте правильность подключения кабеля CANopen 2. Проверьте совпадение скоростей передачи данных Master/Slave-устройств 3. Проверьте настройки Slave-устройств в сети 4. Проверьте нахождение Slave-устройств в сети
Двойное мигание красного индикатора	Один или несколько Ведомых не в сети	1. Проверьте кабель CANopen на соответствие стандарту 2. Проверьте подключение к обоим концам шины CANopen терминальных резисторов
Одиночное мигание красного индикатора.	Счетчик ошибок CAN контроллера достиг предупредительного уровня	1. Проверьте правильность подключения кабеля CANopen 2. Проверьте подключение к обоим концам шины CANopen терминальных резисторов 3. Проверьте наличие помех вокруг кабеля CANopen
Красный индикатор горит постоянно	Нет подключения к шине	1. Проверьте правильность подключения кабеля CANopen 2. Проверьте совпадение скоростей передачи данных Master/Slave-устройств

7.5.2 Отображение состояния сетевых узлов CANopen

- Когда специальное реле M1349 в DVP-ES2-C включается, функция CANopen запускается и D9980~D9998 используются в качестве специальных регистров:

Специальный регистр	Функция
D9980	Отображение состояния DVP-ES2-C.
D9981~D9996	Отображение состояния 16 узлов в сети
D9998	Мониторинг состояния всей сети CANopen

- В Master-режиме DVP-ES2-C поддерживает максимум 16 Slave-устройств - от узла 1 до узла 16. D9998 используется для отображения состояния узлов 1 - 16 в сети. 16 бит D9998 относится к 16 Slave-устройствам (см. таблицу).

Бит	b7	b6	b5	b4	b3	b2	b1	b0
Узел	Узел 8	Узел 7	Узел 6	Узел 5	Узел 4	Узел 3	Узел 2	Узел 1
Bit	**b15**	**b14**	**b13**	**b12**	**b11**	**b10**	**b9**	**b8**
Узел	Узел 16	Узел 15	Узел 14	Узел 13	Узел 12	Узел 11	Узел 10	Узел 9

Когда узел работает нормально, соответствующий бит отключен, если в работе узла возникают проблемы (например, невозможна инициализация и т.п.), соответствующий бит включается.

- Код ошибки для каждого узла отображается в соответствующем специальном регистре:

Специальный регистр	D9981	D9982	D9983	D9984	D9985	D9986	D9987	D9988
Узел	Узел 1	Узел 2	Узел 3	Узел 4	Узел 5	Узел 6	Узел 7	Узел 8
Специальный регистр	**D9989**	**D9990**	**D9991**	**D9992**	**D9993**	**D9994**	**D9995**	**D9996**
Узел	Узел 9	Узел10	Узел11	Узел12	Узел13	Узел14	Узел15	Узел16

- Коды в D9981~D9996 когда DVP32ES2-C в Master-режиме:

Код	Индикация	Метод устранения
E0	DVP-ES2-C Master-модуль получил аварийное сообщение от Slave-устройства	Прочитайте сообщение через программу ПЛК
E1	Длина данных PDO, возвращаемых от Slave-устройств не согласуется со значением, установленным в списке узлов.	Измените длину данных PDO и повторно загрузите их.
E2	PDO от Ведомого не получено.	Проверьте корректность настроек.
E3	Автозагрузка SDO не происходит.	Проверьте корректность настроек автозагрузки SDO.
E4	Конфигурирование параметров PDO не происходит.	Проверьте корректность настроек PDO.
E5	Ошибка настройки ключевых параметров.	Убедитесь, что связь установлена именнос теми Slave-устройствами, которые обозначены в настройках
E6	Slave-устройство не существует в сети	Убедитесь в исправности питания на Slave-устройство и его надлежащее подключение к сети.
E7	Превышение времени ожидания при управлении Slave-устройством	
E8	Повторяются сетевые адреса для Master и Slave.	Переустановите ID узлов Master и Slave-устройств корректно.

- Коды в D9980 когда DVP32ES2-C в Master-режиме:

Код	Индикация	Метод устранения
F1	Slave-устройство не было добавлено в список узлов из ПО CANopen Builder	Добавить Slave-устройство в список узлов и произвести повторную загрузку.
F2	Данные загружаются в DVP-ES2-C	Дождитесь завершения загрузки конфигурационных данных.
F3	Состояние ошибки DVP-ES2-C	Перезагрузите конфигурацию параметров
F4	Обнаружено отключение от шины.	Убедитесь, что кабели CANopen подключены правильно и убедитесь, что все узлы работают на одинаковой скорости и включите повторно.
F5	DVP-ES2-C: неправильный адрес узла	Адреса узлов для DVP-ES2-C устанавливаются в диапазоне: 1~127.
F8	Внутренняя ошибка, ошибка во внутренней памяти	После повторного включения, внесите изменения, если ошибка не устранена.
FB	Буфер сообщений на отправку DVP-ES2-C заполнен.	Убедитесь, что кабели CANopen подключены правильно и включите повторно.
FC	Буфер принимаемых сообщений DVP-ES2-C заполнен.	Убедитесь, что кабели CANopen подключены правильно и включите повторно.

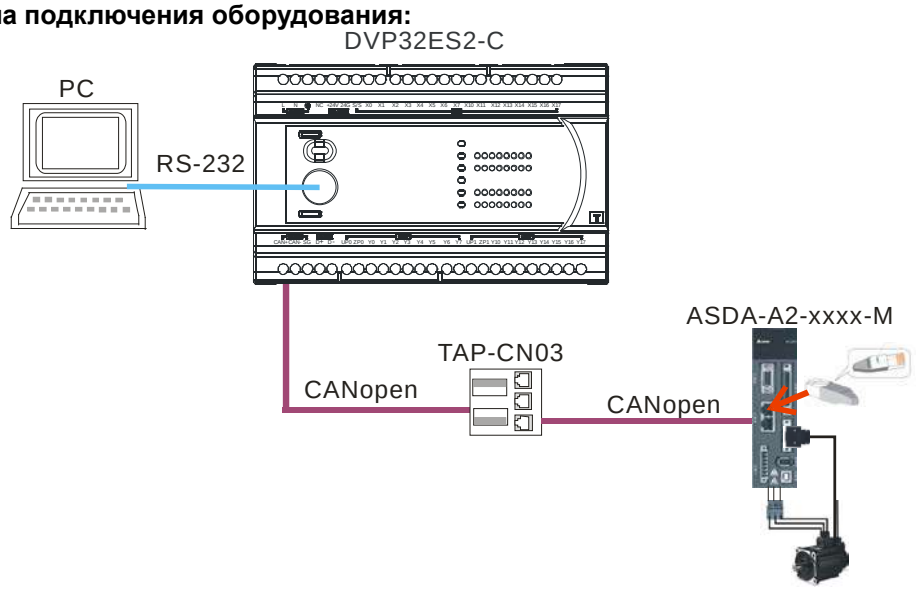
➢ Коды в D9980 когда DVP32ES2-C в Slave-режиме:

Код	Индикация	Метод устранения
A0	DVP-ES2-C инициализируется.	--
A1	DVP-ES2-C в предоперационном режиме.	Убедитесь, что кабели CANopen подключены правильно.
A3	Данные загружаются в DVP-ES2-C	Дождитесь конца загрузки.
B0	Истекло время ожидания сообщения Heartbeat	Убедитесь, что кабели CANopen подключены правильно.
B1	Длина данных PDO, возвращаемых от Slave-устройств не согласуется со значением, установленным в списке узлов.	Сбросьте длину данных PDO data length в Slave-устройстве и повторно загрузите ее в DVP-COPM-SL.
F4	Детектировано состояние отключения от шины.	Убедитесь, что кабели CANopen подключены правильно и убедитесь, что все узлы работают на одинаковой скорости и включите повторно.
FB	Буфер сообщений на отправку DVP-ES2-C заполнен.	Убедитесь, что кабели CANopen подключены правильно и включите повторно.
FC	Буфер принимаемых сообщений DVP-ES2-C заполнен.	Убедитесь, что кабели CANopen подключены правильно и включите повторно.

7.6 Пример применения

Контроллер DVP-ES2-C используется для управления сервоприводом Delta A2 и контроля фактической скорости вращения двигателя в режиме реального времени. Для решения данной задачи необходимо сконфигурировать сообщения PDO, включив в них требуемые регистры сервопривода, и осуществлять чтение/запись привода по шине CAN.

➢ **Схема подключения оборудования:**



Примечание:

1. Рекомендуем применять стандартный соединительный кабель TAP-CB03/04/05 и подсоединять стандартные терминальные резисторы Delta TAP-TR01 для построения сетей.
2. TAP-CN03 – разветвитель. Типы: TAP-CN01/CN02/CN03.
3. Буква «M» в названии привода ASD-A2-xxxx-M означает поддержку CANopen.

➢ **Настройка параметров сервопривода:**

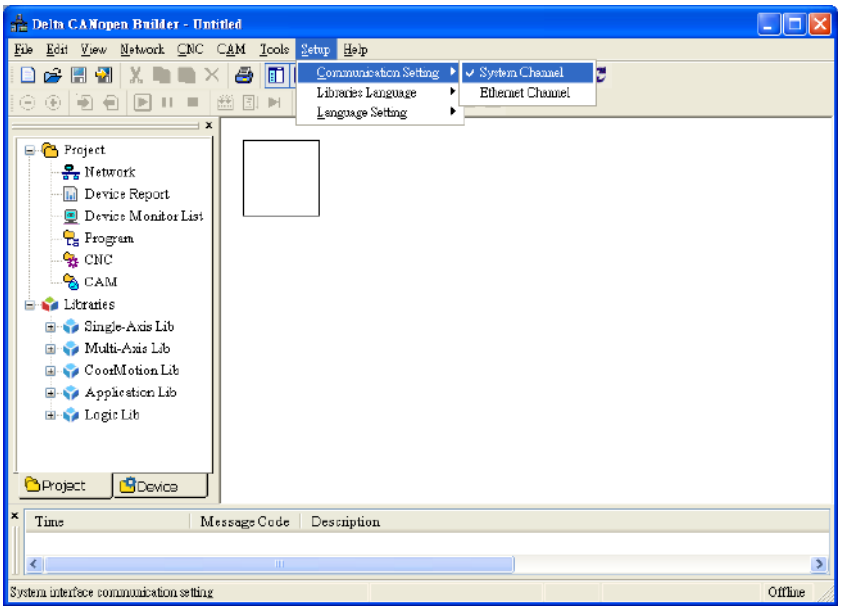
✧ Установка параметров:

Параметр	Установка	Описание
3-00	02	Сетевой адрес сервопривода A2 равен 2
3-01	400	Скорость связи по CAN равна 1 Мбит/с.
1-01	04	Режим скорости
0-17	07	Отображение скорости вращения двигателя (об/мин)
2-10	101	Установка DI1 как сигнал включения сервопривода
2-12	114	Установка DI3 и DI4 как сигналов выбора скорости

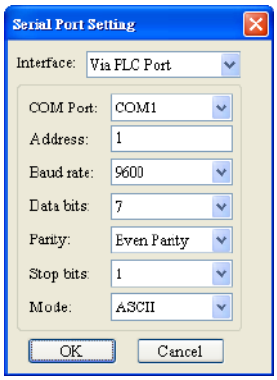
➢ **Настройка скорости связи CANopen и сетевого адреса для DVP-ES2-C**

DVP-ES2-C использует настройки по умолчанию: сетевой адрес: 17 и скорость: 1 Мбит/с. Эти параметры устанавливаются через ПО CANopen Builder. Пошаговое описание операции:

1) Откройте ПО CANopen Builder и войдите “Setup” > “Communication setting” > “System Channel”.

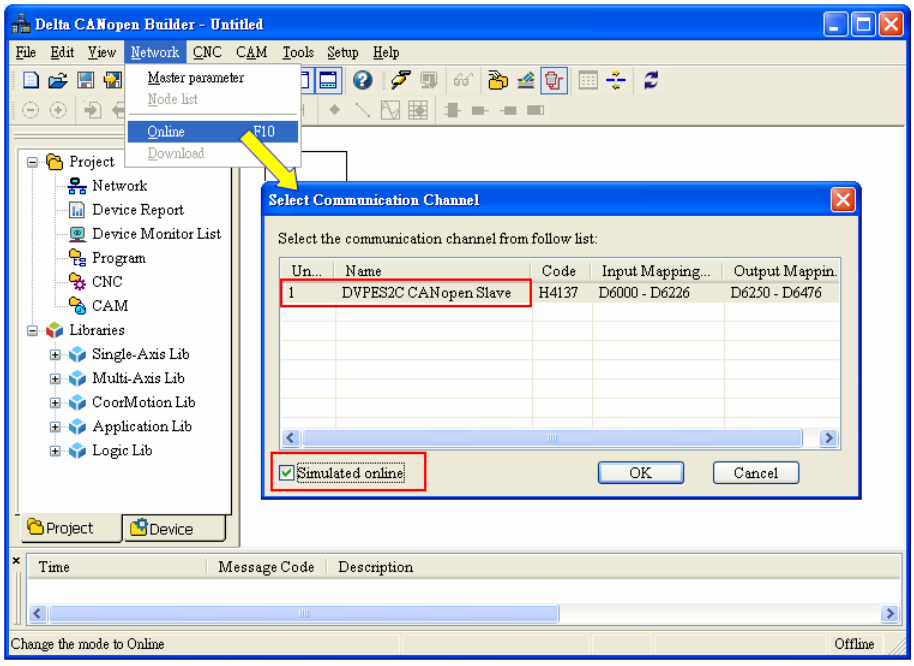


2) Появится следующее окно, где можно настроить параметры последовательного порта связи.



Пункт	Описание	По умолчанию
Интерфейс	Если оборудование, подключенное к ПК – это DVP10MC11T, выберите подключение через локальный порт; в других случаях – через ПЛК порт	--
COM порт	Номер COM-порта компьютера для связи DVP-ES2-C с ПК.	COM1
Адрес	Коммуникационный адресDVP-ES2-C	01
Скорость связи	Скорость связи между ПК и DVP-ES2-C	9600 бит/с
Биты данных	Протокол связи между ПК и DVP-ES2-C	7
Четность		Проверка четности
Стоповый бит		1
Режим	Режим связи между ПК и DVP-ES2-C	Режим ASCII

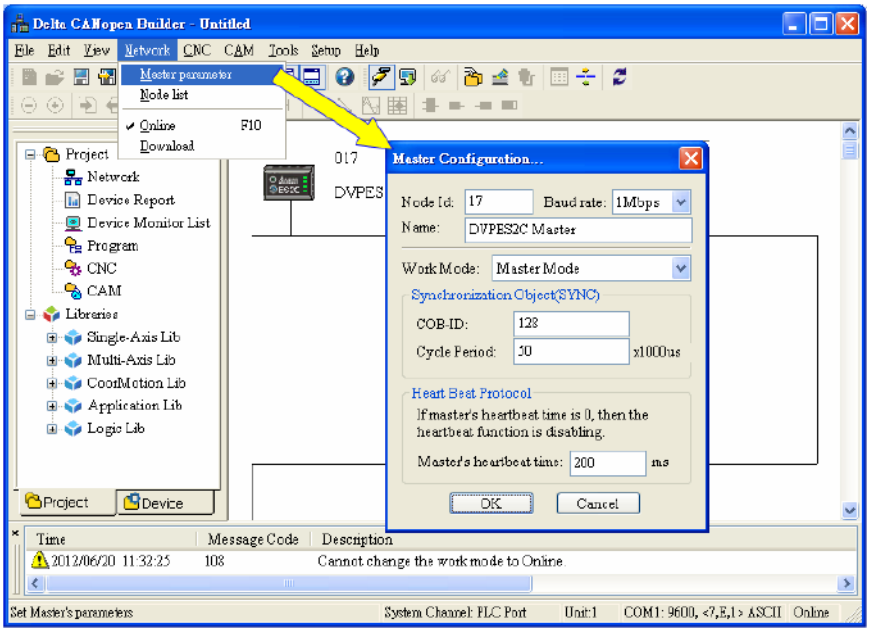
3) После завершения установок нажмите “Network”> “Online” и откроется окно “Select communication channel”.



1> Когда в столбце Name отображается “CANopen Slave”, это означает, что ПЛК находится в Slave-режиме CANopen. Выберите “Simulated online” в нижней левой части страницы и нажмите кнопку "OK", чтобы начать онлайн-сканирование.

2> Когда в столбце Name отображается “CANopen Master”, это означает, что ПЛК находится в Master-режиме CANopen. Нажмите “OK” чтобы начать онлайн-сканирование.

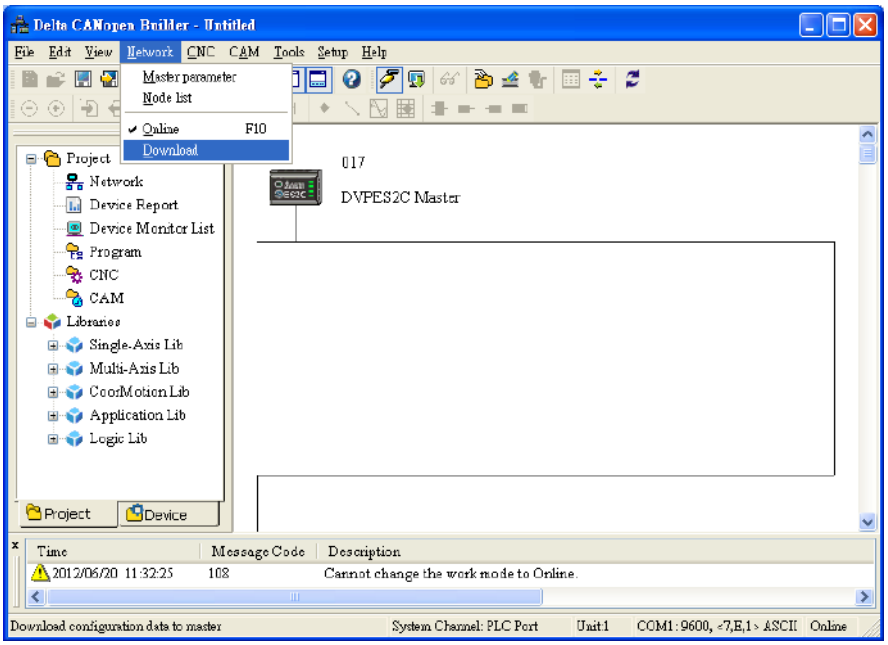
4) Нажмите “Network”> “Master Parameter”, появится окно “Master configuration…”



Пункт	Описание	По умолчанию
Сетевой адрес	Сетевой адрес DVP-ES2-C в сети CANopen	17
Скорость	Скорость связи CANopen	1 Мбит/с
Режим работы	Master/Slave режим CANopen	Master
Период цикла	Время цикла для передачи одного сообщения SYNC	50 мс
Время Heartbeat в Master-режиме	Время для передачи сообщения Heartbeat	200 мс

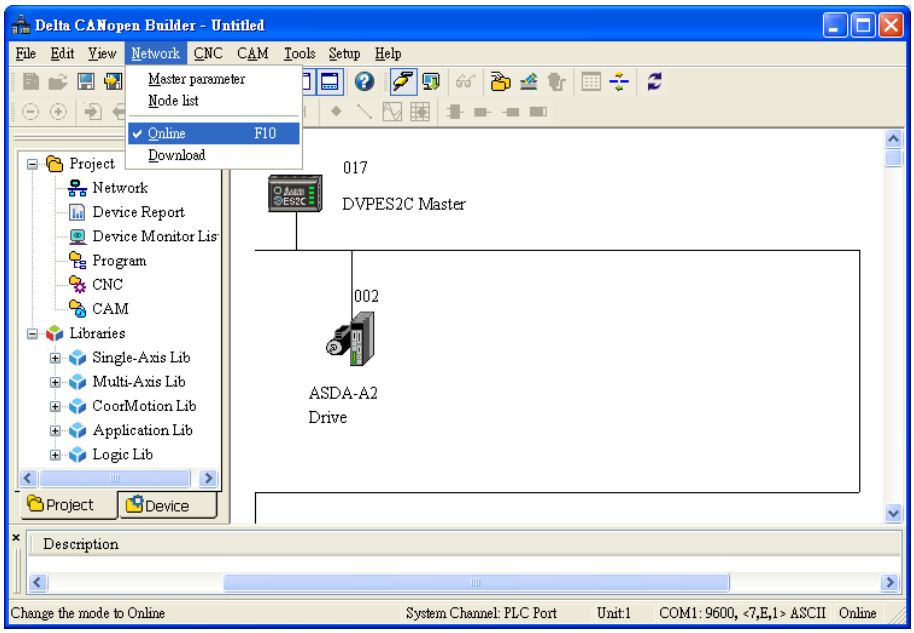
Пользователь может установить Сетевой адрес CANpen, скорость и режим Master/Slave для DVP-ES2-C.

5) После того, как описанные выше действия выполнены, осуществите загрузку как показано на рисунке ниже.



Примечание: Новые параметры вступят в силу после перезапуска DVP-ES2-C.

➢ **Сетевое сканирование:**
Сканирование Master и Slave-устройств в сети CANopen запускается нажатием “Network”>>”Online”. Процесс сканирования показан ниже. Для более детальной информации см. раздел 11.1.1 в файле справки ПО CANopen Builder.



➢ **Конфигурация узлов:**
Дважды кликните по иконке Slave-устройства в данном окне, и появится новое диалоговое окно “Node configuration”.

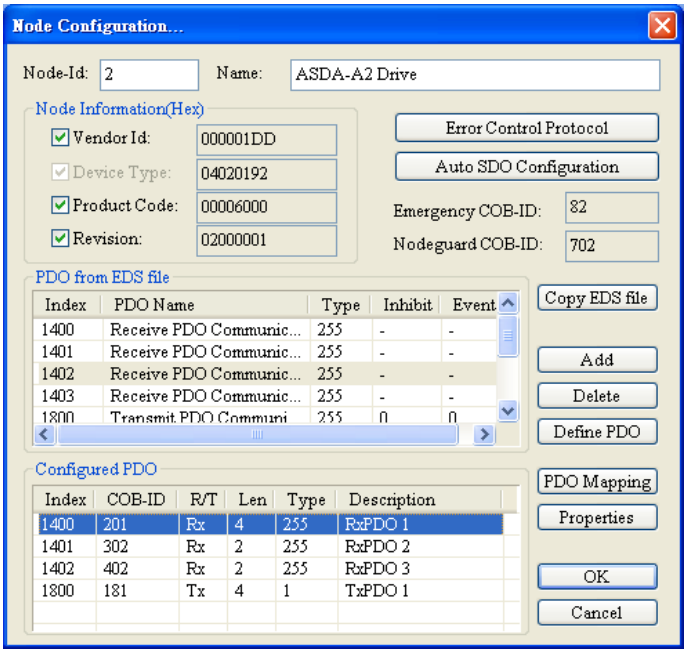
✧ “Error Control Protocol”
Используется для установки протокола контроля ошибок для Master-устройства для контроля Slave-устройств в офлайн.

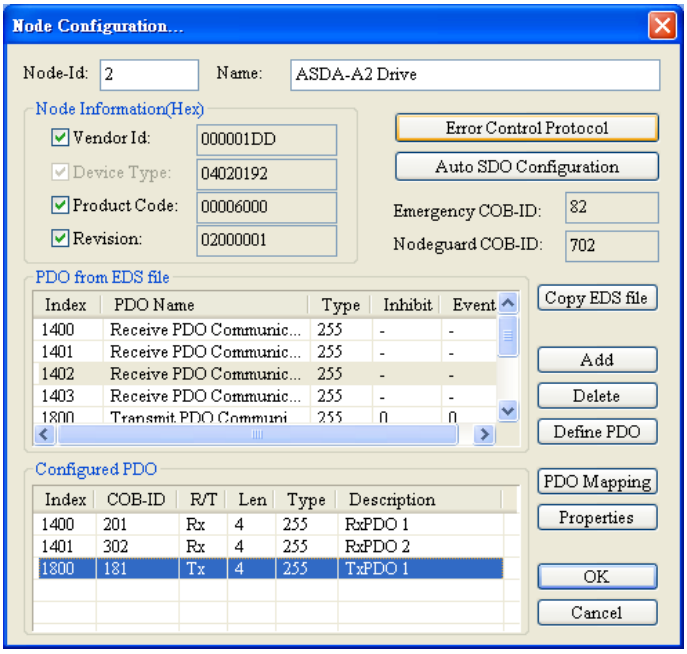
✧ “Auto SDO Configuration”
Используется для одной записи параметров в Slave-устройство с помощью SDO и завершения записи после выхода Slave-устройства в рабочий режим из режима предварительной работы. С помощью “Auto SDO configuration” можно сконфигурировать до 20 SDO.

✧ “PDO Mapping” и “Properties”
Используется для установки параметров отображения и передачи типа PDO.

✧ ″ Для получения подробной информации о функции кнопок, упомянутых выше см.

раздел 11.1.1 в файле справки ПО CANopen Builder.





- Отображение PDO:
  RxPDO1: отображение параметра P1-09; тип передачи 255.
  RxPDO2: отображение параметра P3-06, P4-07; тип передачи 255.
  TxPDO1: отображение параметра P0-09; тип передачи 1.
- Тип передачи PDO:
  PDO можно разделить на RxPDO и TxPDO. RxPDO: данные передаются от Master-устройства к Slave-устройству и TxPDO – наоборот.
  Тип передачи PDO может быть синхронным и асинхронным. При синхронной передаче Master будет посылать сообщение SYNC согласно фиксированному циклу. Длина цикла устанавливается в диалоговом окне свойств Master-устройства и по умолчанию имеет значение 50 мс. В асинхронном режиме сообщение PDO отправляется один раз, когда данные меняются.

Типы передачи PDO:

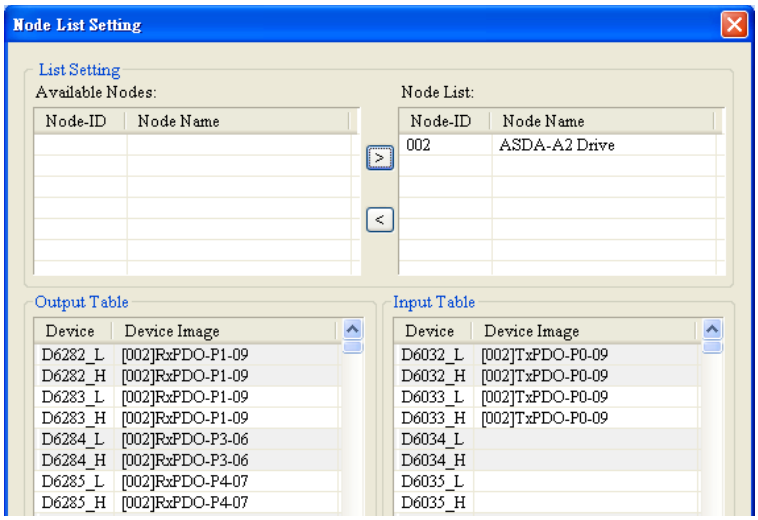
Тип передачи		Описание	Примечание
0	RxPDO	После любого изменения сопоставленных данных данные RxPDO отправляются немедленно. Данные, получаемые Slave-устройством, будут действительны только при получении следующего сообщения SYNCH. Если нет изменений данных RxPDO, они не рассылаются.	SYNCH SYNCH нецикличное
	TxPDO	После любого изменения сопоставленных данных и получении Slave-устройством сообщения SYNC данные отправляются немедленно. Данные TxPDO действительны сразу после получения их Master-устройством. Если нет изменений данных TxPDO, они не рассылаются.	
N （N:1~240）	RxPDO	После отправки N сообщений, и независимо от того, изменились ли отображенные данные, данные, которые получает Slave-устройство, будут действительны только при получении следующего сообщения SYNCH.	SYNCH циклическое
	TxPDO	После отправки N сообщений, и независимо от того, изменились ли отображенные данные, данные, которые получает Master-устройство, будут действительны сразу.	
254	RxPDO	Отображенные данные будут отправлены сразу после изменения и они действительны, как только будут получены Slave-устройством. Данные RxPDO не будет отправлены, если никаких изменений в данных.	ASYNCH
	TxPDO	Slave-устройство посылает данные один раз за одно время, установленное в таймере событий и после этого данные TxPDO не могут быть отправлены в течение времени блокировки таймера. Когда значения таймер и блокировка таймера оба равны 0, данные TxPDO отправляются сразу после изменения и данные, которые получает Master-устройство, будут действительны сразу.	
255	Аналогично Type254		

Примечание:

1> Синхронная передача позволяет осуществлять многокоординатное перемещение в одно и то же время.

2> Если необходимо отслеживать в реальном времени изменения параметров, таких, как фактическая скорость вращения двигателя, предлагается установить TxPDO в режиме синхронного типа передачи, иначе частое изменение данных Slave-устройства приводит к блокированию сети CANopen.

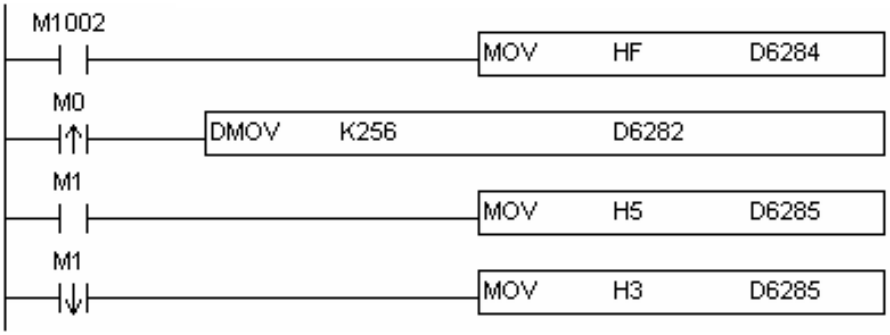
3> После завершения настройки указанных выше параметров дважды кликните по Master-устройству, выберите привод ASDA-A2 и кликните “>” для перемещения A2 в правый список и загрузки настроенных данных.



Отображение связи между Master и Slave-устройствами:

Регистры DVP-ES2-C (Master)	Передача данных по CANopen	Устройство A2
D6282		Младшее слово серво-привода P1-09
D6283		Старшее слово серво-привода P1-09
D6284	⇨	P3-06 сервопривода
D6285		P4-07 сервопривода
D6032		Младшее слово серво-привода P0-09
D6033	⇦	Старшее слово серво-привода P0-09

- Программное управление: регистру D6282 присваивается значение K256 при помощи ПО WPL. Это означает установку скорости как 256 об/мин. См. рисунок ниже.



Описание программы:

Когда DVP-ES2-C запускается первый раз, установите параметр сервопривода P3-06 как F.

- ✧ Когда M0 включается, величина K256 записывается в D6282, затем записывается значение в P1-09 (параметры сервопривода) через RxPDO1.
- ✧ Когда M1 включается, включается P2-12 и передает значение скорости из параметра P1-09 сервопривода для осуществления вращения.
- ✧ Когда M1 выключается, команда указания скорости становится 0 и мотор останавливается.

7.7 Словарь объектов

Объекты коммуникации:

Unsigned – целое число без знака

Индекс	Субиндекс	Наименование	Тип данных	Атрибут	По умолчанию
H'1000	H'00	Тип устройства	Unsigned 32 бит	R	0x00000000
H'1001	H'00	Ошибка регистра	Unsigned 8 бит	R	0
H'1005	H'00	COB-ID SYNC	Unsigned 32 бит	RW	0x00000080
H'1008	H'00	Название устройства	Vis-String (строчный)	R	DVPES2C
H'1014	H'00	COB-ID EMCY	Unsigned 32 бит	R	0x80 + Node-ID
H'1016	--	Время Heartbeat потребителя			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	1
	H'01	Время Heartbeat потребителя	Unsigned 32 бит	RW	0
H'1017	H'00	Время Heartbeat производителя	Unsigned 16 бит	RW	0
H'1018	--	Идентичность объекта			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	Вендор-ID	Unsigned 32 бит	R	0x000001DD
	H'02	Код продукта	Unsigned 32 бит	R	0x00000055
	H'03	Номер версии	Unsigned 32 бит	R	0x00010002

Индекс	Субиндекс	Наименование	Тип данных	Атрибут	По умолчанию
H'1400	--	RxPDO1 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO1	Unsigned 32 бит	RW	0x00000200+Node-ID
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1401	--	RxPDO2 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO2	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1402	--	RxPDO3 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO3	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1403	--	RxPDO4 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO4	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1404	--	RxPDO5 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO5	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1405	--	RxPDO6 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO6	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1406	--	RxPDO7 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO7	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0
H'1407	--	RxPDO8 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	3
	H'01	COB-ID для RxPDO8	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокировки	Unsigned 16 бит	RW	0

Индекс	Субиндекс	Наименование	Тип данных	Атрибут	По умолчанию
H'1600	--	RxPDO1 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	4
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0x20000110
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0x20000210
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0x20000310
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0x20000410
H'1601	--	RxPDO2 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1602	--	RxPDO3 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1603	--	RxPDO4 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1604	--	RxPDO5 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1605	--	RxPDO6 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0

Индекс	Субиндекс	Наименование	Тип данных	Атрибут	По умолчанию
H'1606	--	RxPDO7 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1607	--	RxPDO8 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'01	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1800	--	TxPDO1 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO1	Unsigned 32 бит	RW	0x00000180+Node-ID
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1801	--	TxPDO2 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO2	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1802	--	TxPDO3 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO3	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1803	--	TxPDO4 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO4	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1804	--	TxPDO5 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO5	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100

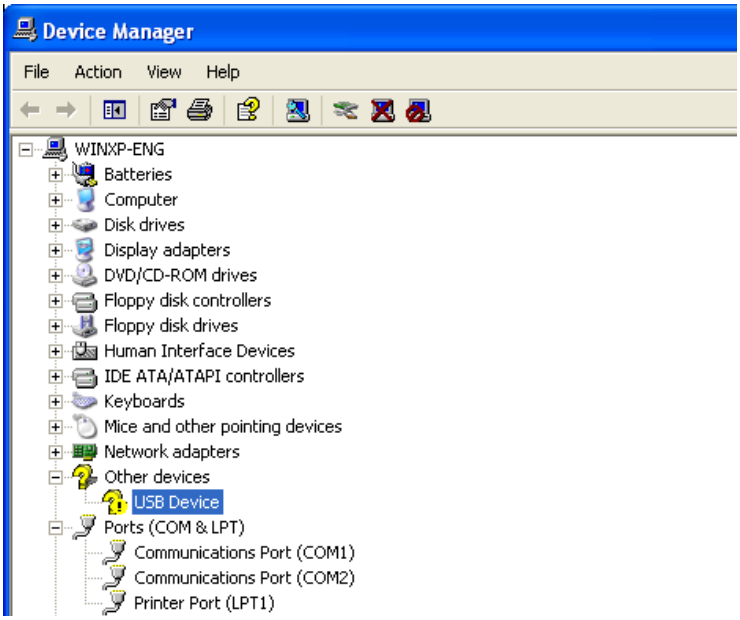
Индекс	Субиндекс	Наименование	Тип данных	Атрибут	По умолчанию
H'1805	--	TxPDO6 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO6	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1806	--	TxPDO7 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO7	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1807	--	TxPDO8 параметр связи			
	H'00	Число действительных субиндексов	Unsigned 8 бит	R	5
	H'01	COB-ID для TxPDO8	Unsigned 32 бит	RW	0x80000000
	H'02	Режим передачи	Unsigned 8 бит	RW	0xFF
	H'03	Время блокирования	Unsigned 16 бит	RW	50
	H'05	Таймер	Unsigned 16 бит	RW	100
H'1A00	--	TxPDO1 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	4
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0x20010110
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0x20010210
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0x20010310
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0x20010410
H'1A01	--	TxPDO2 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1A02	--	TxPDO3 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1A03	--	TxPDO4 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0

Индекс	Субиндекс	Наименование	Тип данных	Атрибут	По умолчанию
H'1A04	--	TxPDO5 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1A05	--	TxPDO6 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1A06	--	TxPDO7 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0
H'1A07	--	TxPDO8 параметр отображения			
	H'00	Число действительных субиндексов	Unsigned 8 бит	RW	0
	H'01	Первый отображаемый объект	Unsigned 32 бит	RW	0
	H'02	Второй отображаемый объект	Unsigned 32 бит	RW	0
	H'03	Третий отображаемый объект	Unsigned 32 бит	RW	0
	H'04	Четвертый отобража-емый объект	Unsigned 32 бит	RW	0

Установка драйвера USB для работы ПЛК Delta с компьютером

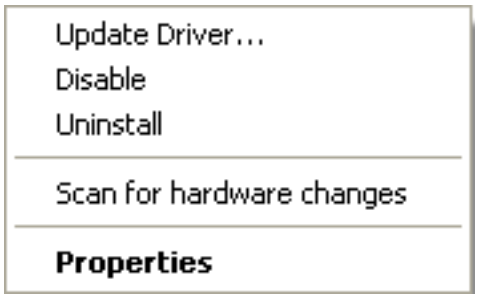
Установка драйвера.
После установки драйвера порт USB может использоваться как последовательный порт (RS-232). Вы можете применять стандартный USB кабель длиной до 5 м.

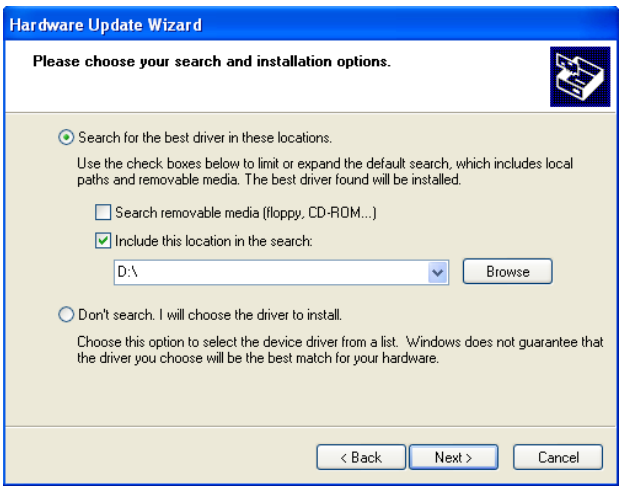
ПК и ПЛК соединяются с помощью USB или миниUSB кабеля. После соединения устройства могут быть найдены в разделе **USB Device (USB устройства)** в стандартном окне **Device Manager (Диспетчер устройств)**.



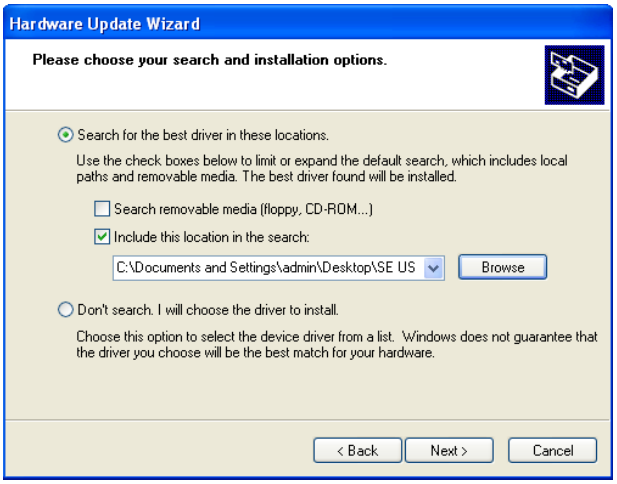
A

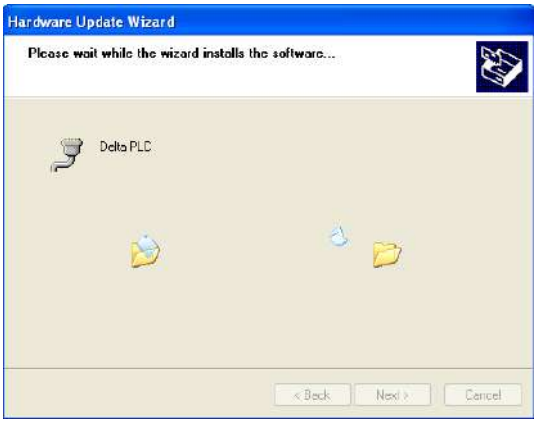
Правой кнопкой мыши выберите **Update Driver (Обновить драйвер)** для открытия окна **Hardware Update Wizard (Мастер обновления оборудования).**
Нажмите **Browse (Обзор)** для выбора папки и **Next (Далее)** для запуска установки драйвера.





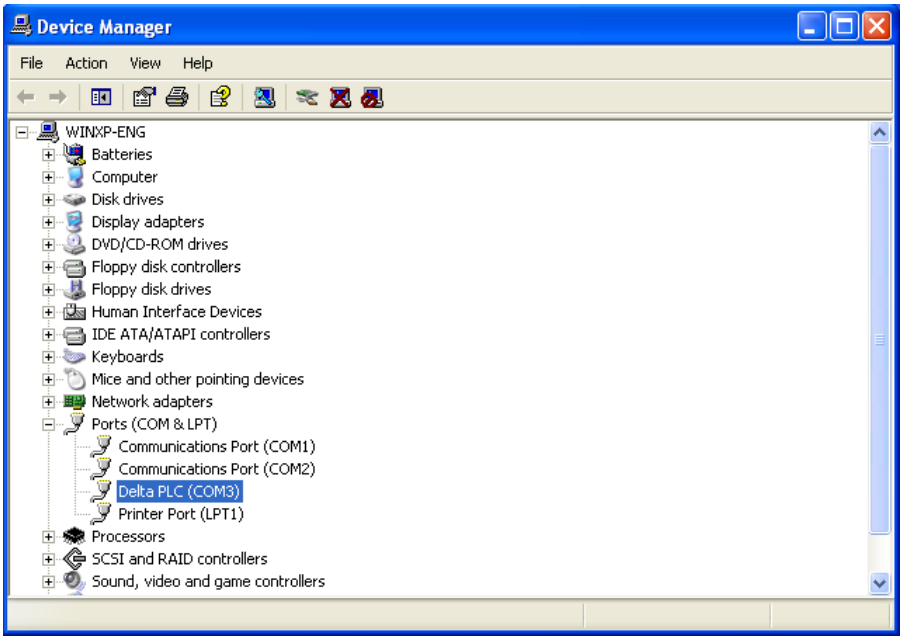




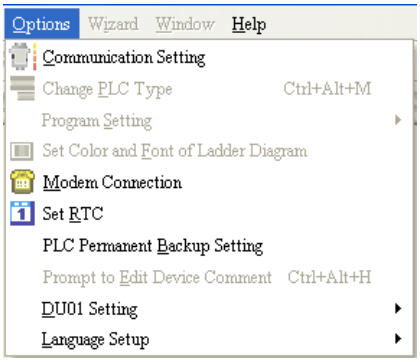


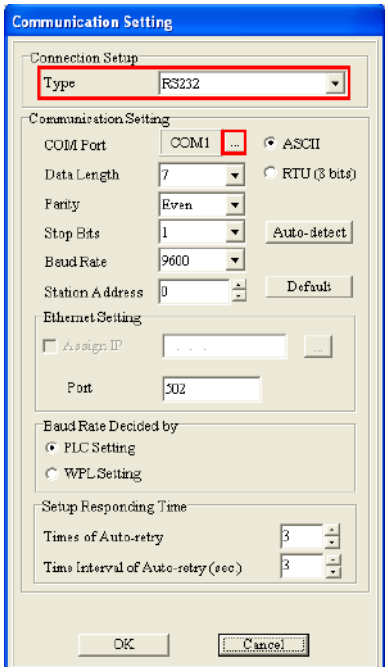


После установки драйера устройство **Delta PLC** и соответствующий коммуникационный порт будут отображены в окне **Device Manger (Диспетчер устройств)**. Работа с этим портом аналогична работе с портом RS-232.

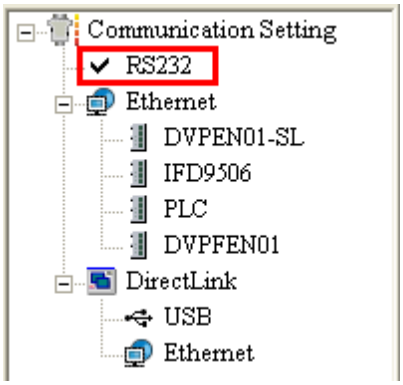


Выберите **Communication Setting (Настройки связи)** в **Options (Опции)**. Выберите **RS232** в **Connection Setup (Настройки подключения)**, выберите присвоенный USB порт в **Communication Setting** и нажмите **OK**. Порт можно применять аналогично порту **RS232** для загрузки программ в ПЛК или для работы онлайн.









КРАТКИЙ ОБЗОР ПРОДУКЦИИ DELTA

Линейка преобразователей частоты Delta VFD

позволяет потребителю выбрать оптимальную модель для любой задачи, требующей применения частотно-регулируемого электропривода и управления асинхронными двигателями.





Панели оператора серий DOP и TP

- это современные программируемые графические терминалы, предназначенные для визуализации информации, поступающей от микропроцессорных устройств (промышленных контроллеров, преобразователей частоты, регуляторов температуры и т.д.), и управления производственными и технологическими процессами. Использование операторских панелей в системах управления позволяет создать удобный человеко-машинный интерфейс (HMI) - централизованное рабочее место человека-оператора, управляющего машиной.

Контрольно-измерительные приборы Delta

представлены линейкой регуляторов температуры DT, предназначенных для управления процессом поддержания таких технологических параметров, как температура, давление, расход и др., а так же комбинированным таймером / счетчиком / тахометром серии CTA, сочетающий широкие функциональные возможности с компактным корпусом и невысокой стоимостью.



Сервоприводы серии ASD

От простых задач позиционирования, управления скоростью и моментом до сложных, требующих специальных функций, таких как дополнительный контур управления, интерполяция, E-CAM, работа с G-кодами и т.п.



Перевод и адаптация выполнены компанией «СТОИК» - авторизованным дистрибьютором средств промышленной автоматизации Delta Electronics в России.