

Программируемый логический контроллер

LogicOn

**Руководство
по программированию**

ВВЕДЕНИЕ	8
1 ТЕХНИКА БЕЗОПАСНОСТИ	9
2 ОБЩАЯ ЧАСТЬ	10
2.1 Принцип работы ПЛК.....	10
2.2 Основы языка LD.....	12
2.3 Быстрый старт.....	16
2.2.1 Создание нового проекта.....	16
2.2.2 Добавление элементов на рабочее поле.....	17
2.2.3 Пример простого проекта.....	19
2.2.4 Сохранение проекта.....	19
2.2.5 Компиляция.....	20
2.2.6 Выбор интерфейса для связи с контроллером.....	20
2.2.7 Настройка связи по USB.....	20
2.2.8 Настройка связи по RS-232/RS-485.....	21
2.2.9 Настройка связи по Ethernet.....	22
2.2.10 Загрузка программы.....	22
2.2.11 Мониторинг исполнения программы.....	23
3 ПРОГРАММНЫЕ ЭЛЕМЕНТЫ ПЛК	25
3.1 Входные реле X.....	27
3.2 Выходные реле Y.....	27
3.3 Аналоговые входы AI.....	29
3.4 Аналоговые выходы AQ.....	30
3.5 Внутренние реле M.....	31
3.6 Специальные (системные) внутренние реле SM.....	32
3.7 Регистры данных D.....	40
3.8 Специальные регистры SD.....	42
3.9 Шаговые реле S.....	50
3.10 Таймеры T.....	51
3.11 Счётчики C.....	54
3.11 Высокоскоростные счетчики.....	58
3.12 Указатели L, P, I.....	63
3.13 Константы K, H, F.....	67
3.13 Индексные регистры V, Z.....	69
4 ПРИКЛАДНЫЕ ИНСТРУКЦИИ	71
4.1 Базовые инструкции (Basic Inst).....	72
[LD].....	73
[LDI].....	74
[OUT].....	75

[LDP].....	76
[LDF].....	77
[MC], [MCR].....	78
[NOT].....	79
[PLS].....	80
[PLF].....	81
[SET].....	82
[RST].....	83
[NOP].....	84
[OUTD].....	84
[ALT].....	85
4.2 Инструкции пошагового управления (Step Inst).....	88
[STL].....	92
[RET].....	93
4.3 Инструкции управления процессом выполнения программы (Program Process Inst)..	94
[LBL].....	95
[CJ].....	96
[CALL / CALLP].....	97
[WDT / WDTP].....	98
[EI].....	99
[DI].....	99
[FOR].....	101
[NEXT].....	103
4.4 Таймеры и счетчики (Timer Inst / Counter Inst).....	104
[TMR].....	105
[CNT].....	106
4.5 Инструкции сравнения (Compare Inst).....	107
[LD= / LDD=].....	108
[LD<> / LDD<>].....	108
[LD> / LDD>].....	109
[LD< / LDD<].....	109
[LD>= / LDD>=].....	110
[LD<= / LDD<=].....	110
[LDR=].....	111
[LDR<>].....	111
[LDR>].....	112
[LDR<].....	112
[LDR>=].....	113
[LDR<=].....	113
4.6 Математические инструкции (Math Inst).....	114
[ADD / DADD].....	115
[SUB / DSUB].....	116

[MUL / DMUL].....	117
[DIV / DDIV].....	118
[INC / DINC].....	119
[DEC / DDEC].....	120
[WAND / DAND].....	121
[WOR / DOR].....	122
[WXOR / DXOR].....	122
[NEG / DNEG].....	124
4.7 Инструкции сравнения и передачи данных (Transfer and Compare Inst).....	125
[CMP / DCMP].....	126
[ZCP / DCPP].....	127
[MOV / DMOV].....	128
[CML / DCML].....	129
[BMOV].....	130
[FMOV / DFMOV].....	131
[XCH / DXCH].....	132
[BCD / DBCD].....	133
[BIN / DBIN].....	134
[DEMOV / DEMOVP].....	135
4.8 Инструкции сдвига (Shift Inst).....	136
[ROR / DROR].....	137
[ROL / DROL].....	138
[RCR / DRCR].....	140
[RCL / DRCL].....	141
[SFTR].....	143
[SFTL].....	144
[WSFR].....	145
[WSFL].....	147
[SFWR].....	149
[SFRD].....	150
4.9 Инструкции обработки данных (Data Process Inst).....	151
[ZRST].....	152
[ENCO].....	153
[SUM / DSUM].....	155
[BON / DBON].....	156
[MEAN / DMEAN].....	157
[ANS].....	158
[ANR].....	159
[SQR / DSQR].....	160
[FLT / DFLT].....	161
[SWAP / DSWAP].....	162
[SCALE / DSCALE].....	163
4.10 Инструкции для чисел с плавающей точкой (Float Process Inst).....	165
[DECMP].....	166

[DEZCP].....	168
[DEBCD].....	169
[DEBIN].....	170
[DEADD].....	171
[DESUB].....	172
[DEMUL].....	173
[DEDIV].....	174
[DESOR].....	175
[INT / DINT].....	176
[DESIN].....	177
[DECOS].....	178
[DETAIL].....	179
[DEPOW].....	180
4.11 Высокоскоростные инструкции (High-speed inst).....	181
[REF].....	182
[REFF].....	183
[SPD].....	184
[PLSY / DPLSY].....	185
[PWM].....	186
[PLSR / DPLSR].....	187
[PLSV / DPLSV].....	188
[PWMR].....	189
[PWMY / DPWMY].....	190
[ZRN / DZRN].....	192
[DHSCT].....	194
[DSZR].....	196
[DRVI / DDRVI].....	201
[DRVA / DDRVA].....	203
[PLSTR / DPLSTR].....	205
[PSTOP].....	207
4.12 Инструкции связи с внешними устройствами (External Device Inst).....	208
[MBUSR].....	209
[MBUSWB].....	211
[MBUSRW].....	213
[MBUSWW].....	215
[PIDT].....	217
[PIDR].....	220
5 ТИПОВЫЕ ЗАДАЧИ.....	222
5.1 Подключение кнопки и дребезг контактов.....	222
5.2 Управление освещением. Функционал шагового (импульсного) реле.....	223
5.3 Подключение модулей расширения.....	224
5.4 Преобразование аналогового сигнала в пользовательский вид.....	226
5.5 ПЛК в режиме Slave-устройства Modbus.....	227

5.6 ПЛК в режиме Master-устройства Modbus.....	230
5.7 Получение и установка битовой маски.....	233
5.8 Энергонезависимая память.....	235
5.9 Часы реального времени (RTC).....	236
5.10 Работа с энкодером.....	237
ПРИЛОЖЕНИЕ 1 - Программное обеспечение.....	238
Описание интерфейса программы.....	238
Главное меню (Menu Bar).....	238
Панели инструментов (Tool Bar).....	239
Главная панель инструментов (Main Tool Bar).....	239
Панель инструментов редактирования программ (Program Editing Tools).....	240
Панель инструментов редактирования программ на языке LD (Ladder node Tools)....	241
Панель инструментов управления ПЛК (PLC Tools).....	241
Панель инструментов редактирования кода(Code Editing Tools).....	242
Рабочее поле и окно управления проектом.....	244
Окно редактирования основной программы(Main program).....	245
Окно редактирования подпрограммы(Sub Programm).....	246
Окно редактирования пользовательских блоков (Custom instruction Library).....	246
Таблицы символов (Symbol Table).....	247
Системные настройки (System Setting).....	248
Таблица программных элементов(Software use table).....	253
Библиотека инструкций (Instruction Library).....	254
Окно таблицы мониторинга (Monitoring Table Window).....	255
Окно сообщений (General Output).....	256
Строка состояния(Status Bar).....	256
ПРИЛОЖЕНИЕ 2 - Программа «Светофор».....	257
Запуск пошагового управления.....	257
Начало пошагового управления (горит красный свет).....	258
Приготовиться к движению (одновременно горят красный и желтый).....	258
Движение разрешено (горит зеленый свет).....	259
Проверка нажатия кнопки (выбор следующего шага).....	259
Предупреждение о переключении сигнала «Движение разрешено» (зелёный мигает)....	261
Окончание сигнала «Движение разрешено» (горит желтый).....	261
Итоговая программа.....	262
ПРИЛОЖЕНИЕ 3 - Создание макросов на языке С.....	264
Описание переменных, пример макроса для математического вычисления.....	264
Переменные со спецификатором static, работа с системным временем и внутренними регистрами ПЛК.....	268

ВВЕДЕНИЕ

Настоящее руководство по программированию (далее – РП) предназначено для ознакомления с принципом работы, техническими характеристиками, конструктивными особенностями, условиями эксплуатации, порядком работы и техническим обслуживанием Программируемого логического контроллера серии LogicOn (далее – ПЛК).



Данный документ предназначен для технического, обслуживающего и эксплуатирующего персонала, а также специалистов, осуществляющих проектирование систем управления с применением данного ПЛК.

Установка, подключение и обслуживание ПЛК должны производиться только квалифицированным персоналом, обладающим навыками и знаниями по работе с электрооборудованием и изучившим данное РП. Невыполнение требований, изложенных в настоящей документации, и нарушение условий эксплуатации может привести к непредвиденным авариям, вплоть до выхода из строя оборудования, а также снятию гарантийных обязательств Поставщика.

Сохраните данное РП для последующего технического обслуживания, осмотра и настройки ПЛК.

Если у Вас возникли вопросы в ходе изучения РП, пожалуйста, свяжитесь с технической поддержкой для получения квалифицированной консультации.

В настоящем РП приняты следующие условные обозначения и сокращения:

	Несоблюдение требований или неправильное обращение может привести к опасным ситуациям для персонала или вызвать повреждения материального имущества
	Примечания, на которые следует обратить внимание
РП	Руководство по программированию
ПЛК	Программируемый логический контроллер






1 ТЕХНИКА БЕЗОПАСНОСТИ

Не приступайте к установке, эксплуатации, техническому обслуживанию или утилизации ПЛК до тех пор, пока не изучите информацию, описанную в данном РП.

К проведению работ по монтажу или демонтажу, наладке, подключению и техническому обслуживанию допускается только квалифицированный персонал . Квалифицированным считается специалист который:

- Обладает необходимой квалификацией и компетенцией для выполнения данного вида работ.
- Имеет допуск к проведению работ на электроустановках с напряжением до 1000 В.
- Прошел инструктаж по технике безопасности.

Заказчик ПЛК несет ответственность за компетенцию допускаемого к работам персонала и должен самостоятельно организовать наблюдение за персоналом . Если персонал не обладает достаточными знаниями, он должен быть обучен.

	Запрещается проводить монтажные работы, коммутацию внешних проводок, а также производить какие либо подключения к ПЛК и дотрагиваться до его токоведущих частей при включенном напряжении питания.
	Запрещается прикасаться к оборудованию внутри ПЛК и монтажной панели влажными руками во избежание поражения электрическим током.
	Запрещается самостоятельно разбирать, модифицировать или ремонтировать ПЛК. Это может привести к выходу из строя оборудования а также снятию гарантийных обязательств Поставщика По вопросам связанным с ремонтом, необходимо обращаться к Поставщику.
	Запрещается эксплуатировать ПЛК в условиях, не соответствующих изложенным в данном РП требованиям.
	Необходимо предотвратить доступ посторонних лиц к ПЛК.

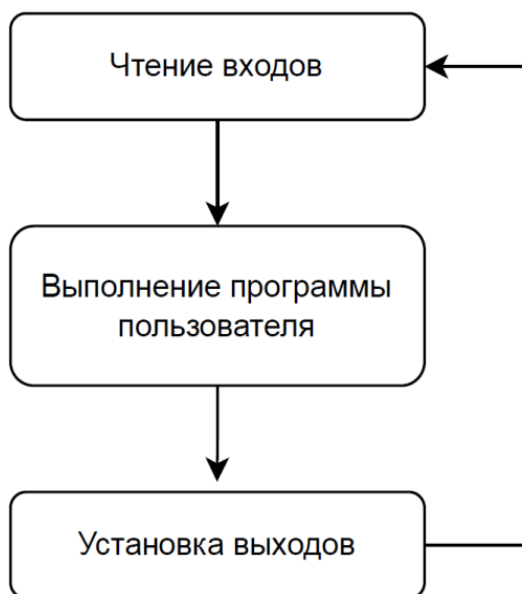
2 ОБЩАЯ ЧАСТЬ

2.1 Принцип работы ПЛК

Процесс работы ПЛК можно условно разделить на основных 3 этапа:

1. Чтение собственных входов;
2. Вычисление значений выходных и внутренних переменных на основании программы пользователя;
3. Установка собственных выходов;

Совокупность всех трех этапов составляет рабочий цикл ПЛК.



Рабочий цикл ПЛК

Циклическое выполнение программы отличает работу ПЛК от обычных релейно-контактных схем, где все управляющие процессы происходят одновременно и каждое изменение входных сигналов сразу же приводит к изменению состояния выходных сигналов. Тем не менее в ПЛК предусмотрена возможность «мгновенного» обновления состояния входов и выходов, независимо от текущего этапа цикла. Для этого используются инструкции описанные в разделе [4.11 Высокоскоростные инструкции \(High-speed inst\)](#).

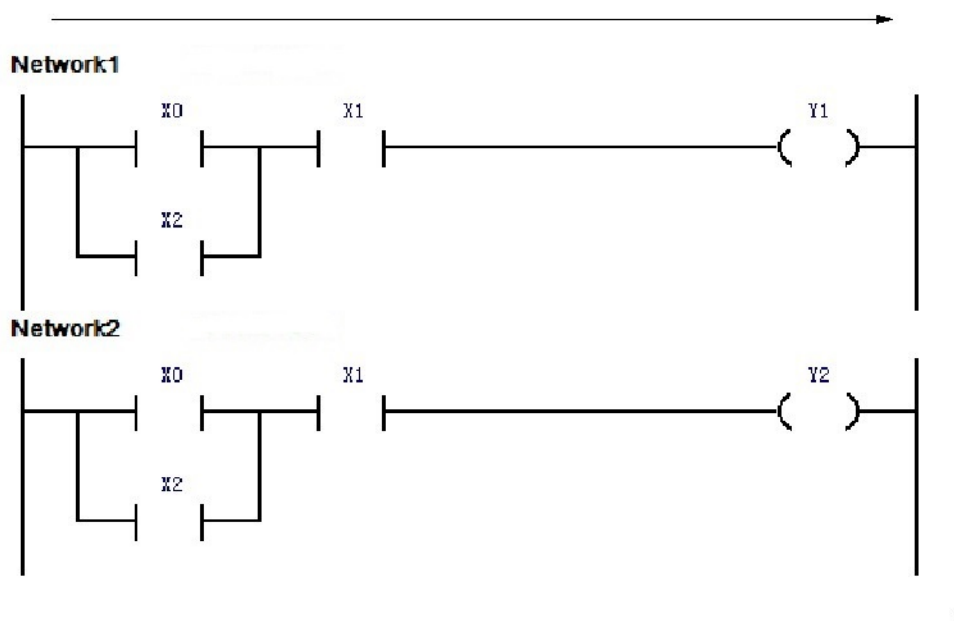
Время, которое контроллер затрачивает на выполнение программы пользователя называется временем цикла или временем сканирования. В зависимости от сложности программы время сканирования будет разным: чем сложнее программа, тем время будет больше. При необходимости можно установить фиксированное время сканирования с помощью специальных реле и регистров (подробнее в разделе [3 ПРОГРАММНЫЕ ЭЛЕМЕНТЫ ПЛК](#)).

В ПЛК заложен специальный механизм защиты от слишком долгого времени выполнения программы - сторожевой таймер. При превышении определенного времени цикла (по умолчанию 200 миллисекунд), контроллер прекращает выполнение программы, выдает сообщение об ошибке и переводит все выходы в выключенное состояние. Время срабатывания сторожевого таймера можно изменить, задав нужное значение в специальном регистре SD0, а также производить периодический сброс накопленного значения инструкцией WDT.

2.2 Основы языка LD

Среда разработки поддерживает язык лестничного программирования (лестничных диаграмм, релейной логики, LD). Также имеется возможность создания пользовательских блоков на языке С.

Метод лестничного программирования в ПЛК является одним из методов проектирования электрической схемы в соответствии с релейной системой управления. Компоненты, используемые в LD такие как кнопка X, вспомогательное реле М, таймер Т, счетчик С и т.д. аналогичны реальным электрическим компонентам.

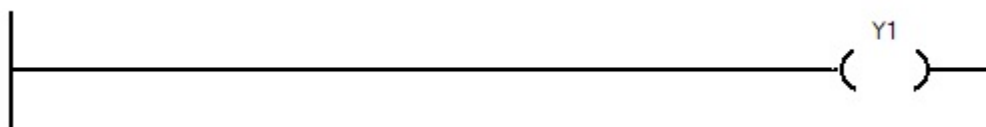


Пример программы на языке LD

В примере на рисунке выше программа выполняется шаг за шагом, слева направо и сверху вниз. «Network» означает группу элементов, находящихся на одной линии. На приведенной выше схеме таких групп две.

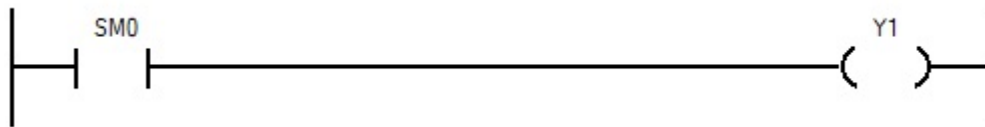
При разработке программ на языке LD необходимо учитывать тот момент, что хотя и используются те же элементы что и в реальных электрических схемах, сам принцип работы ПЛК добавляет некоторые особенности в принципе подключения и использования элементов.

На примере ниже в реальной электрической схеме выход Y1 будет всегда включен, так как подключен к главной питающей шине.



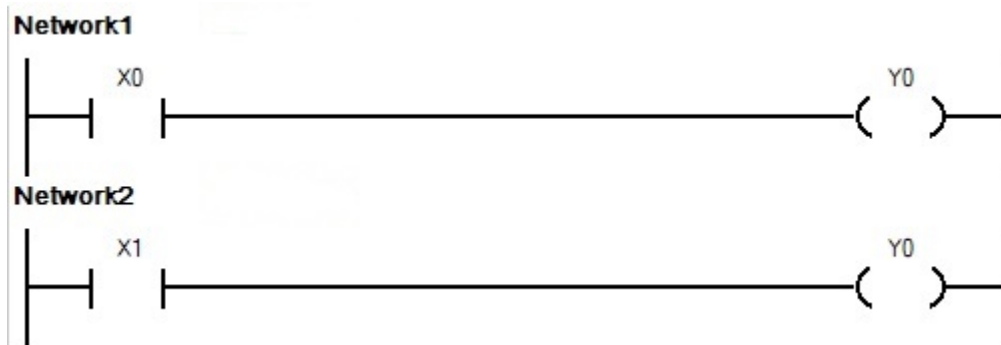
Выход подключен напрямую к левой шине

Но в программе ПЛК так делать нельзя - выходной элемент или инструкция всегда должны подключаться к главной шине через какой-либо контакт¹ и при попытке скомпилировать такую программу среда разработки выдаст ошибку. Если необходимо сделать так, чтобы выход был включен всегда, пока ПЛК находится в режиме «RUN», следует использовать специальное реле SM0 (AlwaysOn) и подключить выход как указано на рисунке ниже. Реле SM0 - это специальное реле, которое включено всегда.



Выход подключен к левой шине через контакт SM0

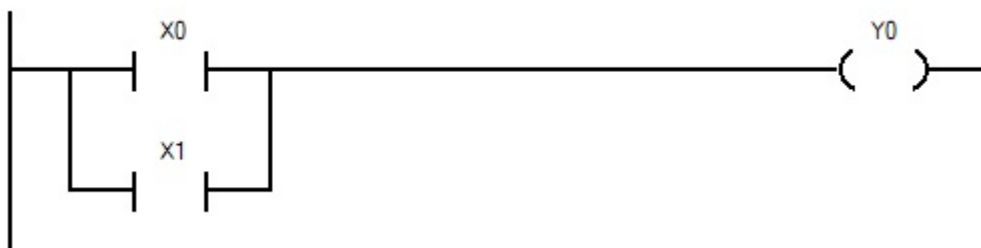
Также необходимо учитывать, что если использовать несколько выходных элементов с одним и тем же идентификатором в разных частях программы, то действительным условием для определения состояния выхода будет только самое последнее из этих условий. Предыдущие условия не окажут никакого влияния на состояние выхода. На примере ниже, в реальной электрической схеме выход Y0 включался бы всегда, когда включен или X0 или X1.



Проблема двойного использования выхода

Но т.к. программа ПЛК выполняется последовательно сверху вниз, а фактическая установка выходов происходит только в самом конце цикла программы, то состояние выхода Y0 будет определяться только входом X1, при этом состояние входа X0 никакого влияния на Y0 не окажет. Правильным решением в таком случае будет использовать один выход и параллельное соединение входов.

¹ некоторые прикладные инструкции не требуют наличия дополнительного контакта между главной левой шиной и самой инструкцией. К ним относятся инструкции управления циклом FOR, NEXT, разрешение и запрет прерываний EI, DI, метки LBL



Использование одного выхода и двух входов, соединенных параллельно

Стоит отметить, что проблема двойного использования характерна не только для физических выходов, но и для многих инструкций, например инструкций связи с внешними устройствами Modbus.

Основные символы языка LD:

Элемент диаграммы	Описание	Команда	Доступные элементы
	Нормально открытый контакт	LD	X, Y, M, SM, S, T, C
	Нормально закрытый контакт	LDI	X, Y, M, SM, S, T, C
	Детектор переднего фронта импульса	LDP	X, Y, M, SM, S, T, C
	Детектор заднего фронта импульса	LDF	X, Y, M, SM, S, T, C
	Выходная катушка	OUT	Y, M, SM, S, T, C
	Шаговое реле	STL	S

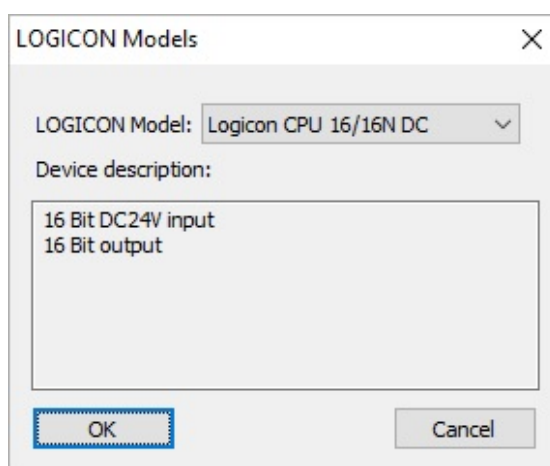
Элемент диаграммы	Описание	Команда	Доступные элементы
	Логическая инверсия	INV	нет

2.3 Быстрый старт

2.2.1 Создание нового проекта

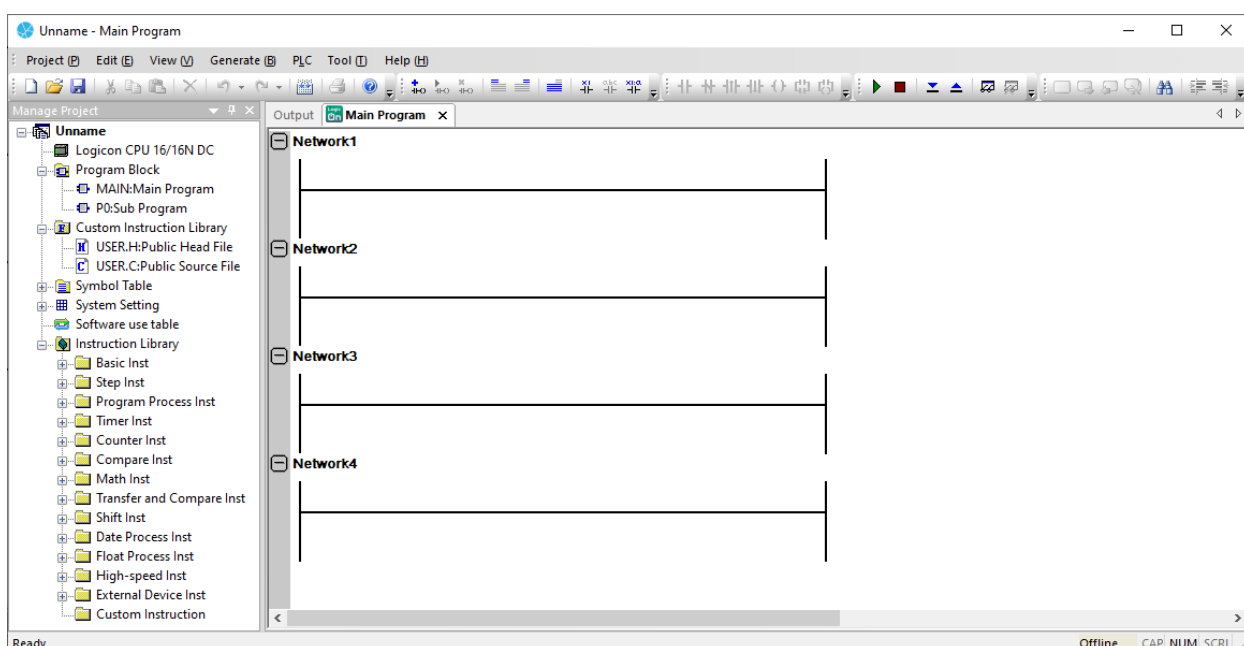
Для создания нового проекта необходимо в главном меню или на главной панели инструментов ELHART LogicOn Soft выбрать «Project» > «New Project». Затем выбрать модель ПЛК для которого будет создаваться программа и нажать «ОК».

При выборе модели ПЛК нужно руководствоваться наименованием и количеством точек ввода-вывода, при этом тип выхода и напряжения питания не имеют значения.



Выбор модели ПЛК

После этого будет доступен основной интерфейс программы и рабочее поле.

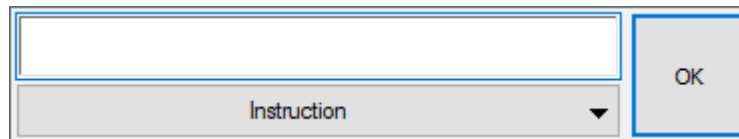


Основной интерфейс среды программирования

2.2.2 Добавление элементов на рабочее поле

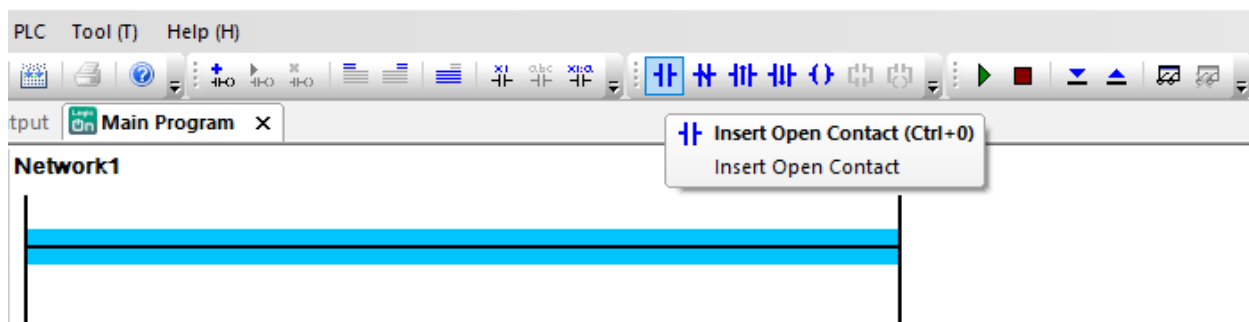
Для добавления элемента на рабочее поле существует несколько способов:

- Сделать двойной щелчок левой кнопкой мыши на участке сети и ввести название инструкции в поле ввода инструкции или выбрать инструкцию из выпадающего списка:



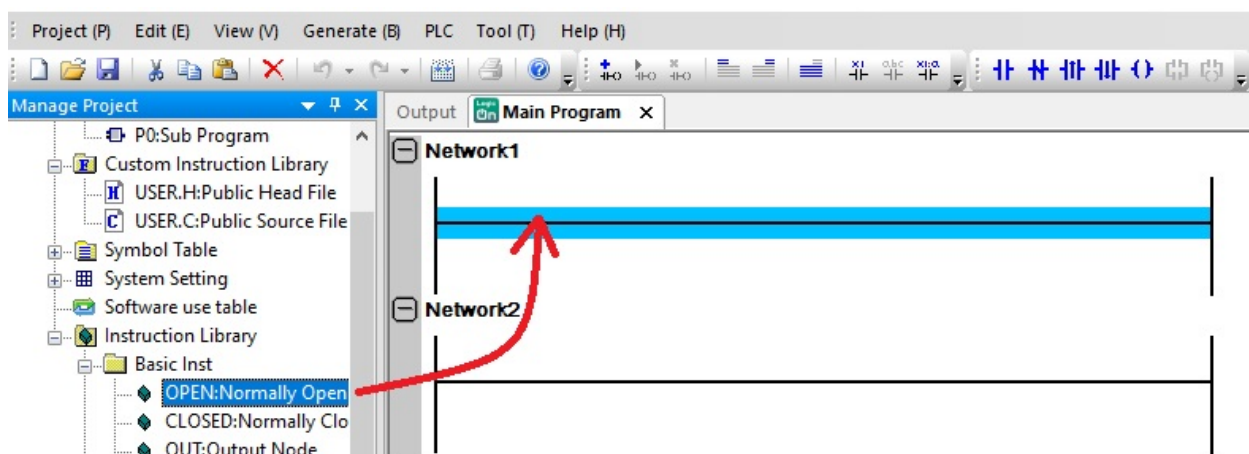
Поле ввода и выбора инструкций

- Левым щелчком мыши выделить сеть и вставить необходимый элемент из панели инструментов редактирования программ на языке LD:



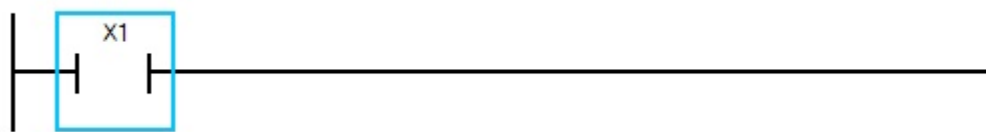
Добавление элемента с помощью панели инструментов

- Выбрать левой кнопкой мыши необходимую инструкцию из библиотеки инструкций и перетащить на нужную сеть:



Добавление элемента на рабочее поле из библиотеки инструкций

После установки элемента необходимо сверху написать его тип и номер.



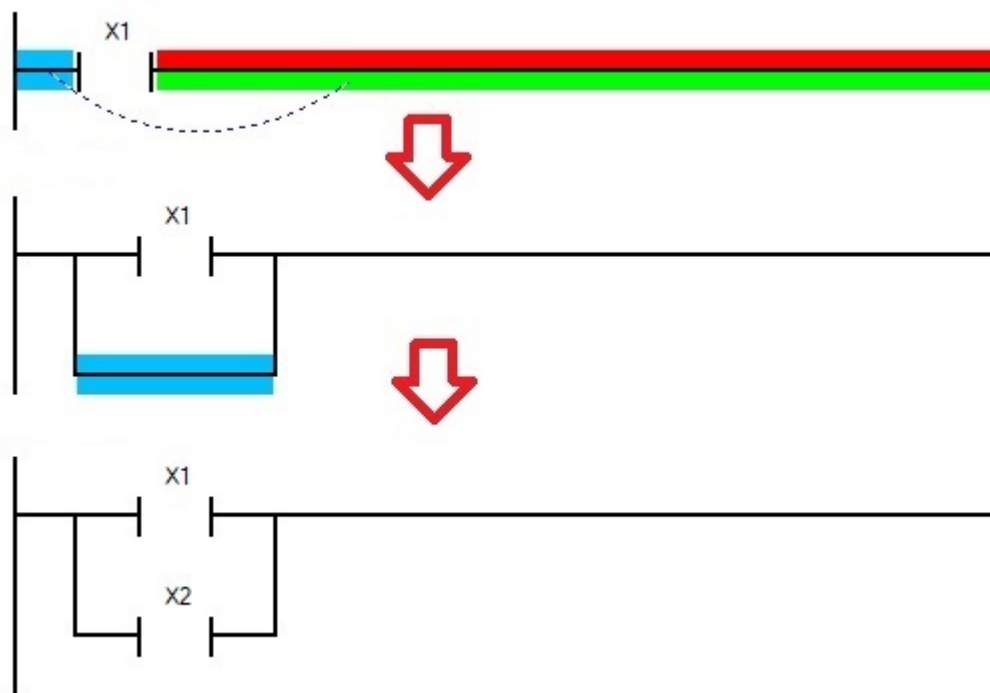
Входное реле X1 установленное в начало сети 1

Для установки элемента последовательно с уже установленным необходимо выделить участок сети до или после него и разместить необходимый элемент.



Установка элементов последовательно

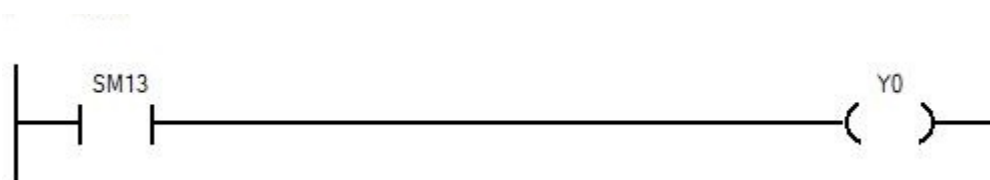
Для установки элемента или сети параллельно с уже установленными необходимо выделить участок сети с одной стороны элемента и удерживая левую кнопку мыши подвести линию к другой стороне. При этом сеть будет выделена цветом. Если необходимо установить элемент выше то линия подводится к области выделенной красным цветом. Если необходимо установить элемент ниже – то к выделенной зеленым. В результате появится участок сети, на котором можно разместить элемент.



Установка элементов параллельно

2.2.3 Пример простого проекта


На рисунке ниже показан пример проекта, который будет генерировать на выходе Y0 импульсы с периодом в 1 секунду.

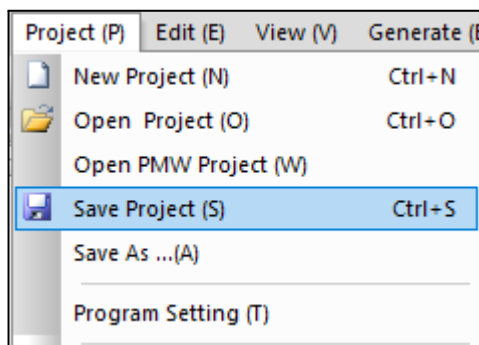


Пример простого проекта

В данном примере используется специальное реле SM13, которое реализует генерацию импульсов с периодом в 1 секунду. Полное описание специальных реле можно найти в разделе [3 ПРОГРАММНЫЕ ЭЛЕМЕНТЫ ПЛК](#).

2.2.4 Сохранение проекта

После создания проекта его необходимо сохранить – «Project» > «Save Project» или нажать на иконку  на панели инструментов, затем выбрать папку расположения и задать имя проекту.



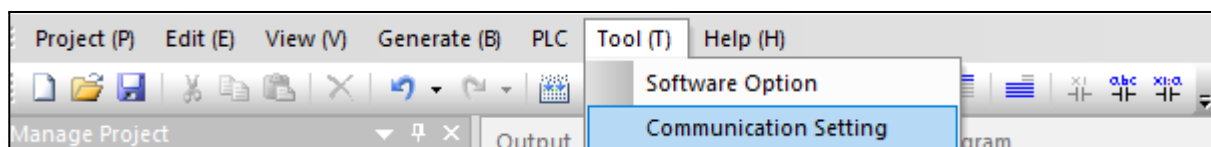
Сохранение проекта

2.2.5 Компиляция

Перед загрузкой в контроллер проект необходимо скомпилировать, для этого в главном меню нужно выбрать «Generate» > «Compile».

2.2.6 Выбор интерфейса для связи с контроллером

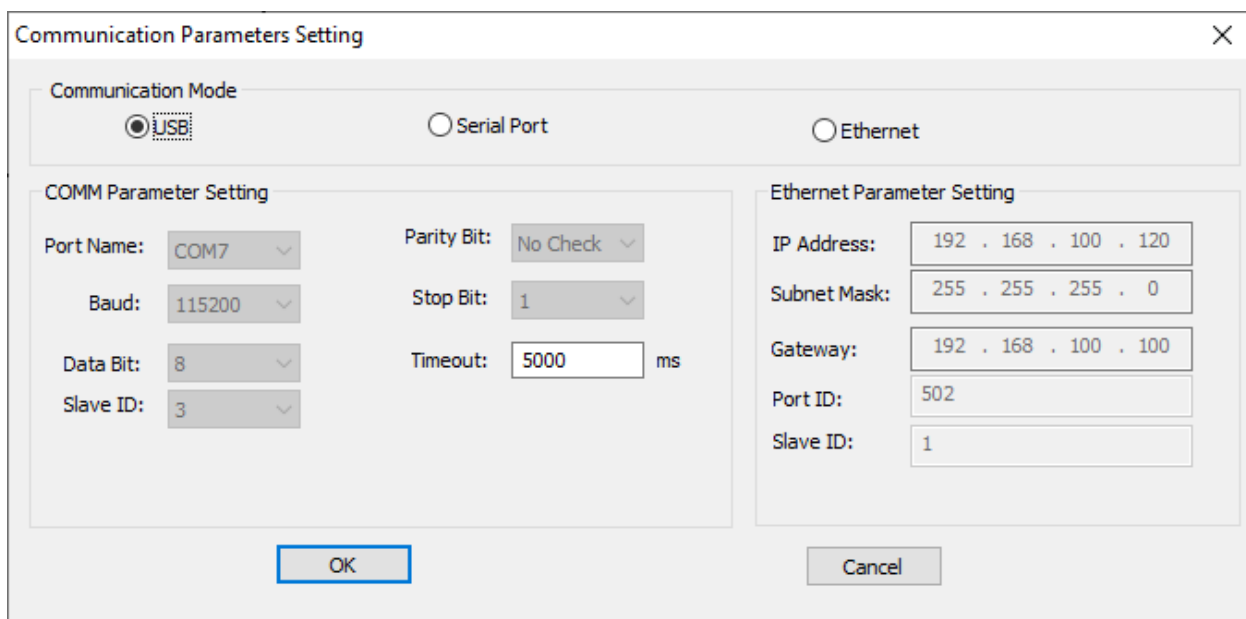
Загрузка программы в ПЛК может производиться через интерфейсы USB, RS-232, RS-485 и Ethernet. Способ загрузки выбирается в главном меню «Tool» > «Communication Setting».



Выбор меню настроек связи

2.2.7 Настройка связи по USB

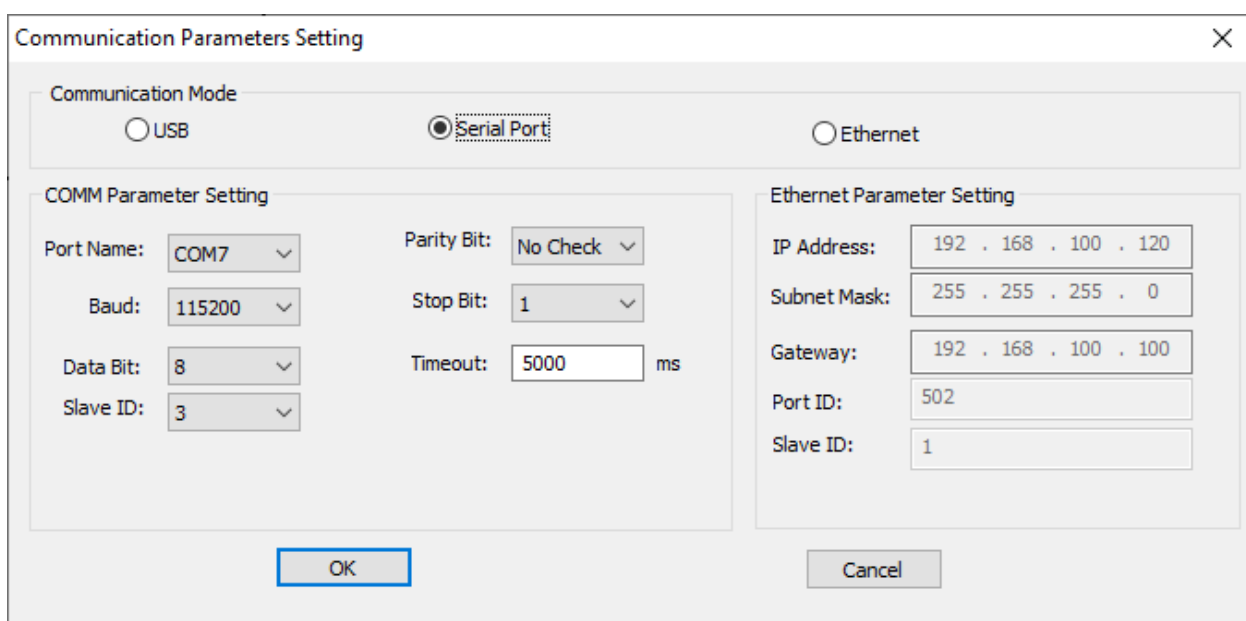
Для загрузки необходимо подключить ПЛК к компьютеру кабелем USB-A – microUSB, выбрать в окне настройки «Communication Mode» - «USB» и нажать «OK». При данном способе подключения дополнительных настроек не требуется.



Настройка параметров связи по USB

2.2.8 Настройка связи по RS-232/RS-485

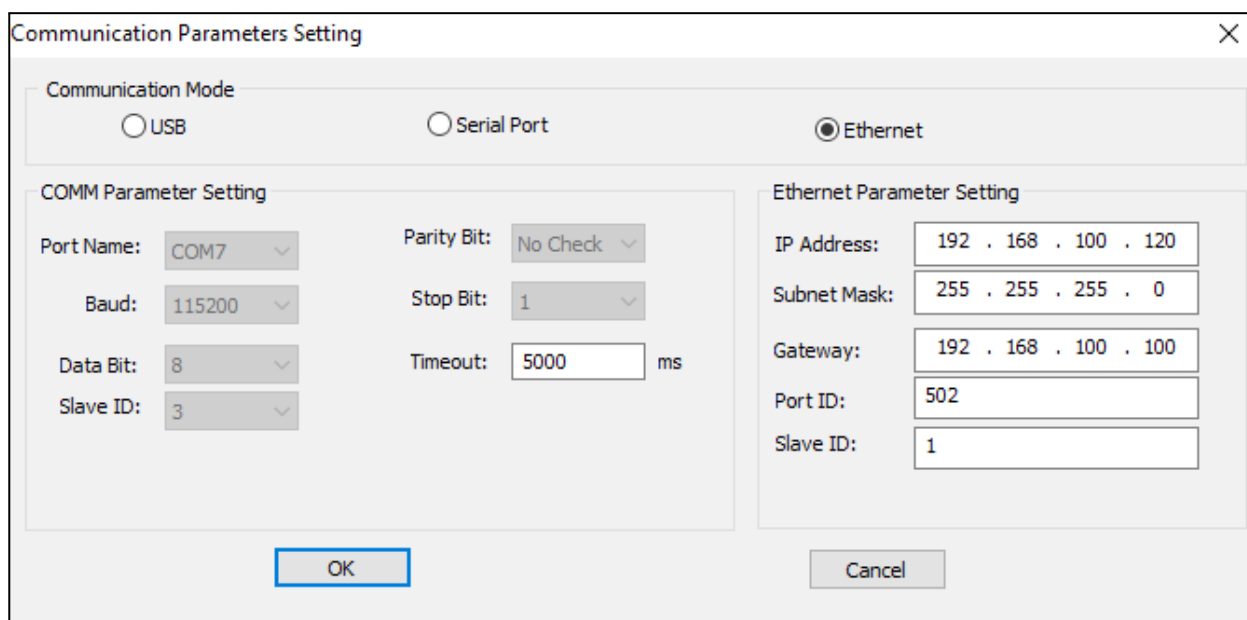
При данном способе подключения в настройках «Tool» > «Communication Setting» необходимо выбрать «Communication Mode» - «Serial Port», а также выбрать соответствующий COM-порт компьютера, к которому в данный момент подключен ПЛК и задать настройки связи.



Настройка параметров связи по RS-232/RS-485

2.2.9 Настройка связи по Ethernet

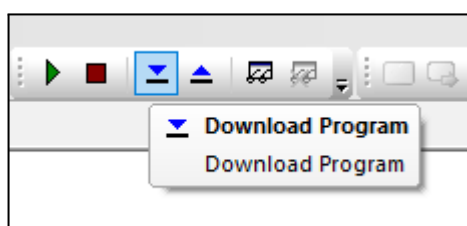
Для данного типа подключения ПЛК необходимо подключить к сети Ethernet, в настройках «Tool»>«Communication Setting» необходимо выбрать «Communication Mode» - «Ethernet» и задать настройки сетевых параметров. Сетевые параметры ПЛК по умолчанию показаны на рисунке ниже.



Настройка параметров связи по Ethernet

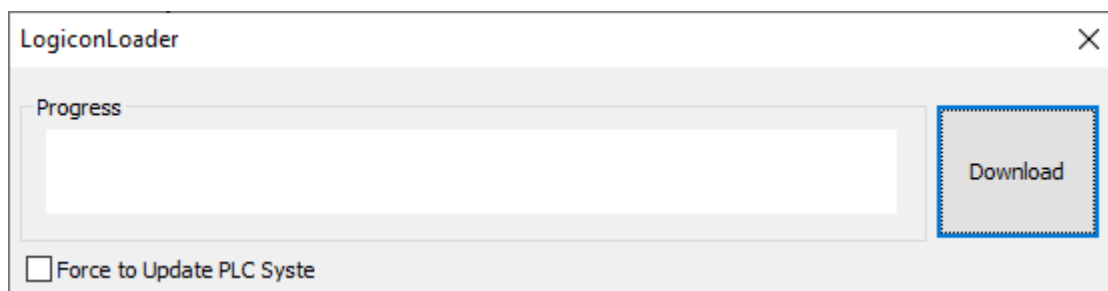
2.2.10 Загрузка программы

На панели инструментов «PLC Tool» нажать «Download Program»



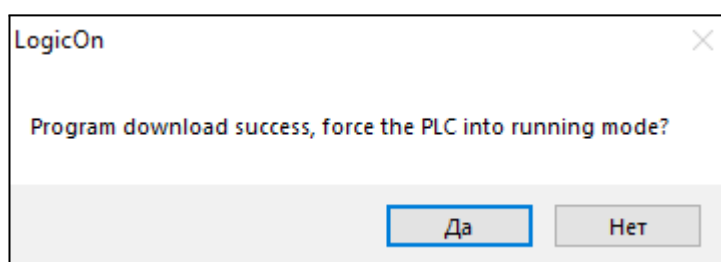
Загрузка программы

В открывшемся окне загрузчика нажать кнопку «Download» после чего пойдет процесс загрузки проекта в ПЛК.



Окно загрузчика

После успешной загрузки программы будет предложено запустить выполнение программы.



Диалоговое окно запуска выполнения программы

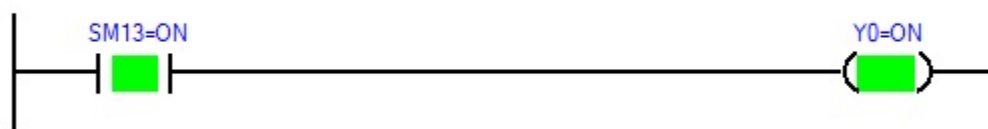
2.2.11 Мониторинг исполнения программы

В программном обеспечении ELHART LogicOn Soft есть возможность отслеживать процесс исполнения программы в контроллере, изменять значения в регистрах, изменять состояние битовых элементов и т.д. в режиме реального времени. Для запуска режима мониторинга необходимо на панели инструментов нажать на кнопку «Start Monitor»



Запуск мониторинга исполнения программы

После запуска режима мониторинга на рассматриваемом примере будет видно состояние специального реле SM13 и выхода Y0.



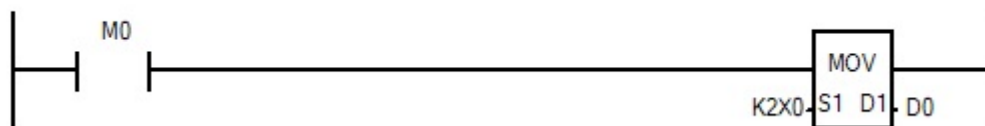
Отображение программы в режиме мониторинга

3 ПРОГРАММНЫЕ ЭЛЕМЕНТЫ ПЛК

Доступные программные элементы, их тип и количество для ПЛК LogicOn приведены в таблице ниже:

Элемент	Диапазон	Примечание
Входные реле X	X0 – X377	Физические дискретные входы ПЛК и модулей расширения
Выходные реле Y	Y0 – Y377	Физические дискретные выходы ПЛК и модулей расширения
Аналоговые входы AI	0 – 255	Физические аналоговые входы
Аналоговые выходы AQ	0 – 255	Физические аналоговые выходы
Внутренние реле M	M0 – M2047	Хранение промежуточных битовых значений
Специальные реле SM	SM0 – SM511	Выполнение специальных функций
Регистры данных D	D0 – D4095	Хранение данных
Специальные регистры данных SD	SD0 – SD511	Системные данные и выполнение специальных функций
Шаговые реле S	S0 – S999	Используются при организации STL - программы
Таймеры T	T0 – T255	Организация временных интервалов
Счетчики C	C0 – C255	Организация функции счета
Указатель L	0 – 127	Переход к указанной метке
Указатель P	0 – 127	Вызов подпрограммы
Указатель I	0 – 11, 16 – 18	Переход к обработке прерывания
Индексные регистры V, Z	0 – 7	Организация косвенной адресации
Объединения KnX, KnY, KnM, KnS, KnSM		Объединение битовых элементов

Битовые элементы X, Y, M, S, SM в прикладных инструкциях можно объединять в регистры данных вида KnX, KnY, KnM, KnS, KnSM, где n - количество бит, объединяемых в регистр выраженное в полубайтах. На примере ниже когда M0 замкнут битовые значение дискретных входов в количестве двух полубайт (всего восемь элементов) X0 - X7 помещаются в биты 0 - 7 регистра D0, при этом биты 8-15 регистра D0 будут равны 0.



Объединение битовых значений в регистр данных

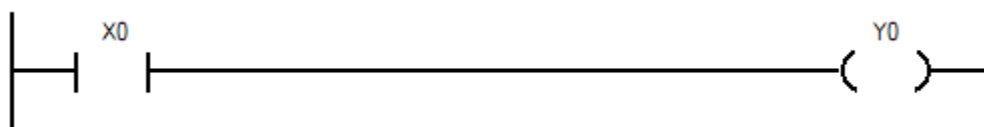
Значение n может быть от 1 до 4 для 16-ти битных инструкций и от 1 до 8 для 32-х битных инструкций.

3.1 Входные реле X

Программные элементы, которые непосредственно связаны с физическими входами ПЛК и модулей расширения. Состояние реле X не может быть изменено программно. Адресация входов осуществляется в восьмеричной системе счисления, поэтому например можно обратиться к входам X0 и X7, а входов с номерами X8, X9, X18, X19 - не существует.

Модель ПЛК	Общее количество реле X	Диапазон
CPU 12/8	12	X0...X7, X10...X13
CPU 16/16	16	X0...X7, X10...X17
CPU 24/24	24	X0...X7, X10...X17, X20...X27

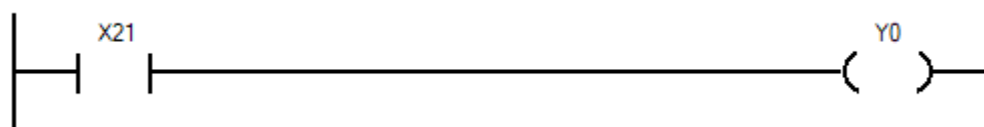
Пример работы с дискретными входами:



Когда замкнут вход X0, включен выход Y0

При обращении к модулям расширения адресация первого модуля будет начинаться с цифры 0 следующего 8-ричного разряда от максимального номера собственных входов контроллера. Адресация второго и последующего модулей строится аналогично с учетом количества входов ПЛК и уже установленных модулей. Таким образом, CPU 12/8N имеет 12 входов и вход с максимальным номером это X13. Соответственно, что бы обратиться к первому модулю расширения имеющим 4 дискретных входа необходимо использовать адреса X20...X23, ко второму – X30...X33 и т.д.

Например, к ПЛК CPU 12/8N подключен модуль расширения DIO 4/4N имеющий 4 дискретных входа. Необходимо управлять дискретным выходом ПЛК Y0 со второго по счету входа модуля расширения (вход на модуле расширения обозначен как X1). Тогда программа будет иметь следующий вид:



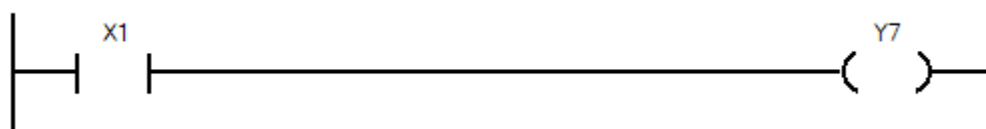
Y0 включен, когда замкнут вход X1 на модуле расширения

3.2 Выходные реле Y

Программные элементы, которые непосредственно связаны с дискретными выходами ПЛК и модулей расширения. Адресация выходов осуществляется в восьмеричной системе счисления (аналогично входам X), поэтому, например, можно обратиться к выходам Y0 и Y7, а выходов с номерами Y8, Y9, Y18, Y19 не существует.

Модель ПЛК	Общее количество реле Y	Диапазон
CPU 12/8	8	Y0...Y7
CPU 16/16	16	Y0...Y7, Y10...Y17
CPU 24/24	24	Y0...Y7, Y10...Y17, Y20...Y27

Пример работы с дискретными выходами:

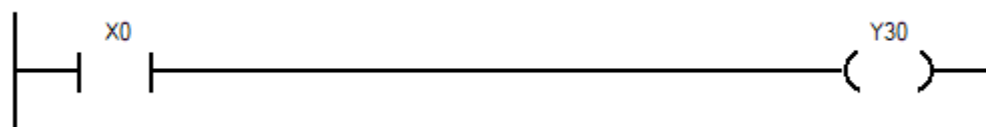


Выход Y7 включен, когда замкнут X1

При использовании модулей расширения адресация первого модуля с дискретными выходами при подключении к CPU 12/8 и CPU 16/16 начинается с Y20. При подключении к CPU 24/24 адресация начинается с Y30.

Т.к., CPU 24/24N имеет 24 выхода, то чтобы обратиться к первому модулю расширения на 8 дискретных выходов необходимо использовать адреса Y30...Y37, ко второму – Y40...Y47 и т.д.

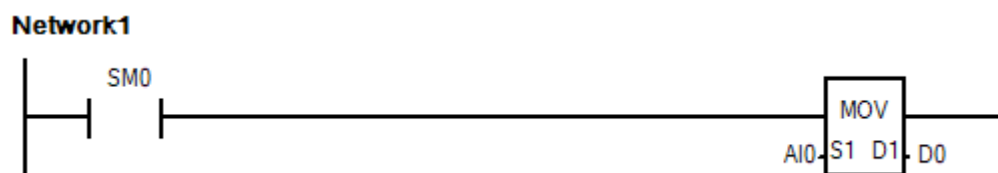
Например, к ПЛК CPU 24/24N подключен модуль расширения DIO 4/4N имеющий 4 дискретных выхода. Необходимо управлять первым дискретным выходом модуля расширения (выход на модуле расширения обозначен как Y0) с входа X0 ПЛК. Тогда программа будет иметь следующий вид:



Выход Y0 на модуле расширения включен, когда замкнут X0 на ПЛК

3.3 Аналоговые входы AI

Программные элементы, непосредственно связанные с аналоговыми входами ПЛК и модулей расширения. Адресация аналоговых входов осуществляется в десятичной системе счисления



Использование аналоговых входов

На примере выше, данные с аналогового входа AI0 помещаются в регистр D0.

Данные аналоговых сигналов по температуре передаются в виде целого числа с разрешением 0,1 C°. Например, если с аналогового входа измерения температуры пришло значение 298, значит фактическая измеренная температура составляет 29,8 C°.

Данные с аналоговых входов по напряжению 0-10В передаются в виде числа от 0 до 10 000. Например, если с аналогового входа по напряжению пришло значение 5600, значит фактическое измеренное напряжение составляет 5,6 В.

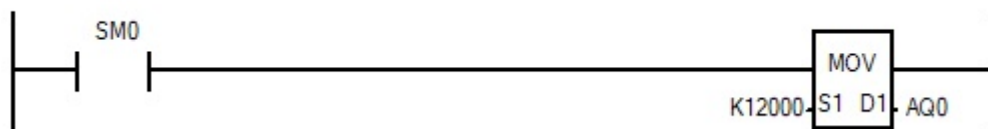
Данные с аналоговых входов по току 0/4-20мА приходят в виде числа от 0 до 20000. Например, если с аналогового входа по току пришло значение 18000, то фактический измеренный ток составляет 18 мА.

Подробная информация по добавлению модулей расширения в проект доступна в разделе [5.3 Подключение модулей расширения](#).

3.4 Аналоговые выходы AQ

Программные элементы, непосредственно связанные с аналоговыми выходами ПЛК и модулей расширения. Адресация аналоговых выходов осуществляется в десятичной системе счисления.

Данные на аналоговые выходы передаются в том же формате, что и получаются с аналоговых входов. Например, если требуется установить на аналоговом выходе по напряжению значение 8,2В, то в соответствующий регистр AQ необходимо передать значение 8200. Если требуется на аналоговом выходе по току установить значение 15 мА, то в регистр AQ необходимо передать значение 15000. На примере ниже на аналоговый выход по току AQ0 передается значение 12000, что соответствует 12 мА на физическом аналоговом выходе.



Установка значения на аналоговом выходе

3.5 Внутренние реле M

Программные элементы, применяемые для хранения промежуточных логических состояний. Адресация внутренних реле десятичная поэтому, в отличии от дискретных входов/выходов, можно использовать элементы с номерами M9, M18 и т.д.

Есть стандартные внутренние реле, значение которых не сохраняется при отключении питания и энергонезависимые, которые сохраняют свое значение при отключении питания. Диапазон адресов энергонезависимых реле можно настроить в среде разработки(подробнее о настройке в приложении А, раздел «Poweroff Save Setting»). По умолчанию энергонезависимые внутренние реле – M500 – M1023.

Модель ПЛК	Общее количество реле M	Диапазон
CPU 12/8	2048	M0...M2047
CPU 16/16		
CPU 24/24		

На примере ниже показано использование внутреннего энергонезависимого реле. Когда X0 включится, Y0 так же включится и сохранит свое состояние до отключения питания, а также восстановит включенное состояние после включения питания.



Пример использования внутреннего реле

3.6 Специальные (системные) внутренние реле SM

Данные программные элементы используются для выполнения специальных функций. Условно специальные реле можно разделить на два типа:

1. Специальные реле, которые пользователь может использовать в программе, но не может их изменить. Например, специальное реле SM0 – включено всегда, когда ПЛК находится в состоянии «RUN». Или специальное реле SM12, которое включается и выключается с периодом 100 миллисекунд.

2. Специальные реле, которые пользователь может изменять в программе. Например, SM200 – направление счета счетчика C200. Изменяя состояние SM200, пользователь может изменять направление счета счетчика C200.

Описание специальных внутренних реле приведено в таблице ниже:

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
SM0	Да	Нет	AlwaysON	Включен, когда ПЛК находится в режиме «RUN»
SM1	Да	Нет	AlwaysOFF	Выключен, когда ПЛК находится в режиме «RUN»
SM2	Да	Нет	FirstScanON	Включается на один цикл сканирования при переходе ПЛК в режим «RUN»
SM3	Да	Нет	FirstScanOFF	Выключается на один цикл сканирования при переходе ПЛК в режим «RUN»
SM11	Да	Нет	10msCycle	Генератор импульсов с периодом 10 мс (5 мс ON/ 5 мс OFF)
SM12	Да	Нет	100msCycle	Генератор импульсов с периодом 100 мс (50 мс ON/ 50 мс OFF)
SM13	Да	Нет	1sCycle	Генератор тактовых импульсов с периодом 1 секунду (0.5 с ON/ 0.5 с OFF)
SM14	Да	Нет	1minCycle	Генератор тактовых импульсов с периодом 1 минуту (30 с ON/ 30 с OFF)
SM15	Да	Да	CalClock	Разрешение установки времени. Если включен, то часы реального времени останавливаются и можно задать значения времени в специальных регистрах SD13-SD19 (только для CPU)

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				24/24).
SM16	Да	Да	StopDisClock	Сохранения текущего значения времени в регистрах SD13...SD19. Если включен, данные в регистрах сохраняются но часы продолжают идти (только для CPU 24/24).
SM20	Да	Нет	ZeroFlag	Флаг нуля. Включается, если результат операции сложения или вычитания равен 0.
SM21	Да	Нет	BorrowFlag	Флаг заимствования. Включается, если результат вычитания меньше самого малого значения.
SM22	Да	Нет	CarryFlag	Флаг переноса. Включается при передаче значения числа, при суммировании или передачи данных, при выполнении инструкций сдвига.
SM24	Да	Да	BMOV_DIR	Направление передачи данных при использовании инструкции BMOV. Если выключен, то передача из S1 в D1. Если включен – из D1 в S1.
SM34	Да	Да	OuputTable	Запрещение работы выходов.
SM39	Да	Да	ConstScanMode	Включение постоянного времени сканирования. Программа будет исполняться со временем цикла заданным в SD39.
SM48	Да	Нет	HasAlarm	Индикация включения сигнала тревоги. Срабатывает, если включено специальное реле управления сигнализацией SM49 и сработал хотя бы один из сигналов. Подробнее о режиме тревожной сигнализации можно посмотреть в описании инструкций ANS и ANR.
SM49	Да	Да	AlarmEnable	Специальное реле управления сигнализацией. Если SM49 включено, то при срабатывании любой из сигнализаций включится специальное реле SM48 а в регистре SD49 будет

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				сохранен номер сигнализации с младшим номером. Подробнее о режиме тревожной сигнализации можно посмотреть в описании инструкций ANS и ANR.
SM50	Да	Да	INT_X0_DISABLE	Запрет прерывания по входу X0 (I0).
SM51	Да	Да	INT_X1_DISABLE	Запрет прерывания по входу X1 (I1).
SM52	Да	Да	INT_X2_DISABLE	Запрет прерывания по входу X2 (I2).
SM53	Да	Да	INT_X3_DISABLE	Запрет прерывания по входу X3 (I3).
SM54	Да	Да	INT_X4_DISABLE	Запрет прерывания по входу X4 (I4).
SM55	Да	Да	INT_X5_DISABLE	Запрет прерывания по входу X5 (I5).
SM56	Да	Да	INT_TIMER0_DISABLE	Запрет прерывания по таймеру 0 (I16).
SM57	Да	Да	INT_TIMER1_DISABLE	Запрет прерывания по таймеру 1 (I17).
SM58	Да	Да	INT_TIMER2_DISABLE	Запрет прерывания по таймеру 2 (I18).
SM61	Да	Нет	HARD_ERR	Аппаратная неисправность ПЛК.
SM67	Да	Нет	CAL_ERR	Ошибка выполнения инструкции. Возникает при недопустимых значениях параметров для инструкции.
SM145	Нет	Да		Остановка импульсного выхода Y0.
SM146	Нет	Да		Остановка импульсного выхода Y1.
SM147	Да	Нет		Состояние импульсного выхода Y0 ON - занят, OFF - готов.
SM148	Да	Нет		Состояние импульсного выхода Y1 ON - занят, OFF - готов.
SM149	Нет	Да		Остановка импульсного выхода Y2.
SM150	Нет	Да		Остановка импульсного выхода Y3.
SM151	Да	Нет		Состояние импульсного выхода Y2 ON -

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				занят, OFF - готов.
SM152	Да	Нет		Состояние импульсного выхода Y3 ON - занят, OFF - готов.
SM153	Нет	Да		Остановка импульсного выхода Y4.
SM154	Нет	Да		Остановка импульсного выхода Y5.
SM155	Да	Нет		Состояние импульсного выхода Y4 ON - занят, OFF - готов.
SM156	Да	Нет		Состояние импульсного выхода Y5 ON - занят, OFF - готов.
SM157	Нет	Да		Остановка импульсного выхода Y6.
SM158	Нет	Да		Остановка импульсного выхода Y7.
SM159	Да	Нет		Состояние импульсного выхода Y6 ON - занят, OFF - готов.
SM160	Да	Нет		Состояние импульсного выхода Y7 ON - занят, OFF - готов.
SM168	Да	Да		Если включен, то выполнение инструкции SMOV происходит в шестнадцатеричном формате.
SM169	Да	Да		Специальный режим выполнения инструкции XCH (см. описание инструкции XCH).
SM170	Да	Да		Захват импульса со входа X0.
SM171	Да	Да		Захват импульса со входа X1.
SM172	Да	Да		Захват импульса со входа X2.
SM173	Да	Да		Захват импульса со входа X3.
SM174	Да	Да		Захват импульса со входа X4.
SM175	Да	Да		Захват импульса со входа X5.
SM200- SM255	Нет	Да		Выбор направления счета соответствующего счетчика. Если выключен - счетчик считает в прямом

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				направлении, если включен – счетчик считает в обратном направлении.
SM341	Да	Да		Разблокировка сброса выхода Y0
SM342	Да	Да		Направление движения возврата в нулевую точку для Y0: OFF – Возврат в нулевую точку происходит при движении в обратном направлении ON – Возврат в нулевую точку происходит при движении в прямом направлении
SM343	Да	Да		Концевой выключатель при движении вперед для выхода Y0
SM344	Да	Да		Концевой выключатель при движении назад для выхода Y0
SM345	Да	Да		Тип сигнала начала зоны DOG для входа X0: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM346	Да	Да		Тип сигнала окончания зоны DOG для входа X0: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM348	Да	Нет		Работает команда позиционирования на выходе Y0
SM464	Да	Да		Если включен, то в базовой точке сбрасывается не выход Y0, а соответствующий элемент указанный в специальном регистре SD464
SM351	Да	Да		Разблокировка сброса выхода Y1
SM352	Да	Да		Направление движения возврата в нулевую точку для Y1: OFF – Возврат в нулевую точку происходит при движении в обратном направлении ON – Возврат в нулевую точку

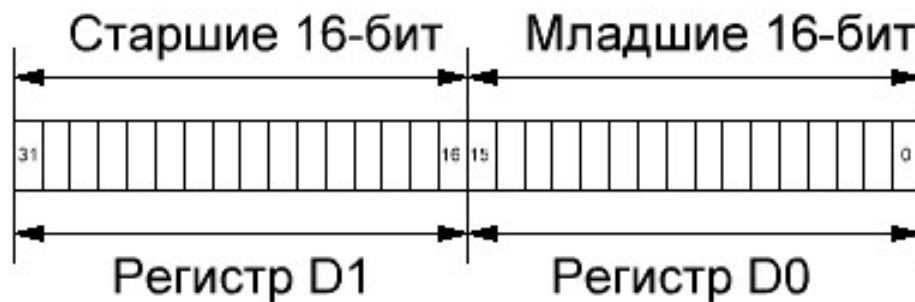
Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				происходит при движении в прямом направлении
SM353	Да	Да		Концевой выключатель при движении вперед для выхода Y1
SM354	Да	Да		Концевой выключатель при движении назад для выхода Y1
SM355	Да	Да		Тип сигнала начала зоны DOG для входа X1: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM356	Да	Да		Тип сигнала окончания зоны DOG для входа X1: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM358	Да	Нет		Работает команда позиционирования на выходе Y1
SM465	Да	Нет		Если включен, то в базовой точке сбрасывается не выход Y1, а соответствующий элемент указанный в специальном регистре SD465
SM361	Да	Да		Разблокировка сброса выхода Y2
SM362	Да	Да		Направление движения возврата в нулевую точку для Y2: OFF – Возврат в нулевую точку происходит при движении в обратном направлении ON – Возврат в нулевую точку происходит при движении в прямом направлении
SM363	Да	Да		Концевой выключатель при движении вперед для выхода Y2
SM364	Да	Да		Концевой выключатель при движении назад для выхода Y2
SM365	Да	Да		Тип сигнала начала зоны DOG для входа X2: OFF – нормально открытый контакт

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				ON – нормально закрытый контакт
SM366	Да	Да		Тип сигнала окончания зоны DOG для входа X2: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM368	Да	Нет		Работает команда позиционирования на выходе Y2
SM466	Да	Да		Если включен, то в базовой точке сбрасывается не выход Y2, а соответствующий элемент указанный в специальном регистре SD466
SM371	Да	Да		Разблокировка сброса выхода Y3
SM372	Да	Да		Направление движения возврата в нулевую точку для Y3: OFF – Возврат в нулевую точку происходит при движении в обратном направлении ON – Возврат в нулевую точку происходит при движении в прямом направлении
SM373	Да	Да		Концевой выключатель при движении вперед для выхода Y3
SM374	Да	Да		Концевой выключатель при движении назад для выхода Y3
SM375	Да	Да		Тип сигнала начала зоны DOG для входа X3: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM376	Да	Да		Тип сигнала окончания зоны DOG для входа X3: OFF – нормально открытый контакт ON – нормально закрытый контакт
SM378	Да	Нет		Работает команда позиционирования на выходе Y3
SM467	Да	Да		Если включен, то в базовой точке сбрасывается не выход Y3, а

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				соответствующий элемент указанный в специальном регистре SD467

3.7 Регистры данных D

Регистры данных используются для хранения данных таких как константы для арифметических операций, хранение значений промежуточных вычислений, уставок таймера, счетчика, данные с аналоговых входов и т.д. Каждый регистр имеет размерность в 16 бит. Если для операции с регистрами используется 32-битная инструкция, то соседние регистры будут автоматически сгруппированы в 32-битный регистр. В регистре с меньшим номером хранятся младшие 16 бит, в регистре с большим – старшие 16 бит.



32-х битное число в двух соседних 16-ти битных регистрах

При вычислениях ПЛК хранит данные в 16-битном регистре в виде 15-битного числа, а в последнем разряде хранится знак (0 – положительное число, 1 – отрицательное).



Структура 16-ти битного регистра



Структура 32-х битного регистра (два 16-ти битных регистра).

Диапазон значений для 16-битного регистра от -32 768 до +32 767. При 32-битных операциях (два регистра) диапазон значений составляет от -2 147 483 648 до 2 147 483 647.

Все регистры данных подразделяются на стандартные, энергонезависимые, специальные и индексные.

Стандартные регистры не сохраняют свое значение после отключения питания.

Энергонезависимые сохраняют значение после отключения питания. Диапазон адресов энергонезависимых регистров можно изменить.

Специальные регистры используются для работы с системными данными и выполнения специальных функций.

Индексные регистры – используются для реализации косвенной адресации.

Стандартные	Энергонезависимые	Специальные	Индексные
D0 – D199, D512-D4095	D200 -D511*	SD0 -SD511	V0-V7, Z0-Z7

* Диапазон адресов энергонезависимых регистров можно изменить.

3.8 Специальные регистры SD

Специальные регистры используются для работы с системными данными ПЛК и выполнения специальных функций. Все специальные регистры можно разделить на два типа:

1. Которые можно только считывать и нельзя изменять в программе;
2. Которые можно считывать и изменять в программе.

Описание специальных регистров приведено в таблице ниже:

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
SD0	Да	Да		Уставка таймера контроля времени цикла(Watch-dog). Единица измерения 1 миллисекунда, по умолчанию значение 200 мс.
SD1	Да	Нет		Номер версии внутреннего программного обеспечения ПЛК.
SD10	Да	Нет		Текущее время сканирования. Единица измерения 0.1 мс.
SD11	Да	Нет		Минимальное время сканирования. Единица измерения 0.1 мс
SD12	Да	Нет		Максимальное время сканирования. Единица измерения 0.1 мс
SD13	Да	Да		Часы реального времени, секунды (0...59)*
SD14	Да	Да		Часы реального времени, минуты (0...59)*
SD15	Да	Да		Часы реального времени, часы (0...23)*
SD16	Да	Да		Часы реального времени, день (1...31)*
SD17	Да	Да		Часы реального времени, месяц (1...12)*
SD18	Да	Да		Часы реального времени, год (0...99)*
SD19	Да	Да		Часы реального времени, день недели (0...6)*
SD20	Нет	Да		Время фильтрации входного сигнала на дискретных входах. Диапазон 0 - 60 мс,

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				по умолчанию 10 мс.
SD28	Да	Нет		Текущее значение в индексном регистре Z0
SD29	Да	Нет		Текущее значение в индексном регистре V0
SD39	Нет	Да		Задание постоянного времени сканирования, единица измерения 1 мс.
SD49	Да	Нет		Номер последней активированной тревожной сигнализации, диапазон от S900 до S999 (см. описание инструкций ANS и ANR).
SD61	Да	Нет		Код ошибки при аппаратном сбое
SD82	Да	Нет		Текущее значение в индексном регистре Z1
SD83	Да	Нет		Текущее значение в индексном регистре V1
SD84	Да	Нет		Текущее значение в индексном регистре Z2
SD85	Да	Нет		Текущее значение в индексном регистре V2
SD86	Да	Нет		Текущее значение в индексном регистре Z3
SD87	Да	Нет		Текущее значение в индексном регистре V3
SD88	Да	Нет		Текущее значение в индексном регистре Z4
SD89	Да	Нет		Текущее значение в индексном регистре V4
SD90	Да	Нет		Текущее значение в индексном регистре Z5
SD91	Да	Нет		Текущее значение в индексном регистре V5

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
SD92	Да	Нет		Текущее значение в индексном регистре Z6
SD93	Да	Нет		Текущее значение в индексном регистре V6
SD94	Да	Нет		Текущее значение в индексном регистре Z7
SD95	Да	Нет		Текущее значение в индексном регистре V7
SD140 SD141	Да	Нет		Количество выданных импульсов на выходе Y0, 32-бит данных.
SD142 SD143	Да	Нет		Количество выданных импульсов на выходе Y1, 32-бит данных.
SD145	Нет	Да		Базовая скорость при использовании инструкций DRVI и DRVA
SD146	Нет	Да		Максимальная скорость при использовании инструкций DRVI и DRVA
SD148	Нет	Да		Время разгона и замедления при использовании инструкций DRVI и DRVA
SD150 SD151	Да	Нет		Количество выданных импульсов на выходе Y2, 32-бит данных.
SD152 SD153	Да	Нет		Количество выданных импульсов на выходе Y3, 32-бит данных.
SD154 SD155	Да	Нет		Количество выданных импульсов на выходе Y4, 32-бит данных.
SD156 SD157	Да	Нет		Количество выданных импульсов на выходе Y5, 32-бит данных.
SD158 SD159	Да	Нет		Количество выданных импульсов на выходе Y6, 32-бит данных.
SD160 SD161	Да	Нет		Количество выданных импульсов на выходе Y7, 32-бит данных.
SD235	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				C235
SD236	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C236
SD237	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C237
SD238	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C238
SD239	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C239
SD240	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C240
SD241	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C241
SD242	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C242
SD243	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C243
SD244	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C244
SD245	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C245
SD246	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C246
SD247	Да	Нет		Номер текущей стадии ступенчатой

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
				импульсной инструкции для счетчика C247
SD248	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C248
SD249	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C249
SD250	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C250
SD251	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C251
SD252	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C252
SD253	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C253
SD254	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C254
SD255	Да	Нет		Номер текущей стадии ступенчатой импульсной инструкции для счетчика C255
SD276	Да	Нет		Первое слово уникального аппаратного кода
SD277	Да	Нет		Второе слово уникального аппаратного кода
SD278	Да	Нет		Третье слово уникального аппаратного кода
SD279	Да	Нет		Четвертое слово уникального аппаратного кода

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
SD280	Да	Нет		Пятое слово уникального аппаратного кода
SD281	Да	Нет		Шестое слово уникального аппаратного кода
SD342	Нет	Да		Базовая скорость для выхода Y0 при выполнении инструкции DSZR
SD343 SD344	Нет	Да		Максимальная скорость для выхода Y0 при выполнении инструкции DSZR, 32 бит.
SD345	Нет	Да		Скорость подхода к цели(ползучая скорость) для выхода Y0 при выполнении инструкции DSZR
SD346 SD347	Нет	Да		Скорость возврата в нулевую точку для выхода Y0 при выполнении инструкции DSZR, 32 бит.
SD348	Нет	Да		Время ускорения для выхода Y0 при выполнении инструкции DSZR
SD349	Нет	Да		Время замедления для выхода Y0 при выполнении инструкции DSZR
SD464	Нет	Да		Номер программного элемента, который сбрасывается при выполнении инструкций DSZR и ZNR для выхода Y0
SD352	Нет	Да		Базовая скорость для выхода Y1 при выполнении инструкции DSZR
SD353 SD354	Нет	Да		Максимальная скорость для выхода Y1 при выполнении инструкции DSZR, 32 бит.
SD345	Нет	Да		Скорость подхода к цели(ползучая скорость) для выхода Y1 при выполнении инструкции DSZR
SD356 SD357	Нет	Да		Скорость возврата в нулевую точку для выхода Y0 при выполнении инструкции DSZR, 32 бит.
SD358	Нет	Да		Время ускорения для выхода Y1 при выполнении инструкции DSZR

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
SD349	Нет	Да		Время замедления для выхода Y1 при выполнении инструкции DSZR
SD465	Нет	Да		Номер программного элемента, который сбрасывается при выполнении инструкций DSZR и ZNR для выхода Y1
SD362	Нет	Да		Базовая скорость для выхода Y2 при выполнении инструкции DSZR
SD363 SD364	Нет	Да		Максимальная скорость для выхода Y2 при выполнении инструкции DSZR, 32 бит.
SD355	Нет	Да		Скорость подхода к цели(ползучая скорость) для выхода Y2 при выполнении инструкции DSZR
SD366 SD367	Нет	Да		Скорость возврата в нулевую точку для выхода Y2 при выполнении инструкции DSZR, 32 бит.
SD368	Нет	Да		Время ускорения для выхода Y2 при выполнении инструкции DSZR
SD369	Нет	Да		Время замедления для выхода Y2 при выполнении инструкции DSZR
SD466	Нет	Да		Номер программного элемента, который сбрасывается при выполнении инструкций DSZR и ZNR для выхода Y2
SD372	Нет	Да		Базовая скорость для выхода Y3 при выполнении инструкции DSZR
SD373 SD374	Нет	Да		Максимальная скорость для выхода Y3 при выполнении инструкции DSZR, 32 бит.
SD375	Нет	Да		Скорость подхода к цели(ползучая скорость) для выхода Y3 при выполнении инструкции DSZR
SD376 SD377	Нет	Да		Скорость возврата в нулевую точку для выхода Y3 при выполнении инструкции DSZR, 32 бит.
SD378	Нет	Да		Время ускорения для выхода Y3 при выполнении инструкции DSZR

Номер	Опрос состоян.	Измен. состоян.	Текстовый идентифик.	Описание
SD379	Нет	Да		Время замедления для выхода Y3 при выполнении инструкции DSZR
SD467	Нет	Да		Номер программного элемента, который сбрасывается при выполнении инструкций DSZR и ZNR для выхода Y3

3.9 Шаговые реле S

Программные элементы, применяемые в функциях пошагового управления совместно с инструкцией STL. Если инструкция STL не используется, то шаговые реле S можно применять в качестве стандартных внутренних реле.

Подробнее о работе с шаговыми реле указано в описании инструкций STL и RET.

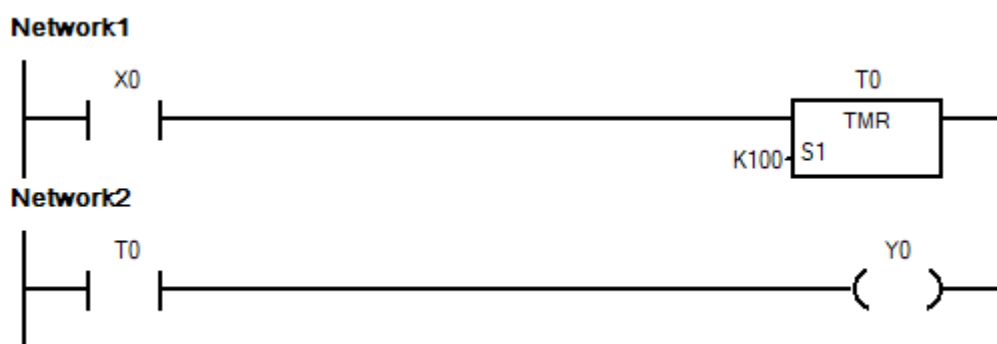
Модель ПЛК	Общее количество шаговых реле S	Стандартные	Энергонезависимые
CPU 12/8	1000	S0...S499	S500...S999
CPU 16/16			
CPU 24/24			

3.10 Таймеры Т

Программные элементы, применяемые для формирования временных интервалов в программе пользователя. Когда значение таймера достигает заданного, срабатывает связанный с ним контакт.

По принципу работы и назначению таймеры можно условно разделить на три группы: общего назначения, накопительные и таймеры для подпрограмм и прерываний.

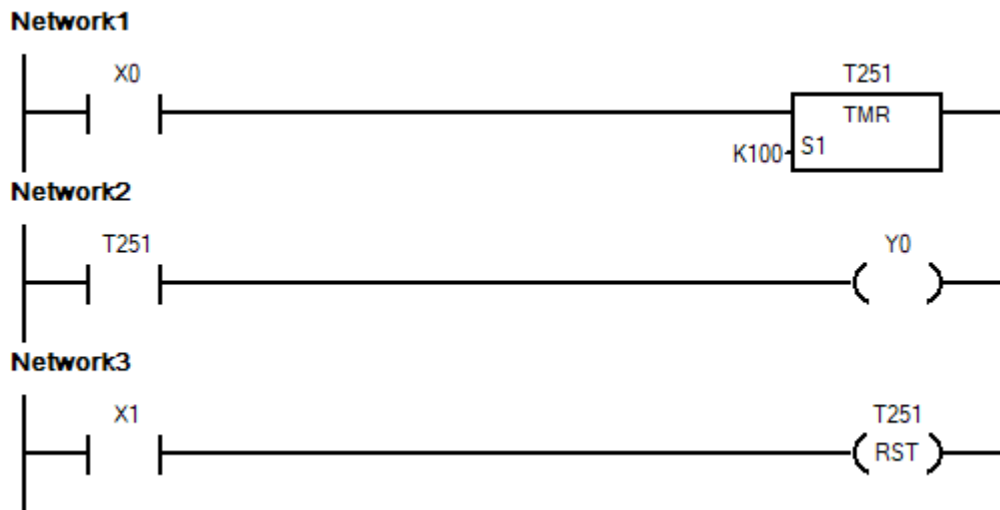
Общего назначения — при выключении питания или при снятии условия работы, таймеры сбрасывают накопленное значение;



Применение таймера общего назначения

Когда X0 включен, таймер T0 начинает отсчет и по достижению уставки срабатывает выход T0 и связанный с ним выход Y0. Если X0 будет отключен или выключено питание, значение таймера сбросится в 0 и выход Y0, работающий от контакта таймера T0, отключится.

Накопительные таймеры сохраняют накопленное значение после отключения питания или при снятии условия работы. После появления условия работы, если заданное значение не достигнуто, начинают считать с накопленного значения.



Применение накопительного таймера

Когда X0 включен, накопительный таймер T251 начинает отсчет до достижения заданного значения. Если X0 выключится или будет отключено напряжение питания, таймер сохранит накопленное значение и когда X0 снова включится продолжит отсчет с сохраненного значения. При включении X1 произойдет сброс накопленного значения таймера T251 и выключение связанного с ним контакта.

Все таймеры, за исключением T246 - T249, после достижения заданного значения останавливают отсчет времени. Таймеры T246 - T249 при наличии условия для работы после достижения уставки будут вести отсчет до максимального значения диапазона - 32,767 секунд или до снятия условия для работы.

Таймеры для подпрограмм и прерываний – используются для отсчета времени в подпрограммах или программах обработки прерываний. Для реализации отсчета времени в подпрограммах или программах обработки прерывания должны использоваться только таймеры T192 - T199. Остальные таймеры вне основной программы будут работать некорректно – отсчет будет вестись только тогда, когда подпрограмма будет вызываться.

Таймеры с разными номерами имеют разный временной шаг и особенности работы в соответствии с таблицей ниже:

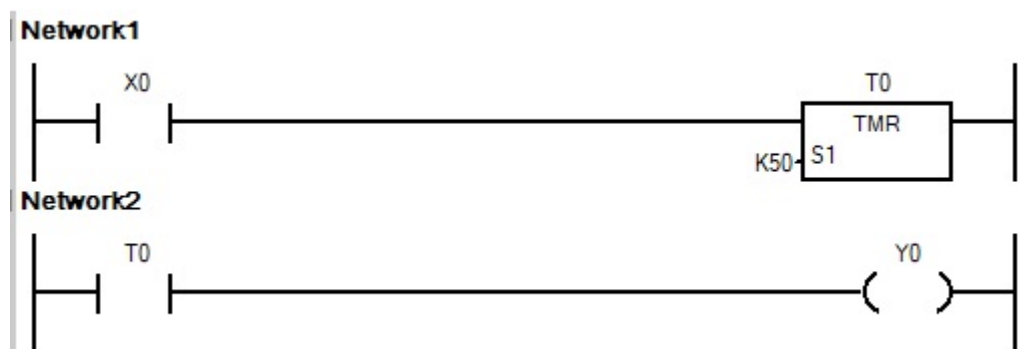
Номер	Шаг, мс	Диапазон, с	Количество	Примечание
T0 -T191	100	0,1 – 3276,7	192	Общего назначения
T192 -T199	100	0,1 – 3276,7	8	Таймеры для

Номер	Шаг, мс	Диапазон, с	Количество	Примечание
				подпрограмм
T200 -T245	10	0,01 – 327,67	46	Общего назначения
T246 -T249	1	0,001 – 32,767	4	С сохранением значения
T250 -T255	100	0,1 – 3276,7	6	С сохранением значения

Истинное значение времени равно уставке таймера умноженное на временной шаг.

На примере ниже используется таймер T0 с шагом 100 мс, уставка таймера 50. Истинное значение времени, отсчитываемое таймером, $50 \cdot 100 = 5000$ мс или 5 секунд.

При включении X0 таймер T0 начнет отсчет, после истечения 5-ти секунд включится нормально- открытый контакт таймера T0 и включится выход Y0.



Пример использования таймера

3.11 Счётчики С

Программные элементы, предназначенные для подсчета импульсов. При достижении установленного числа импульсов срабатывает связанный со счетчиком контакт. В ПЛК существуют программные и аппаратные счетчики.

Программные счетчики реализуются с помощью программных средств и их быстродействие зависит от времени цикла ПЛК. Если период следования импульсов на входе будет значительно меньше чем время сканирования то программный счетчик может не успеть корректно их обработать. Преимуществом программных счетчиков является возможность использовать в качестве счетного любой физической вход контроллера, внутреннее реле, возможность организовывать очередность выполнения нескольких инструкций.

Высокоскоростные счетчики работают по принципу аппаратных прерываний и их быстродействие не зависит от времени сканирования ПЛК, поэтому их часто используют для подсчета импульсов с высокой частотой. Входы высокоскоростных счетчиков жестко привязаны к физическим входам контроллера.

По типу счетчики можно условно разделить на следующие группы :

Стандартные 16 бит — программный счетчик, счет только вверх, при отключении питания не сохраняют подсчитанное значение.

Энергонезависимые 16 бит — программный счетчик, счет только вверх, при сохраняют подсчитанное значение при отключении питания.

Стандартные 32 бит — программный счетчик, счет вверх или вниз, направление счета выбирается специальными реле SM. При отключении питания не сохраняют подсчитанное значение.

Энергонезависимые 32 бит — программный счетчик, счет вверх или вниз, направление счета выбирается специальными реле SM. Сохраняют подсчитанное значение при отключении питания.

Высокоскоростные 32 бит — аппаратный счетчик, направление счета определяется физическим входом ПЛК, на который приходит сигнал или специальными реле SM, быстродействие не зависит от времени сканирования программы.

Тип	Адреса	Направление счета	Диапазон
Стандартные 16 бит	C0...C99	Счет вверх	1...32767
Энергонезависимые 16 бит	C100...C199*	Счет вверх	1...32767
Стандартные 32 бит	C200...C219	Изменяемое специальными реле SM	1...2147483647
Энергонезависимые 32 бит	C220...C234*	Изменяемое специальными реле SM	1...2147483647
Высокоскоростные 32 бит	C235...C255	Изменяемое сигналами на дискретных входах и специальными реле SM	-2147483648... +2147483647

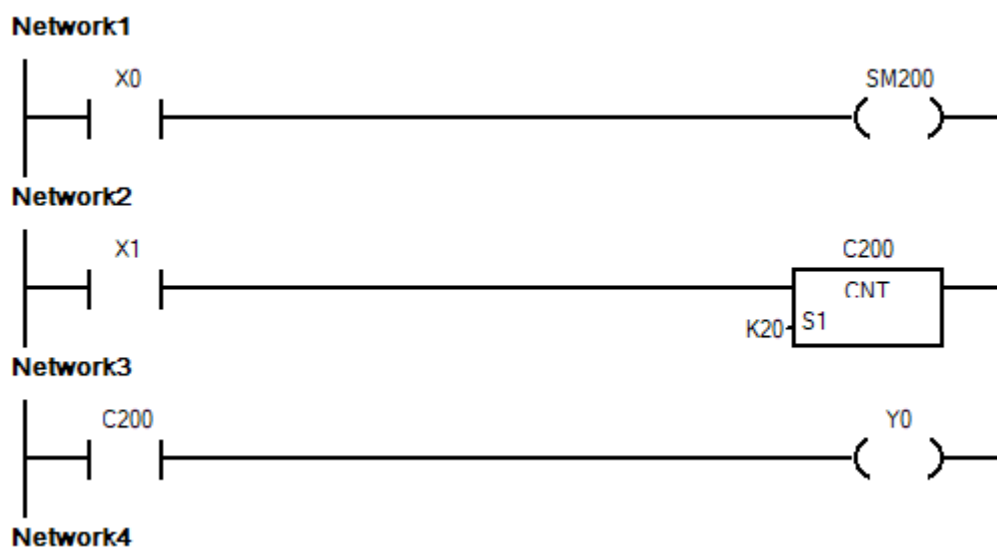
*Диапазон энергонезависимых счетчиков можно изменить.

Таблица соответствия специальных реле и счетчиков. Если SM – OFF, счет идет в прямом направлении (инкремент), если ON – в обратном (декремент).

Регистр	Функция	Регистр	Функция
SM200	C200 направление счета	SM228	C228 направление счета
SM201	C201 направление счета	SM229	C229 направление счета
SM202	C202 направление счета	SM230	C230 направление счета
SM203	C203 направление счета	SM231	C231 направление счета
SM204	C204 направление счета	SM232	C232 направление счета
SM205	C205 направление счета	SM233	C233 направление счета
SM206	C206 направление счета	SM234	C234 направление счета
SM207	C207 направление счета	SM235	C235 направление счета
SM208	C208 направление счета	SM236	C236 направление счета

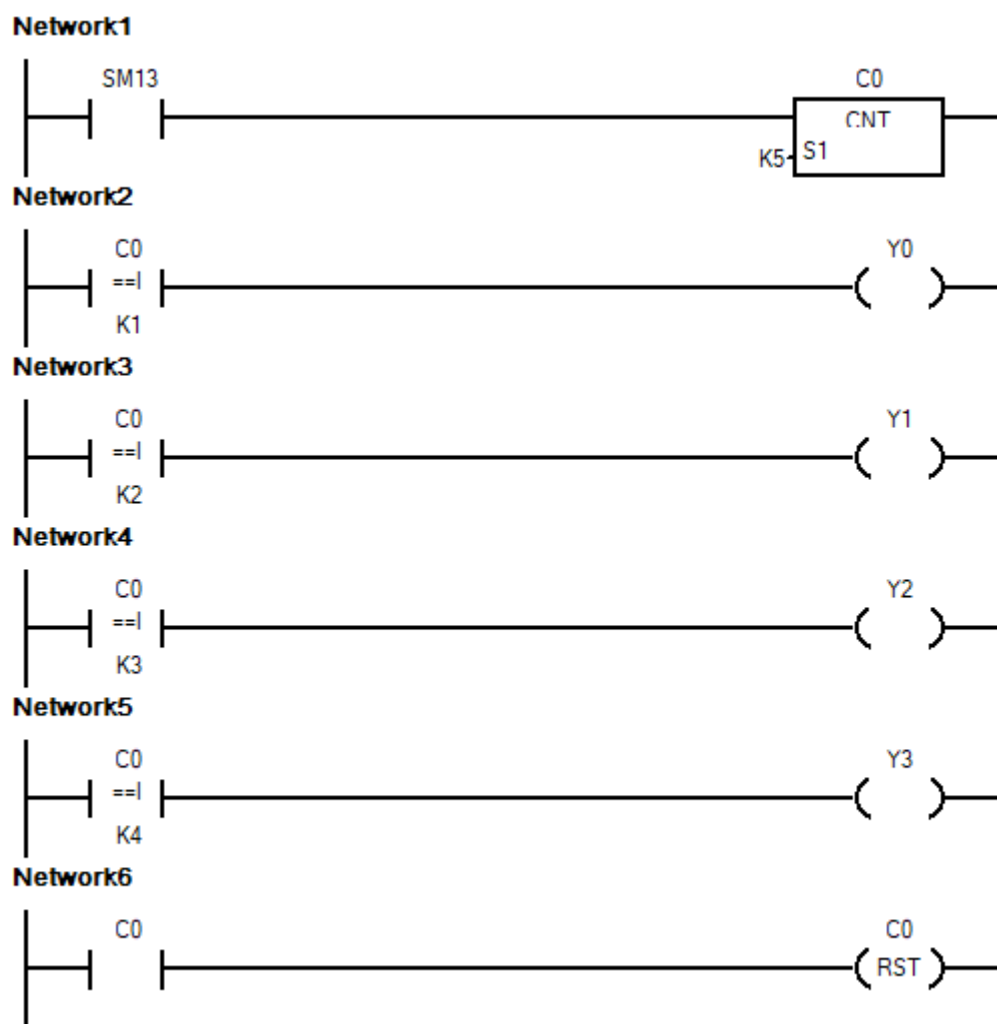
Регистр	Функция	Регистр	Функция
SM209	C209 направление счета	SM237	C237 направление счета
SM210	C210 направление счета	SM238	C238 направление счета
SM211	C211 направление счета	SM239	C239 направление счета
SM212	C212 направление счета	SM240	C240 направление счета
SM213	C213 направление счета	SM241	C241 направление счета
SM214	C214 направление счета	SM242	C242 направление счета
SM215	C215 направление счета	SM243	C243 направление счета
SM216	C216 направление счета	SM244	C244 направление счета
SM217	C217 направление счета	SM245	C245 направление счета
SM218	C218 направление счета	SM246	C246 направление счета
SM219	C219 направление счета	SM247	C247 направление счета
SM220	C220 направление счета	SM248	C248 направление счета
SM221	C221 направление счета	SM249	C249 направление счета
SM222	C222 направление счета	SM250	C250 направление счета
SM223	C223 направление счета	SM251	C251 направление счета
SM224	C224 направление счета	SM252	C252 направление счета
SM225	C225 направление счета	SM253	C253 направление счета
SM226	C226 направление счета	SM254	C254 направление счета
SM227	C227 направление счета	SM255	C255 направление счета

Примеры использования счётчиков:



Пример использования счетчика

Счетчик C200 считает импульсы со входа X1. При достижении значения 20 импульсов включится выход Y0. Вход X0 управляет направлением счета – если сигнала нет то счет идет в вверх, если X0 включен – счет идет вниз.



Пример использования счетчика

Счетчик C0 считает импульсы, которое выдает специальное реле SM13.

В сетях 2-5 происходит сравнение текущего значения с константой и при совпадении включается соответствующий выход. При достижении значения 5 происходит сброс счетчика и цикл повторяется заново. В результате исполнения программы будут поочередно включаться выходы с Y0 по Y3. Если вместо выходов использовать какие-либо инструкции или внутренние реле для включения условий выполнения определенных участков программы то с помощью счетчика можно реализовать строго определенную очередность выполнения нужных операций.

Данный подход нельзя использовать если один и тот же выход необходимо включать на разных шагах, так как это приведет к конфликту двойного использования выхода. В таких случаях рекомендуется использовать инструкции шагового управления STL.

3.11 Высокоскоростные счетчики

В ПЛК LogicOn есть возможность использовать 21 высокоскоростных счетчиков. В зависимости от функции счетчика их можно разделить на следующие группы:

Однофазный счетчик — счетчик имеет только один вход, сброс и разрешение работы реализуются в программе.

Однофазный счетчик с входом сброса — счетчик имеет один счетный вход и вход для сброса накопленного значения.

Однофазный счетчик с входами сброса и разрешения — счетчик имеет один счетный вход, один вход для сброса и один вход разрешения работы.

Двухфазный счетчик — счетчик имеет два счетных входа, по одному из которых счет идет вверх а по другому — вниз.

Двухфазный счетчик с входом сброса — счетчик имеет два счетных входа и вход для сброса накопленного значения.

Двухфазный счетчик с входами сброса и разрешения — счетчик имеет два счетных входа, один вход для сброса и один вход разрешения работы.

А/В-фазный счетчик — счетчик имеет два входа. Если сигнал на входе А опережает сигнал на входе В, то значение в счетчике увеличивается, если сигнал на входе В опережает сигнал А, то уменьшается. Такой тип счетчиков предназначен, в основном, для работы с энкодерами.

А/В-фазный счетчик с входом сброса — счетчик имеет два входа для А/В сигнала и дополнительный вход для сброса накопленного значения.

А/В-фазный счетчик с входами сброса и разрешения — счетчик имеет два входа для А/В сигнала, вход для сброса и вход для разрешения работы.

Значения всех высокоскоростных счетчиков сохраняются в энергонезависимую память.

Высокоскоростные счетчики не могут применяться с 16-ти битными инструкциями.

Адресация высокоскоростных счетчиков в зависимости от назначения приведена таблице ниже:

Счётчик		X0	X1	X2	X3	X4	X5	X6	X7
Однофазный	C235	U/D							
	C236		U/D						
	C237			U/D					
	C238				U/D				
	C239					U/D			
	C240						U/D		
Однофазный со сбросом	C241	U/D	R						
	C242		U/D	R					
	C243			U/D	R				
Однофазный со сбросом и разрешением	C244	U/D	R					S	
	C245		U/D	R					S
Двухфазный	C246	U	D						
Двухфазный со сбросом	C247	U	D	R					
	C248				U	D	R		
Двухфазный со сбросом и разрешением	C249	U	D	R				S	
	C250				U	D	R		S
А/В-фазный	C251	A	B						
А/В-фазный со сбросом	C252	A	B	R					
	C253				A	B	R		
А/В-фазный со сбросом и разрешением	C254	A	B	R				S	
	C255				A	B	R		S

U – Вход прямого счета

B – Вход фазы B

D – Вход обратного счета

R – Вход сброса

A – Вход фазы A

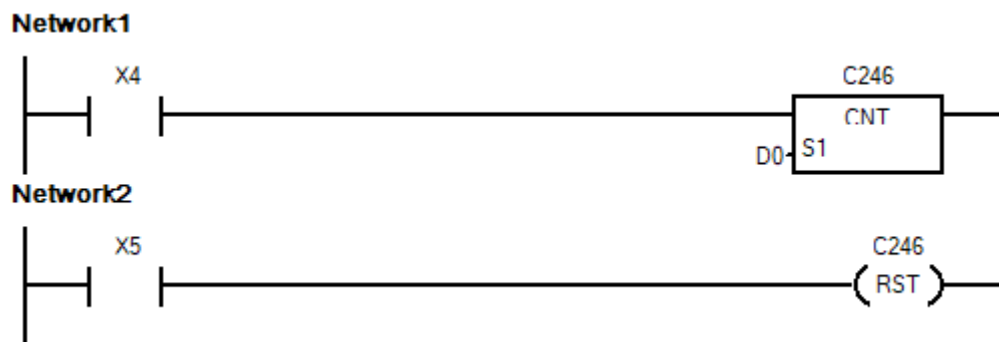
S – Вход разрешения работы

Примеры применения высокоскоростных счетчиков



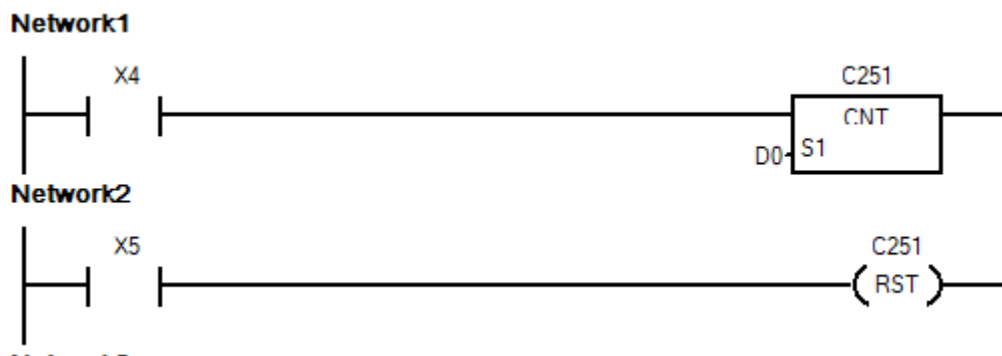
Пример применения однофазного счетчика

Когда X4 включен, счетчик C235 считает импульсы со входа X0 (см. таблицу 27). Если X3 выключен, то счет идет вверх, если включен - счет идет вниз. При включении X2 происходит сброс накопленного значения.



Пример использования двухфазного счетчика

Когда X4 включен, двухфазный счетчик считает импульсы со входов X0 и X1 (см. таблицу 27). При поступлении импульсов на X0 значение счетчика увеличивается, при поступлении импульсов на X1 — уменьшается. При включении X5 накопленное значение сбрасывается. В регистре D0 задается уставка счетчика, нужно иметь ввиду что высокоскоростные счетчики работают с 32-битными значениями, таким образом значение уставки будет находиться в D0 и D1.



Пример использования A/B-фазного счетчика

Когда X4 включен, счетчик C251 считает сигналы на входе X0(Фаза A) и X1(фаза B). При включении X5 происходит сброс накопленного значения.

Также имеется возможность использования прерывания, генерируемого высокоскоростным счетчиком при совпадении с уставкой. Подробнее в описании инструкции DHSCT.

3.12 Указатели L, P, I

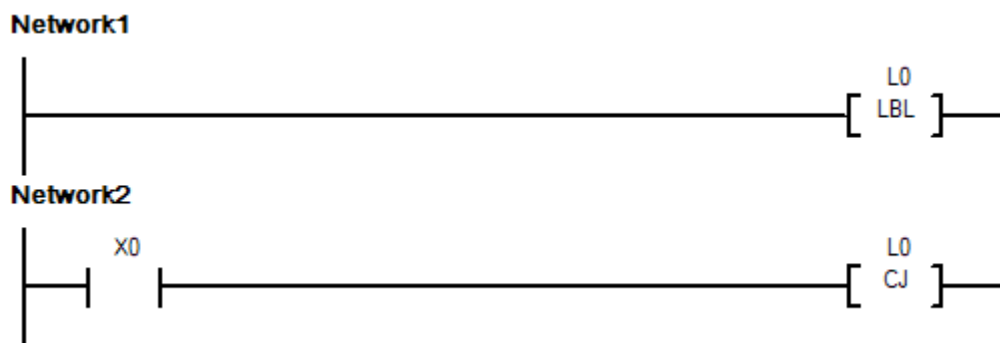
Указатели применяются для перехода между участками программы, вызова подпрограмм или переходу к подпрограмме обработки прерывания.

Указатель L – Указатель перехода к указанной метке L0- L127. Применяется совместно с инструкциями CJ (переход к метке Ln) и LBL (метка Ln).



Пример применения инструкции CJ и указателя L

Когда X1 выключен, последовательно выполняется вся программа. Когда X1 включен, участок программы между CJ и L0 (Network 2) не выполняется. Если перед включением X1 был включен X2 и, соответственно, Y1, то после включения X1 участок программы с этими элементами пропускается и Y1 останется включенным даже если сигнал на X2 изменит свое состояние на выключенное.

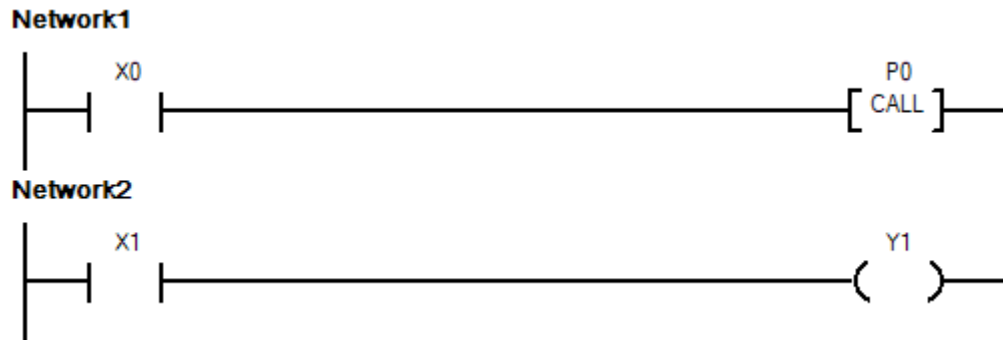


Условный переход вверх по программе

К использованию условных переходов необходимо подходить внимательно.

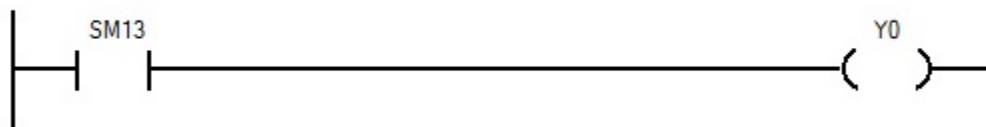
На примере выше показан условный переход вверх по программе. Если сигнал на входе X0 будет присутствовать дольше 200 мс (по умолчанию время Watchdog), то это вызовет срабатывание сторожевого таймера и ПЛК выдаст ошибку.

Указатель P – Указатель вызывает подпрограмму с соответствующим номером от P0 до P127. Используется совместно с инструкцией CALL.



Вызов подпрограммы P0 из основной программы

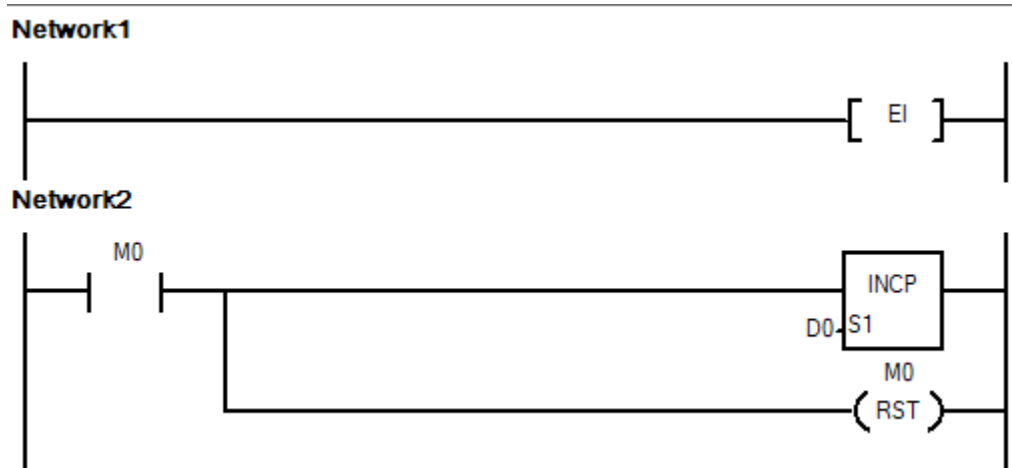
Сама подпрограмма создается в отдельном окне в меню Sub Program.



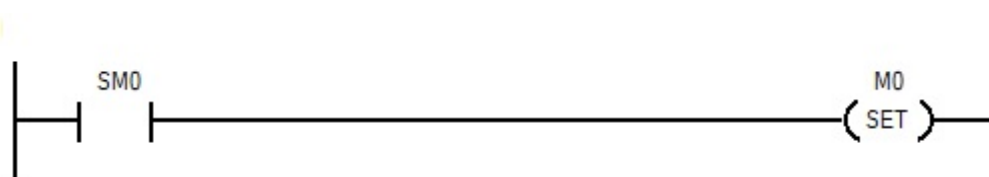
Подпрограмма P0

Когда X0 включен, будет вызвана подпрограмма P0, которая будет включать и выключать выход Y0 с периодом в 1 секунду. Когда X1 выключен подпрограмма вызываться не будет и выход Y0 останется в том же состоянии, которое установилось при последнем исполнении подпрограммы.

Указатель I – При возникновении прерывания выполнение основной программы прекращается и выполняется участок программы на которую ссылается указатель. Всего доступно 12 прерываний по входам X0-X5 для вызова прерывания по внешним сигналам и 3 прерывания по времени для вызова прерывания через равные промежутки времени. Для использования прерываний в программе необходимо их разрешить командой EI. При необходимости можно выборочно запретить определенные прерывания специальными реле SM50-SM61.

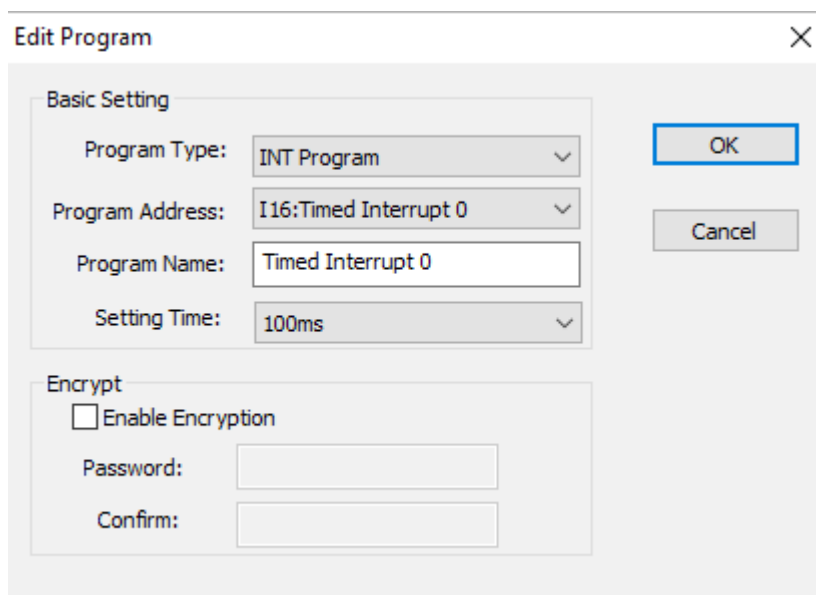


Пример использования прерываний в программе



Программа в прерывании

На примере выше используется прерывание I16 по таймеру и вызывается каждые 100 мс.



Создание подпрограммы прерывания

Когда срабатывает прерывание, в программе обработке прерывания внутреннее реле M0 устанавливается в 1. В основной программе срабатывание реле вызывает прибавление 1 к содержимому регистра D0 и сброс реле M0.

Доступные диапазоны и команды указаны в таблице ниже:

Переход по указанной метке	Вызов подпрограммы	Прерывания по входным сигналам	Прерывания по времени
<p>L0~L127</p> <p>Используются совместно с инструкциями LBL и CJ</p>	<p>P0~P127</p> <p>Используются совместно с инструкцией CALL</p>	<p>I0: Прерывание по переднему фронту на X0. I1: Прерывание по заднему фронту на X0. I2: Прерывание по переднему фронту на X1. I3: Прерывание по заднему фронту на X1. I4: Прерывание по переднему фронту на X2. I5: Прерывание по заднему фронту на X2. I6: Прерывание по переднему фронту на X3. I7: Прерывание по заднему фронту на X3. I8: Прерывание по переднему фронту на X4. I9: Прерывание по заднему фронту на X4. I10: Прерывание по переднему фронту на X5. I11: Прерывание по заднему фронту на X5.</p>	<p>I16, I17, I18</p>

3.13 Константы К, Н, F

Для выполнения операций ПЛК оперирует семью типами данных, их описание приведено в таблице ниже:

Тип данных	Применение
Десятичный (DEC)	Задание значений счетчиков и таймеров, нумерация внутренних реле (M), таймеров (T), счетчиков (C), шаговых реле (S) и т.д. Для определения десятичного значения используется символ «K».
Шестнадцатеричный (HEX)	Используется для задания констант и операндов для некоторых инструкций. Для определения шестнадцатеричного значения используется символ «H».
Двоичный (BIN)	Используется для внутренних вычислений ПЛК и хранения значений в памяти.
Восьмеричный (OCT)	Нумерация входных реле(X) и выходных реле (Y).
Двоично-десятичный (BCD)	BCD формат представляет собой форму записи вещественных чисел, при котором каждый десятичный разряд числа записывается в виде его четырехбитного двоичного кода. Используется в основном для работы с внешними DIP-переключателями и сегментными индикаторами.
Двоичное число с плавающей точкой	Используется для внутренних вычислений ПЛК и при выполнении определенных инструкций. Для определения числа с плавающей значения используется символ «F».
Десятичное число с плавающей точкой	Десятичное число с плавающей точкой в основном используется только для мониторинга исполнения программы.

Константа К - символ [K] обозначает десятичную систему счисления. В основном используется для задания значения таймера, счетчика, для передачи значения прикладным инструкциям. Диапазон значений для 16-битной инструкции -32 768...32 767. Для 32-битной 2 147 483 648...2 147 483 647.

Константа Н – символ [H] обозначает шестнадцатеричную систему счисления. В основном используется для передачи значения прикладным инструкциям. Диапазон значений для 16-битной инструкции 0000 – FFFF. Для 32-битной инструкции 0000 0000 – FFFF FFFF.

Константа F - символ [F] обозначает число с плавающей точкой. В основном используется для передачи значения прикладным инструкциям.

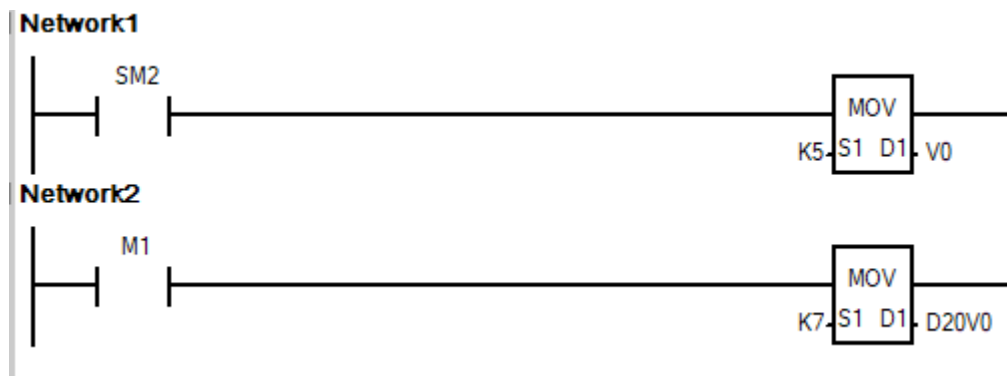
3.13 Индексные регистры V, Z

Индексные регистры используются для возможности косвенной адресации. Косвенная адресация может использоваться, например, в циклах, для операций с данными в нескольких последовательных регистрах. Если в программе используется индексация, то адрес элемента состоит из суммы значения адреса и содержимого адресного регистра. Например, если значение регистра Z0 = 10, то запись вида D5Z0 означает обращение к регистру с номером (D5+10), т.е. D15. Всего возможно использовать 16 индексных регистров V0-V7 и Z0-Z7. Индексировать сами индексные регистры нельзя.

Если индексный регистр используется совместно с элементами с десятичной адресацией (M, SM, S, T, C, D, SD, KnM, KnS, L, P) или десятичными константами K, то содержимое индексных регистров преобразуется в десятичный формат.

Если индексный регистр используется совместно с элементами с восьмеричной адресацией (X, Y, KnX, KnY) то содержимое регистра преобразуется в восьмеричный формат. Например, если в индексном регистре V1 находится значение K8 и индекс применяется ко входам X0V1, то константа 8 будет преобразована в восьмеричный формат и команда будет применена не к X8 а к X10.

Если индексный регистр используется совместно с шестнадцатеричной константой H, то содержимое регистра преобразуется в шестнадцатеричный формат.

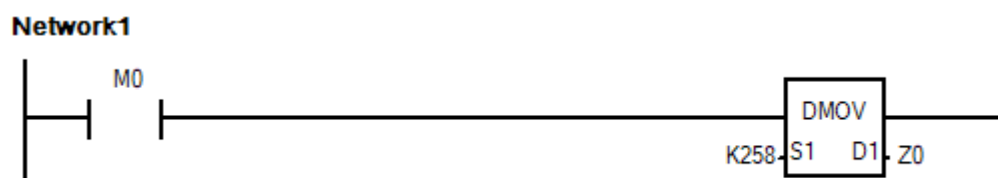


Пример косвенной адресации

На примере выше в регистр V0 помещается константа 5. При включении M1 константа 7 помещается в регистр D25 (D20+V0).

Индексные регистры – 16-ти битные. При необходимости хранения 32-х битного значения необходимо использовать пары регистров V-Z – V0 и Z0, V1 и V2 и т.д. В такой паре регистры V содержат старшие 16 бит, в Z- младшие.

Для переноса значения в пары регистров V и Z необходимо использовать только 32-х битные команды, даже если значение не выходит за граница 16-ти битного числа.



Перенос значения в пару регистров Z0 и V0

4 ПРИКЛАДНЫЕ ИНСТРУКЦИИ

В данном разделе приведено описание инструкций, доступных для программирования ПЛК LogicOn.

Все инструкции можно разделить на несколько категорий, в зависимости от типа данных, с которыми они оперируют, а также в зависимости от способа выполнения инструкции.

Различия по типу данных:

- инструкции для работы с **целыми числами**;
- инструкции для работы с **вещественными числами** (с плавающей точкой);

В свою очередь, инструкции для работы с целыми числами можно разделить на две категории:

- **16-битные**: для работы с целыми значениями в диапазоне -32 768...+32 767;
- **32-битные**: для работы с целыми значениями в диапазоне -2 147 483 648...+2 147 483 647;

32-битные инструкции используют для своих данных 2 регистра типа D. Например, если выводу такой инструкции назначить регистр D0, то автоматически будет задействован следующий регистр D1.

Все инструкции для работы с числами с плавающей точкой являются 32-битными инструкциями.

Дополнительно, инструкции можно разделить на категории по способу выполнения:

- **Непрерывного выполнения**: инструкции выполняются постоянно, пока присутствует разрешающий сигнал;
- **Импульсного выполнения**: инструкции выполняются один раз, по переднему фронту разрешающего сигнала;

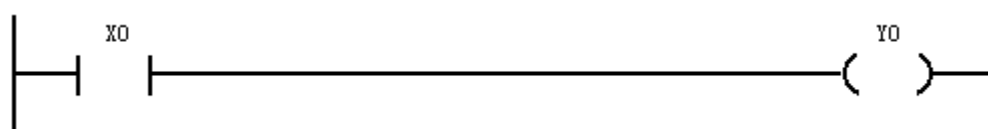
4.1 Базовые инструкции (Basic Inst)

- LD – Нормально-открытый контакт
- LDI – Нормально-закрытый контакт
- OUT – Выходной контакт
- LDP – Детектор переднего фронта
- LDF – Детектор заднего фронта
- MC, MCR – Мастер-контроль
- NOT – Инверсия сигнала
- PLS – Импульс по переднему фронту
- PLF – Импульс по заднему фронту
- SET – Установка включенного состояния
- RST – Сброс включенного состояния
- NOP – Пропуск такта
- OUTD – Мгновенное обновление выхода
- ALT, ALTP – Инверсия состояния

[LD]

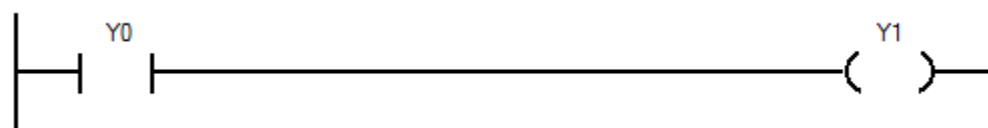
Инструкция	LD
Функция	Нормально-открытый контакт
Доступные элементы	X, Y, M, SM, S, T, C
Описание	Нормально-открытый контакт используется для начала логических цепочек.

Если на входе X0 присутствует сигнал, то выход Y0 включен.



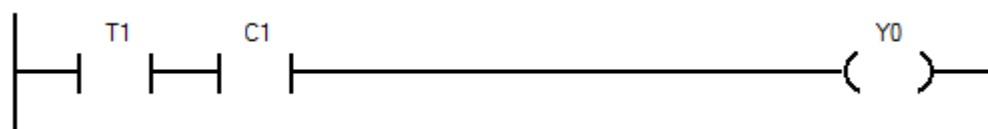
Данная инструкция применима не только для физических входов контроллера, но также и для выходов, внутренних реле, таймеров, счетчиков и т.п.

В следующем примере выход Y1 включится только если до этого где-то в программе был включен выход Y0.

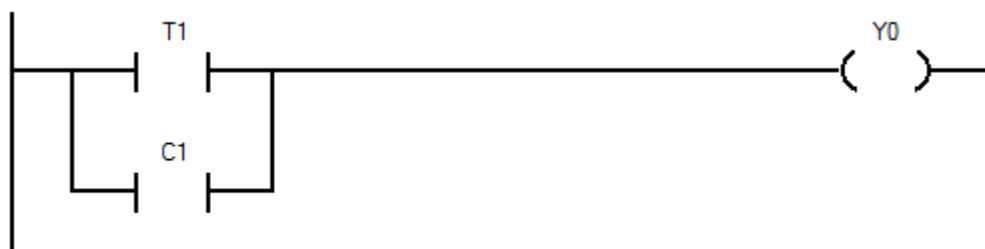


Возможно последовательное, параллельное и смешанное соединение контактов.

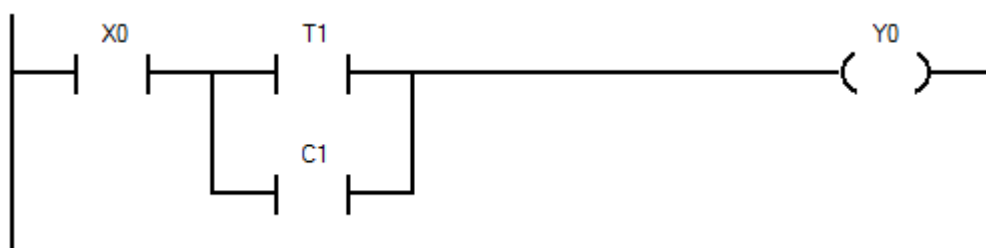
Последовательное соединение: в следующем примере выход Y0 включится только когда будут выполнены оба условия: достигнуты уставки таймера T1 и счетчика C1.



Параллельное соединение: в следующем примере выход Y0 включится только когда будет выполнено одно из двух условий: достигнута либо уставка таймера T1, либо уставка счетчика C1.



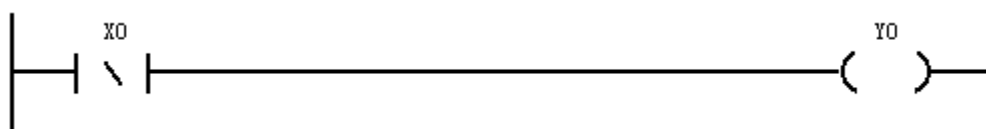
Смешанное соединение: в следующем примере выход Y0 включится только когда будет замкнут вход X0, при этом дополнительно должно быть выполнено одно из двух условий: достигнута либо уставка таймера T1, либо уставка счетчика C1.



[LDI]

Инструкция	LDI
Функция	Нормально-закрытый контакт
Доступные элементы	X, Y, M, SM, S, T, C
Описание	Нормально-закрытый контакт используется для начала логических цепочек.

Если на входе X0 присутствует сигнал, то выход Y0 выключен.

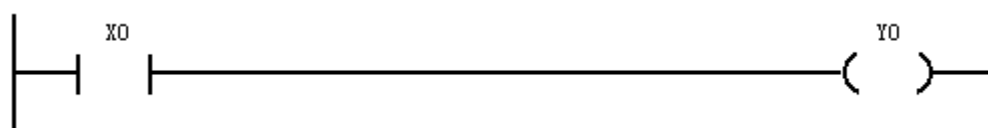


Для данной инструкции применимы свойства, ранее описанные для инструкции LD (нормально-открытый контакт).

[OUT]

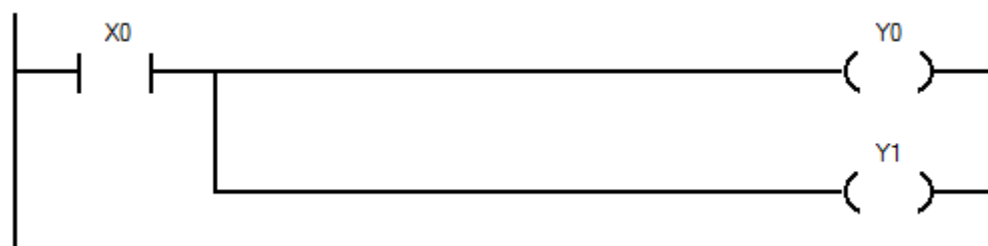
Инструкция	OUT
Функция	Выходной контакт
Доступные элементы	Y, M, SM, S
Описание	Используется для включения или отключения выхода.

Выход Y0 включен, если на входе X0 присутствует сигнал.



Возможно параллельное соединение выходных контактов.
Последовательное соединение невозможно, т.к. не имеет смысла.

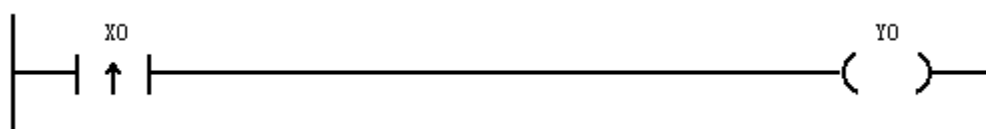
В следующем примере при замыкании контакта X0 одновременно включатся оба выхода Y0 и Y1.



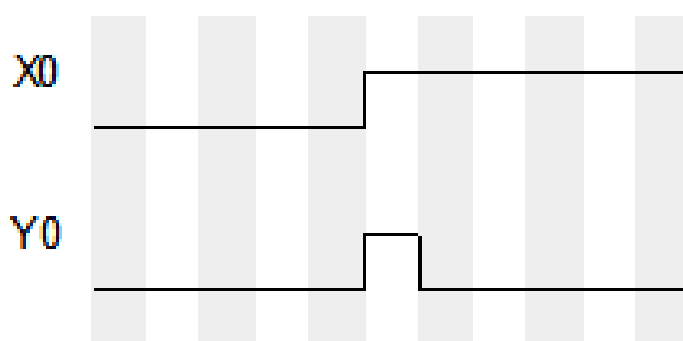
[LDP]

Инструкция	LDP
Функция	Детектор переднего фронта
Доступные элементы	X, Y, M, SM, S, T, C
Описание	Обнаружение переднего фронта импульса

Выход Y0 срабатывает на один цикл ПЛК, когда X0 изменяет свое состояние с выключенного на включенный.



Временная диаграмма работы инструкции. Одна клетка соответствует одному циклу ПЛК.

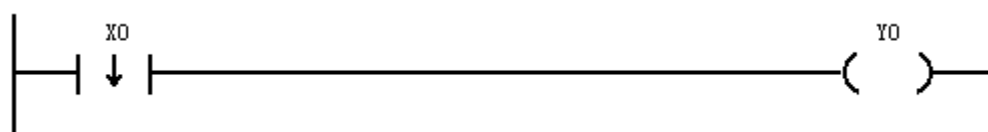


Для данной инструкции применимы свойства, ранее описанные для инструкции LD (нормально-открытый контакт).

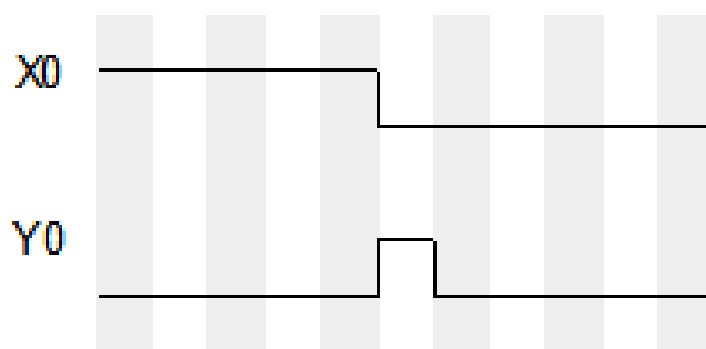
[LDF]

Инструкция	LDF
Функция	Детектор заднего фронта
Доступные элементы	X, Y, M, SM, S, T, C
Описание	Обнаружение заднего фронта импульса

Выход Y0 срабатывает на один цикл ПЛК, когда X0 изменяет свое состояние с включенного на выключенный.



Временная диаграмма работы инструкции. Одна клетка соответствует одному циклу ПЛК.

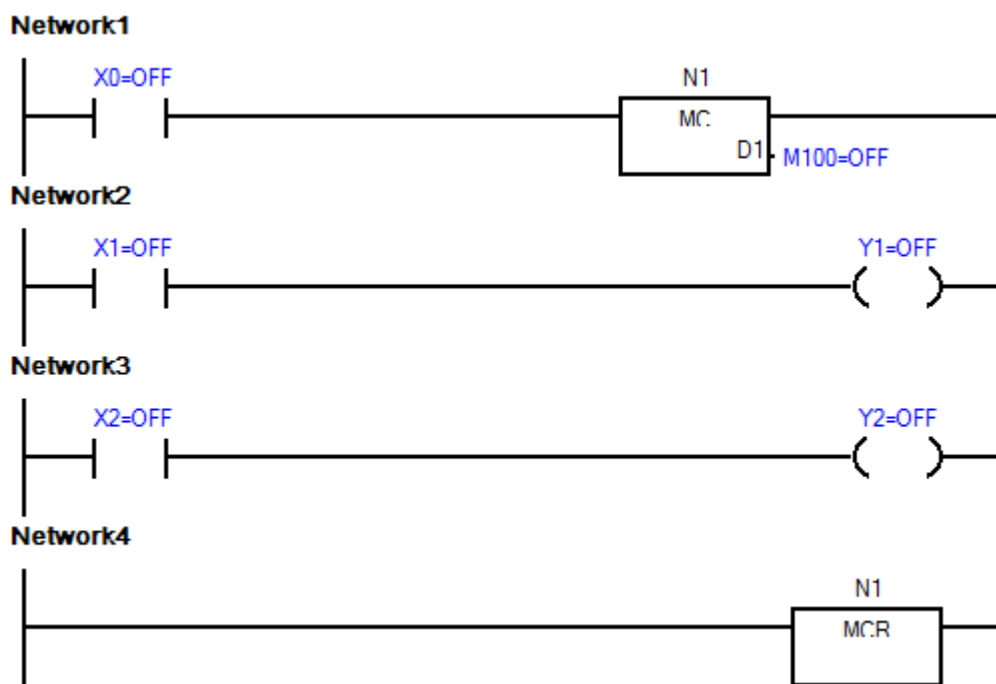


Для данной инструкции применимы свойства, ранее описанные для инструкции LD (нормально-открытый контакт).

[MC], [MCR]

Инструкция	MC, MCR
Функция	Мастер контроль
Идентификаторы	N0...N7
Доступные выходы	Сигнализация работы участка кода D1 – Y, M
Описание	Включение / отключение участка программы.

Когда X0 включен, то участок кода между инструкциями MC и MCR выполняется. Когда X0 выключен, то участок программы между инструкциями MC и MCR пропускается и выполняться не будет.



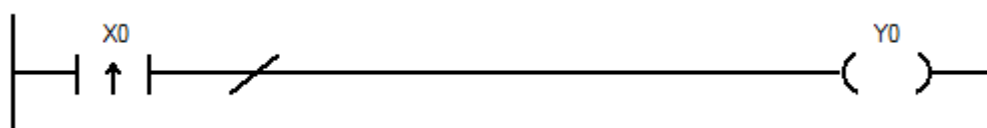
Возможно использование до 8 таких инструкций в программе.

Выход D1 – сигнализация выполнения участка кода. В примере выше в регистре M100 дублируется состояние входа X0.

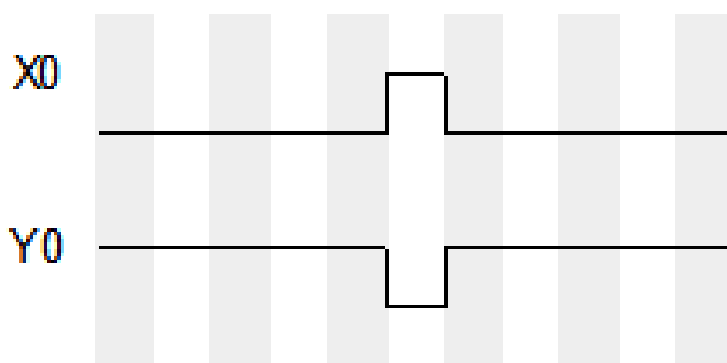
[NOT]

Инструкция	NOT
Функция	Инверсия сигнала
Описание	Инвертирует результат операции.

Когда X0 выключен, выход Y0 будет все время включен. При переходе X0 из выключенного во включенное состояние формируется импульс, который затем инвертируется. Поэтому Y0 выключится на один цикл работы ПЛК.



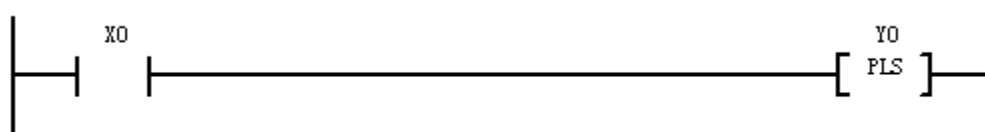
Временная диаграмма работы вышеописанного примера. Одна клетка соответствует одному циклу ПЛК.



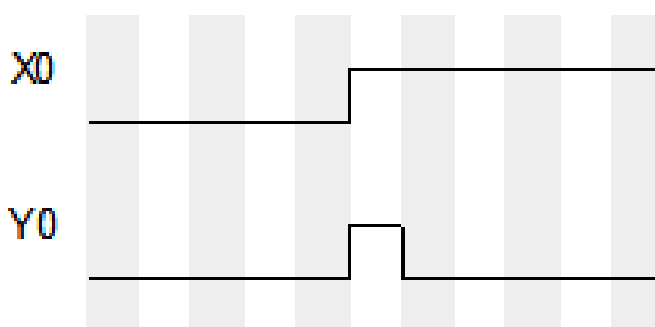
[PLS]

Инструкция	PLS
Функция	Импульс по переднему фронту
Доступные элементы	Y, M
Описание	Создание импульса по переднему фронту.

При поступлении сигнала на вход X0 выход Y0 включится на время равное одному циклу ПЛК в независимости от продолжительности сигнала на входе X0.



Временная диаграмма работы инструкции. Одна клетка соответствует одному циклу ПЛК.

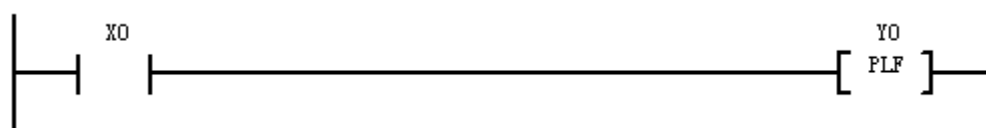


Применение данной инструкции позволяет добиться аналогичного результата с инструкцией LDP (детектор переднего фронта).

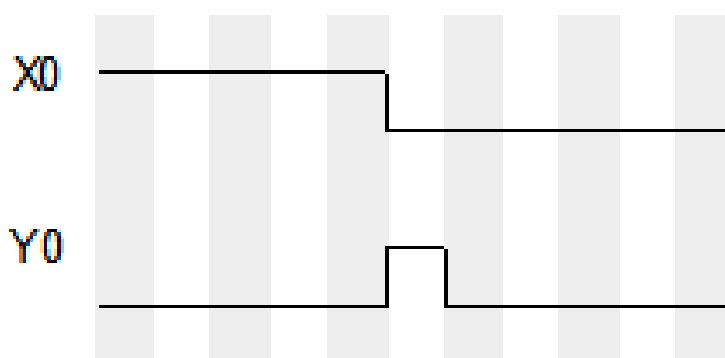
[PLF]

Инструкция	PLF
Функция	Импульс по заднему фронту
Доступные элементы	Y, M
Описание	Создание импульса по заднему фронту.

При изменении состояния X0 с ON на OFF выход Y0 включится на время равное одному циклу сканирования ПЛК.



Временная диаграмма работы инструкции. Одна клетка соответствует одному циклу ПЛК.

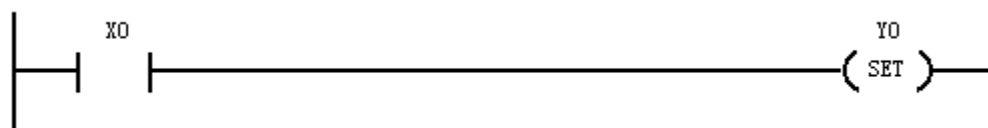


Применение данной инструкции позволяет добиться аналогичного результата с инструкцией LDF (детектор заднего фронта).

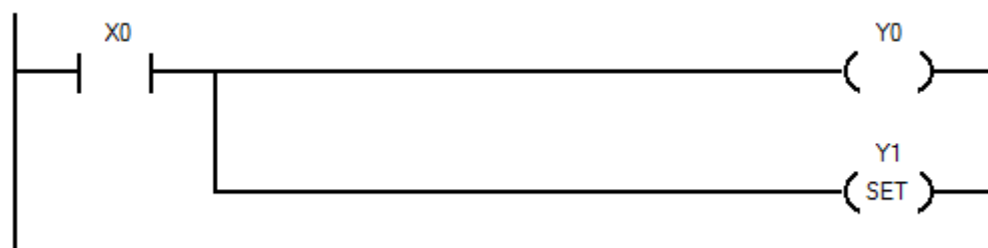
[SET]

Инструкция	SET
Функция	Установка включенного состояния
Доступные элементы	Y, M, SM, S
Описание	Устанавливает элемент в состояние ON

При включении X0 выход Y0 включиться и останется в таком состоянии даже если X0 будет выключен. Можно сказать, что данная инструкция реализует работу триггера.

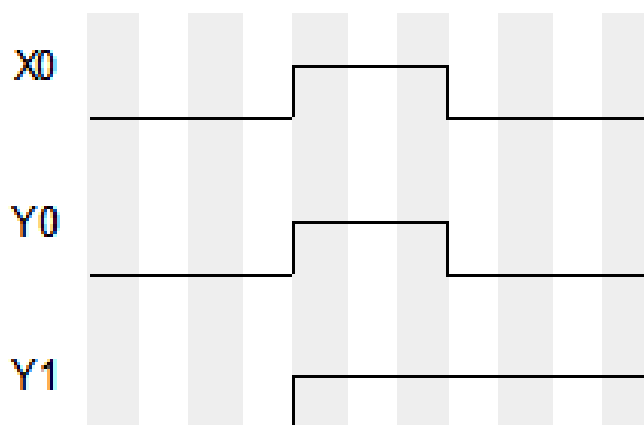


Следующий пример демонстрирует разницу в использовании инструкций OUT и SET.



При включении контакта X0 происходит включение выходов Y0 и Y1. При выключении контакта X0 происходит выключение Y0, при этом Y1 остается включенным.

Временная диаграмма данного примера представлена ниже. Одна клетка соответствует одному циклу ПЛК.

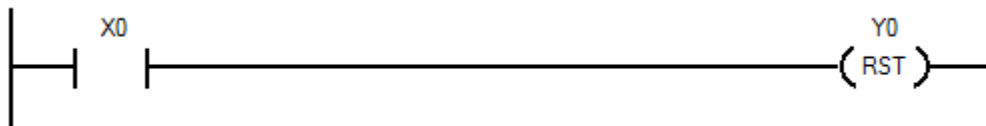


Для сброса Y0 необходимо использовать инструкцию RST (сброс включенного состояния).

[RST]

Инструкция	RST
Функция	Сброс включенного состояния
Доступные элементы	Y, M, SM, S
Описание	Устанавливает элемент в состояние OFF

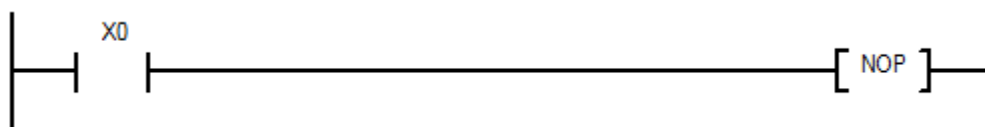
При включении X1 выход Y1 выключиться и останется в таком состоянии даже если X0 будет выключен.



Данная инструкция необходима для сброса регистров, установленных инструкцией SET (Установка включенного состояния).

[NOP]

Инструкция	NOP
Функция	Пропуск такта
Доступные элементы	Нет
Описание	Пустая строка в программе

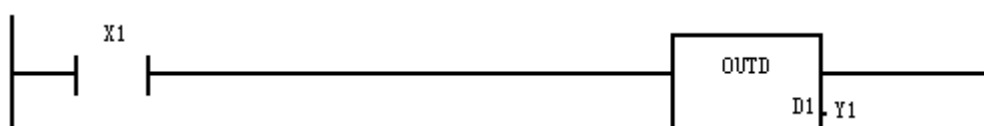


Инструкция не выполняет никаких действий. Применяется при удалении действующих инструкций для сохранения длины программы или последующей замены на какую-либо инструкцию.

[OUTD]

Инструкция	OUTD
Функция	Мгновенное обновление выхода
Доступные элементы	Физический выход D1 – Y
Описание	ПЛК обновляет состояние выходов в конце цикла. При использовании данной инструкции обновление произойдет немедленно, независимо от времени цикла.

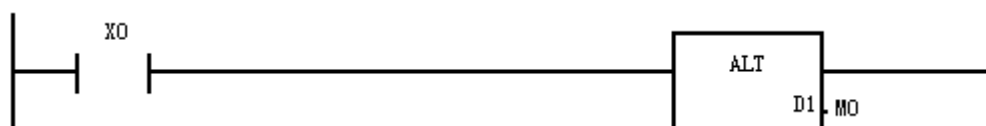
При включении входа X1 выход Y1 немедленно обновляет свое состояние, не дожидаясь окончания цикла ПЛК.



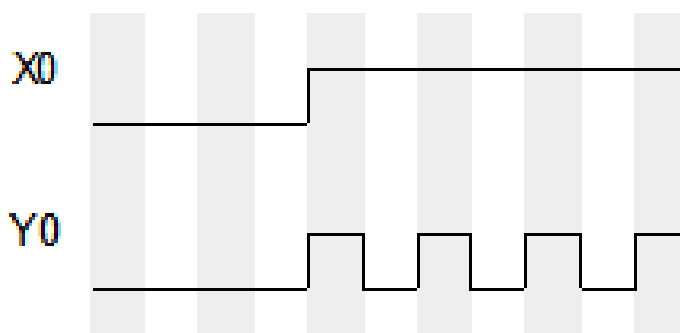
[ALT]

Инструкция	ALT
Функция	Изменение состояния
Доступные элементы	Изменяемый регистр D1 – Y, M, SM, S
Описание	При каждом срабатывании инструкции соответствующий элемент изменят свое состояние на противоположное.

При каждом срабатывании X0 внутреннее реле M0 будет менять свое состояние на противоположное.



Временная диаграмма данного примера представлена ниже. Одна клетка соответствует одному циклу ПЛК.



4.2 Инструкции пошагового управления (Step Inst)

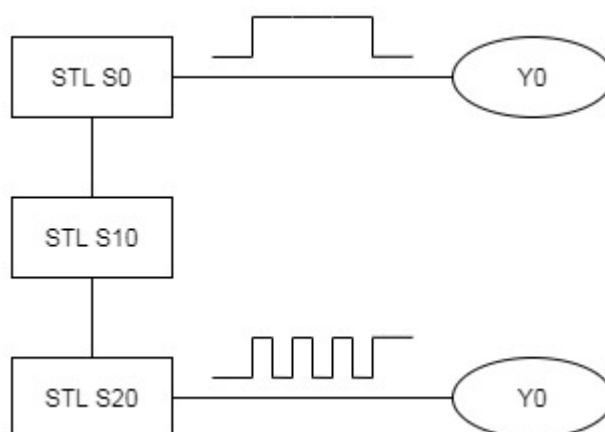
Инструкции пошагового управления

- STL – Запуск пошагового управления
- RET – Завершение пошагового выполнения

Пошаговое управление – достаточно простой, но в то же время очень мощный инструмент для организации программы на языке релейных диаграмм. Применение данных инструкций позволяет упростить процесс написания, отладки и последующего понимания работы программы. Выполнение STL программы происходит аналогично выполнению блок-схемы.

Использование инструкций STL во многом напоминает использование языка SFC. Основная идея заключается в том, что каждая инструкции STL создает изолированный участок кода, на который не влияют другие инструкции STL. Таким образом решается проблема конфликтов разных условий на одни и те же выходы.

Чтобы лучше понять смысл STL инструкций, можно рассмотреть программу, в которой требуется управлять одним выходом из нескольких мест программы.



Если не использовать пошаговое управление, то выход Y0 будет срабатывать только по самому последнему из условий (подробнее «Основы языка LD»).

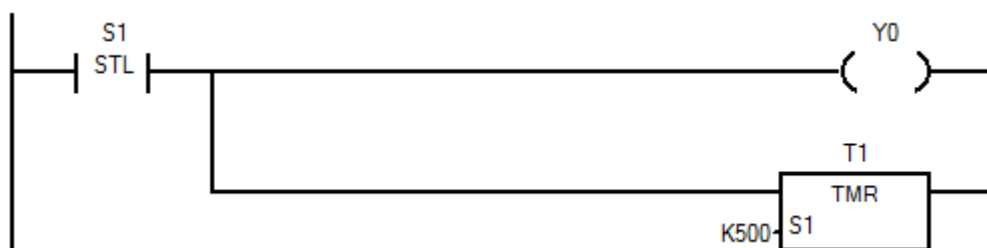
При использовании инструкций STL, выполняются только условия, которые активны в текущем шаге, а остальные шаги игнорируются.

Другими словами, когда активен шаг STL S0, условия из шага STL S20 никаким образом на выход Y0 влиять не будут.

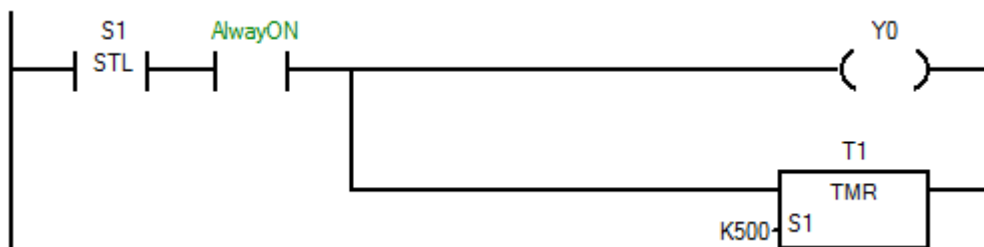
Подробный пример использования инструкций STL будет показан в программе «Управление светофором» (см. Приложение №2).

При использовании инструкций STL необходимо руководствоваться следующими правилами:

- Активация следующего шага происходит с помощью инструкции SET. При переходе на следующий шаг, сбрасывать предыдущий шаг не нужно – происходит автоматический сброс;
- Активный шаг STL работает до тех пор, пока не будет активирован другой шаг. Если следующий шаг активирован не будет, то текущий шаг будет работать всё время;
- Нельзя подключать выходы и инструкции сразу после команды STL. Поведение выходов и инструкций в таком случае непредсказуемо, т. к. аналогично прямому подключению выхода к шине питания. Рекомендуется устанавливать системный регистр SM0 сразу после инструкции STL, и уже от него делать все ответвления (см. пример ниже);

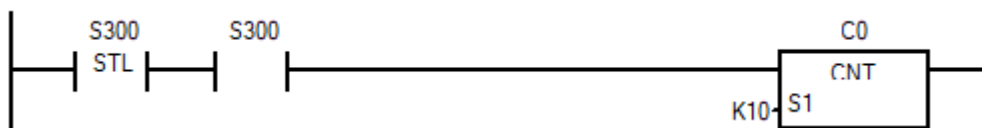


Поведение непредсказуемо, такая организация не рекомендуется



Правильная организация кода с инструкцией STL

- Возможно одновременное исполнение двух и более инструкций STL. При одновременном исполнении двух и более инструкций необходимо обращать внимание на возможный конфликт условий срабатывания (см раздел «Основы языка LD»);
- При переходе из одного шага в другой, возможно исполнение двух шагов STL в пределах одного цикла ПЛК. Поэтому следует избегать использование одного таймера в двух последовательных шагах;
- Следует обращать внимание на использование счётчиков и импульсных инструкций в шагах STL. Такого рода инструкции будут срабатывать только один раз, при первом входе в шаг STL. Чтобы импульсные инструкции срабатывали каждый раз при входе в шаг, рекомендуется дополнительно использовать контакт типа S.

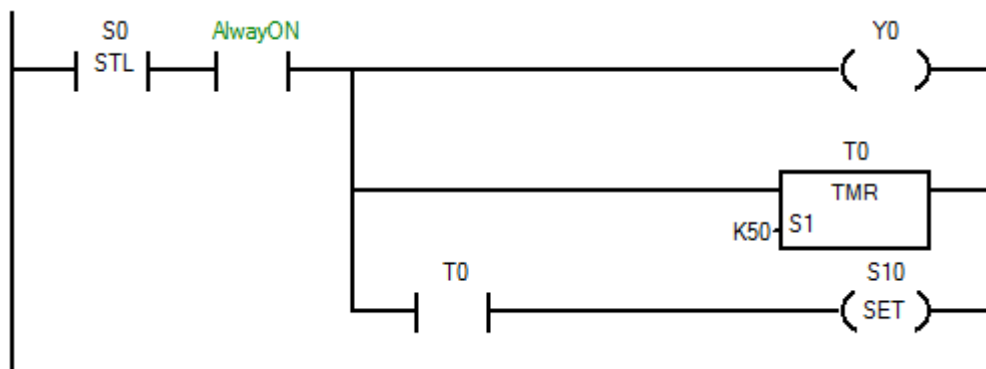


- Высокоскоростные счётчики не работают в инструкциях шагового управления, их следует выносить за пределы инструкций STL;
- Последний STL шаг всегда должен заканчиваться инструкцией RET;

[STL]

Инструкция	STL
Функция	Запуск пошагового управления
Доступные элементы	S
Описание	Команда запускает режим пошагового выполнения.

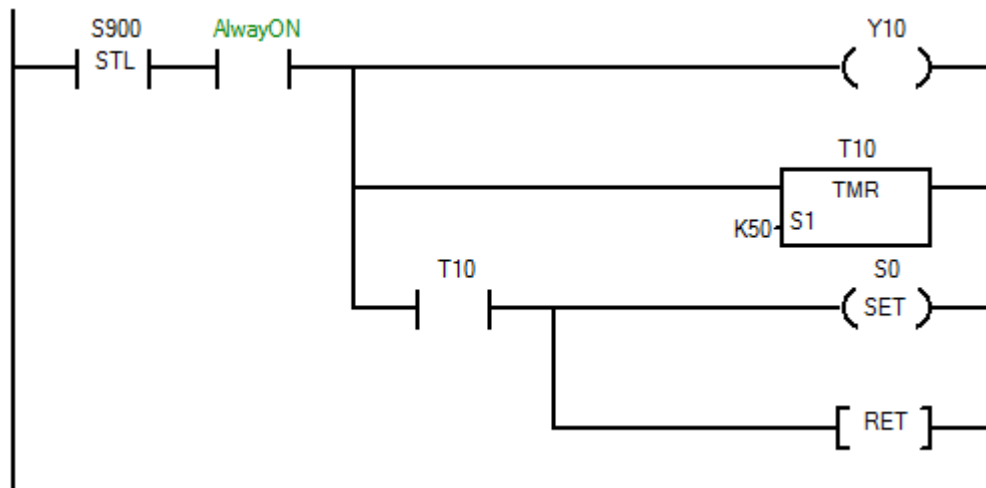
Когда S0 включен, выход Y0 включен и происходит отсчет таймера T0. По срабатыванию таймера управление передается шагу S10. Шаг S0 сбрасывается автоматически.



[RET]

Инструкция	RET
Функция	Завершение пошагового выполнения
Доступные элементы	нет
Описание	Команда завершает режим пошагового выполнения.

На последнем шаге STL программы происходит передача управления нулевому шагу. Инструкция RET необходима для правильной работы компилятора.



4.3 Инструкции управления процессом выполнения программы (Program Process Inst)

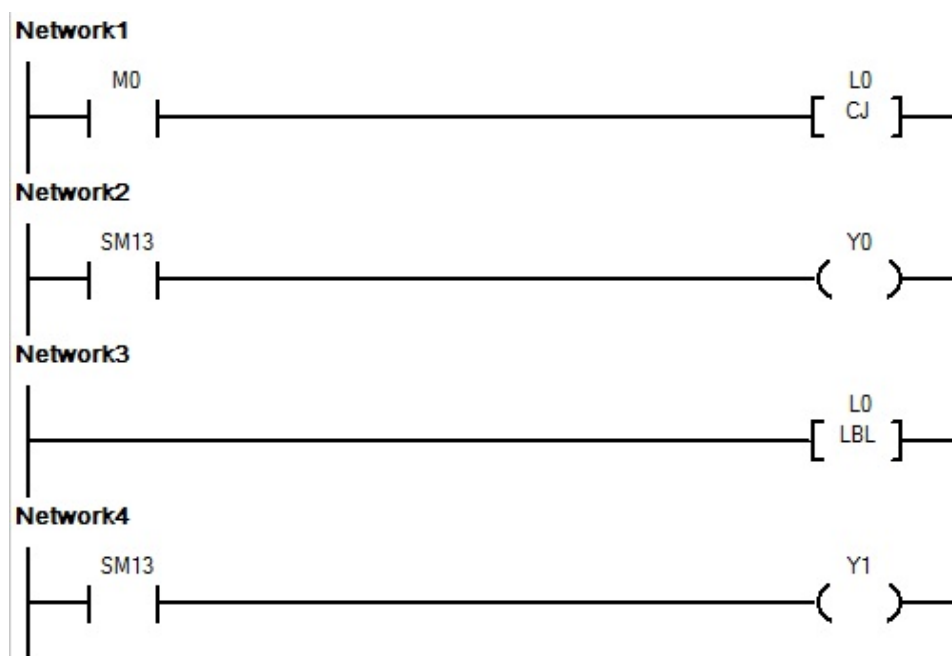
- LBL / CJ – Переход по метке
- CALL – Вызов подпрограммы
- WDT – Вызов сторожевого таймера
- EI / DI – Разрешение / запрет прерываний
- FOR / NEXT – Организация циклов

[LBL]

Инструкция	LBL		
Функция	Метка		
Доступные элементы	<table border="1"><tr><td>L0-L127</td><td>Номер метки</td></tr></table>	L0-L127	Номер метки
L0-L127	Номер метки		
Описание	Обозначение сети меткой для возможности перехода к указанному участку программы.		

[CJ]

Инструкция	CJ	
Функция	Переход по указанной метке	
Доступные элементы		
	L0-L127	Номер метки
Описание	При выполнении инструкции происходит переход к указанной метке.	



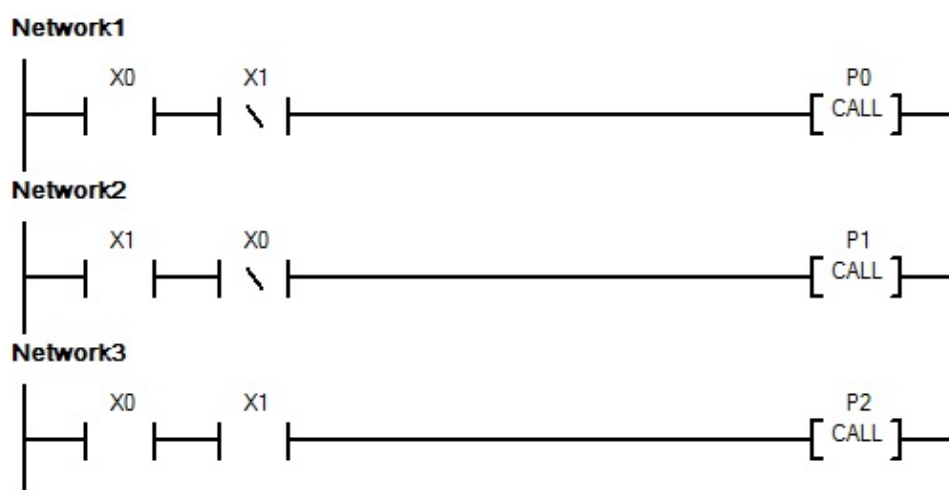
Когда M0 выключен, на выходах Y0 и Y1 будут выдаваться импульсы с периодом в 1 секунду. При включении M0 произойдет условный переход к метке L0 минувя сеть 2, в результате чего пока M0 включен импульсы на выходе Y0 выдаваться не будут.

К использованию условных переходов следует подходить внимательно, так как нарушается последовательность исполнения программы и повышается вероятность допущения ошибок при программировании, программа становится сложнее для понимания. При этом любой алгоритм, где применяются условные переходы может быть реализован без их использования, с полным сохранением функционала но, возможно, с некоторыми потерями во времени исполнения программы.

[CALL / CALLP]

Инструкция	CALL (16-ти битная инструкция) CALLP (16-ти битная инструкция, импульсное исполнение)		
Функция	Вызов подпрограммы		
Доступные элементы	<table border="1"> <tr> <td>P0-P127</td> <td>Номер подпрограммы</td> </tr> </table>	P0-P127	Номер подпрограммы
P0-P127	Номер подпрограммы		
Описание	<p>При выполнении инструкции происходит вызов подпрограммы с указанным номером.</p> <p>Активированные в процессе работы подпрограммы элементы остаются включенными до следующего вызова подпрограммы.</p> <p>Если в подпрограмме необходимо использовать таймер, то необходимо выбирать таймеры T192-T199.</p>		

В зависимости от состояния X0 и X1 происходит вызов и исполнение одной из подпрограмм. Если X0 замкнут а X1 разомкнут, то вызывается подпрограмма P0. Если X1 замкнут а X0 разомкнут, то вызывается подпрограмма P1. Если замкнуты X0 и X1, то вызывается подпрограмма P2. Если X0 и X1 разомкнуты, то вызов подпрограмм не происходит.



[WDT / WDTP]

Инструкция	WDT (16-ти битная инструкция) WDTP (16-ти битная инструкция, импульсное исполнение)
Функция	Сброс сторожевого таймера
Доступные элементы	Нет
Описание	Если время цикла превысит значение заданное в регистре SD0 (по умолчанию 200 мс) ПЛК прекратит выполнение программы, выключит все выходы и выдаст сообщение об ошибке. Инструкция WDT сбрасывает накопленное значение времени сторожевого таймера.



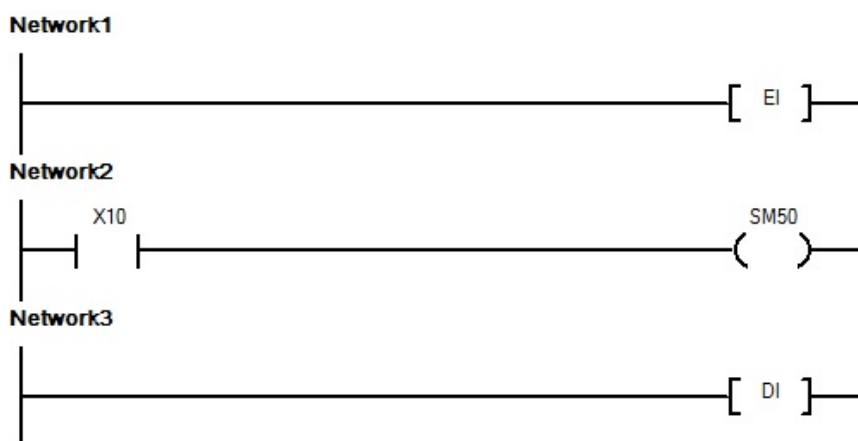
На приведенном примере показан пример программы которая выполняется за 320 миллисекунд. При обычной работе такое время цикла вызвало бы срабатывание сторожевого таймера. Использование инструкции WDT в середине программы позволяет ПЛК выполнить программу.

[EI]

Инструкция	EI
Функция	Разрешение прерываний
Доступные элементы	Нет
Описание	При исполнении данной инструкции разрешаются прерывания. По умолчанию прерывания запрещены, чтобы использовать функционал прерываний по входам или таймеру прерывания необходимо разрешить. Инструкция разрешает все прерывания, если одновременно будет вызвано несколько прерываний, то сначала обрабатывается прерывание с наименьшим номером.

[DI]

Инструкция	DI
Функция	Запрет прерываний
Доступные элементы	Нет
Описание	При исполнении данной инструкции все прерывания запрещаются. Если после разрешения прерываний инструкцией EI необходимо запретить только некоторые из прерываний то необходимо использовать специальные реле SM50-SM58 (см. описание специальных реле).

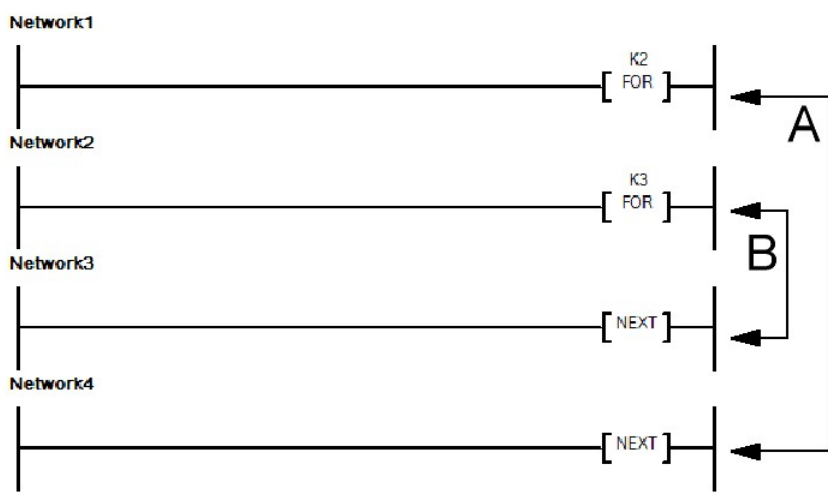


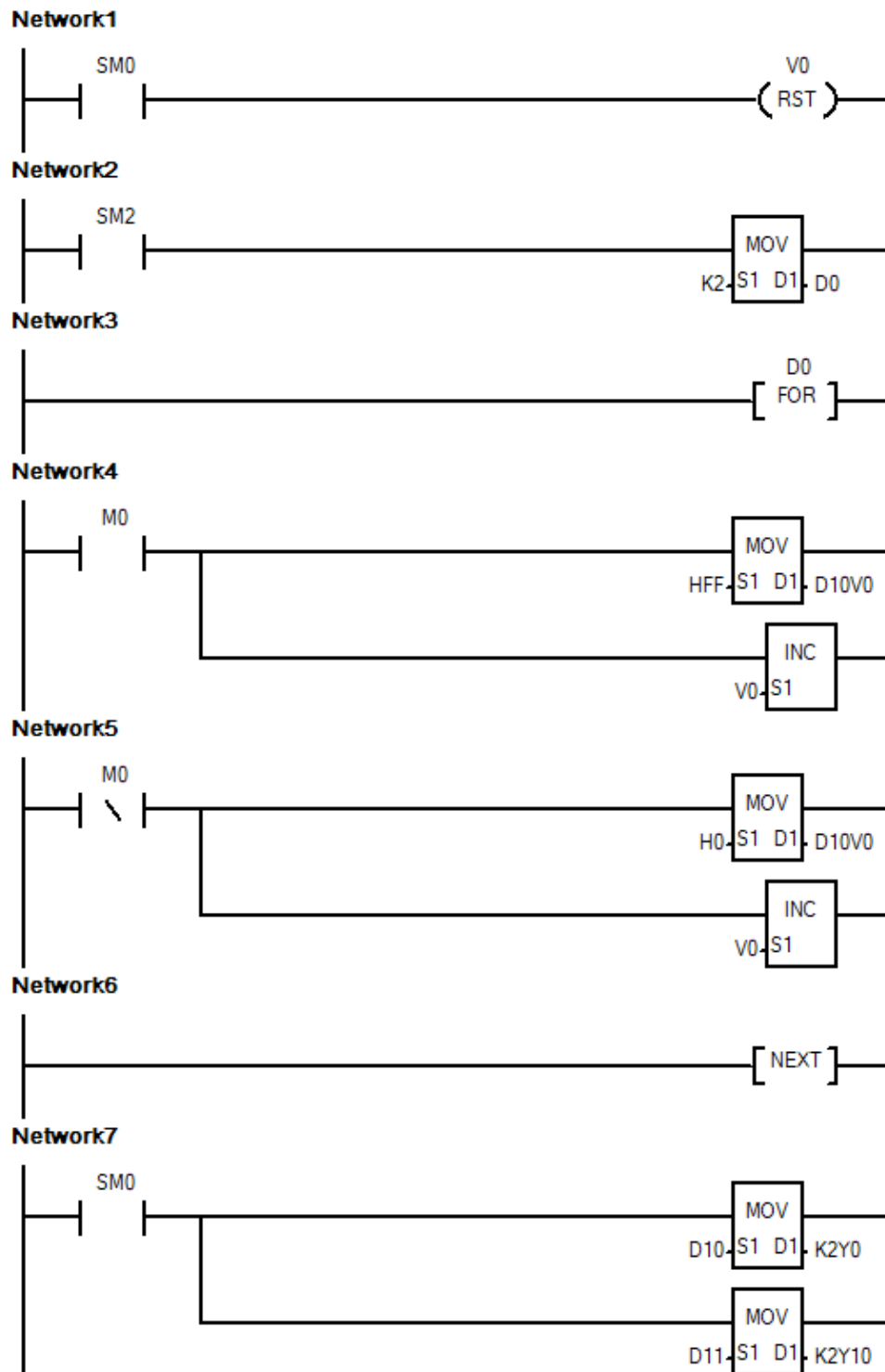
Если сигнал прерывания по входу X0 приходит во время выполнения программы между EI и DI инструкций, то происходит вызов подпрограммы обработки прерывания с номером I0. После обработки программы прерывания ПЛК

вернется к исполнению основной программы. Если X10 включен, то включается специальное реле SM50, которое запрещает вызов прерывания по входу X0 и соответственно, программа прерывания I0 вызвана не будет, но прерывания по остальным входам и таймерам останутся включенными.

Так как при исполнении программы прерывания ПЛК прекращает исполнение основной программы, программа обработки прерывания должна выполняться максимально быстро. Не следует делать подпрограмму обработки прерываний слишком большой или выполнять в ней инструкции, требующие большого времени исполнения.

[FOR]

Инструкция	FOR	
Функция	Начало участка повторения программы	
Доступные элементы	K, H, KnX, KnY, KnM, KnS, KnSM, D, SD, V, Z	Количество повторений участка программы
Описание	<p>Повтор части программы между инструкциями FOR и NEXT заданное количество раз. После завершения цикла FOR выполняется следующий шаг после инструкции NEXT. Количество повторений может задаваться в диапазоне от 1 до 32767. Если будет задано отрицательное число, то цикл выполнится один раз.</p> <p>Допускается использование вложенных циклов.</p>  <p>Цикл А выполнится 2 раза, цикл В выполняется 3 раза (за каждое выполнение цикла А), т.е. Цикл В выполнится 6 раз. Количество инструкций NEXT должно равняться количеству инструкций FOR. Инструкция NEXT всегда должна идти после инструкции FOR. Большое количество повторений цикла может привести к срабатыванию сторожевого таймера.</p>	



Цикл FOR повторяется 2 раза. Когда M0 включен, с помощью индексного регистра V0 записываются значения FF(hex) в регистры D10 и D11. Затем значения присваиваются выходам из D10 в Y0-Y7, из D11 в Y10-Y17. В результате выполнения программы при включении M0 выходы Y0-Y7 и Y10-Y17 включаются, а при выключении M0 – выключаются.

[NEXT]

Инструкция	NEXT
Функция	Конец повторения программы
Доступные элементы	Нет
Описание	Конец участка повторения программы. Используется только совместно с инструкцией FOR.

4.4 Таймеры и счетчики (Timer Inst / Counter Inst)

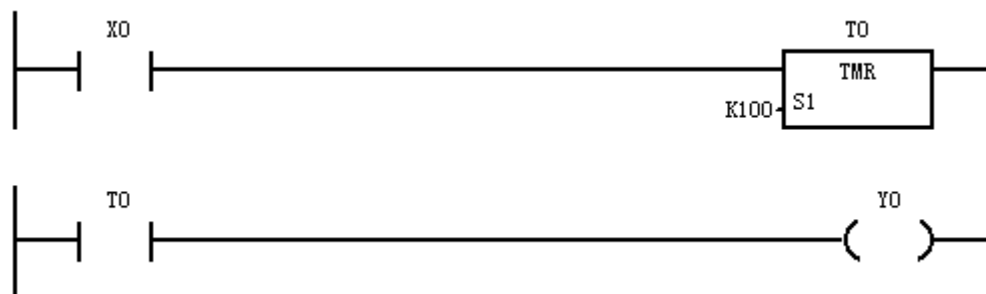
- TMR – Таймер
- CNT – Счетчик

Более подробное описание видов таймеров (общего назначения, накопительных, таймеров для подпрограмм и прерываний) представлено в разделе [3.10 Таймеры T](#). Описание разницы между стандартными, энергонезависимыми и высокоскоростными счетчиками приводится в разделе [3.11 Счётчики C](#).

[TMR]

Инструкция	TMR	
Функция	Таймер	
Идентификаторы	T0...T255	
Доступные элементы		
	Уставка таймера S1	K, H, D, SD
Описание	При активации таймер начинает отсчет времени. При достижении заданного значения включается контакт, соответствующий номеру таймера. Для повторного использования таймера необходимо его сбросить, т. е. обесточить соответствующую линию.	

При включении X0 = ON таймер T0 начинает отсчет и по достижению заданного значения в 10 секунд (т.к. шаг для таймера T0 = 100 мс, а заданное значение уставки S1 = 100) включится выход Y0. Для повторного отсчета необходимо выключить и снова включить X0.



[CNT]

Инструкция	CNT	
Функция	Счетчик	
Идентификаторы	C0...C255	
Доступные элементы		
	Уставка счетчика S1	K, H, D, SD
Описание	Счетчик производит подсчет импульсов и при достижении заданного значения включается контакт, соответствующий номеру счетчика. Для сброса текущего значения следует использовать инструкцию RST.	

Каждый раз, при изменении состояния контакта X0 с OFF на ON, происходит увеличение значения счетчика C1. Как только значение счетчика достигает 10, происходит включение выхода Y0. При замыкании входа X1 происходит сброс счетчика и, соответственно, выключение выхода Y0.



4.5 Инструкции сравнения (Compare Inst)

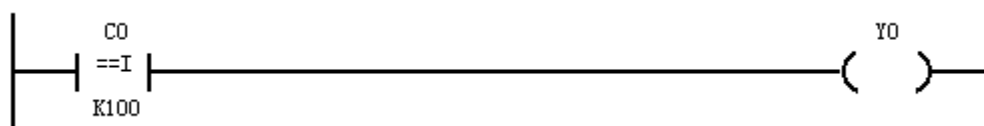
- LD= / LDD= – Проверка на «равно», целые числа
- LD<> / LDD<> – Проверка на «не равно», целые числа
- LD> / LDD> – Проверка на «больше», целые числа
- LD< / LDD< – Проверка на «меньше», целые числа
- LD>= / LDD>= – Проверка на «больше или равно», целые числа
- LD<= / LDD<= – Проверка на «меньше или равно», целые числа
- LDR= – Проверка на «равно», числа с плавающей точкой
- LDR<> – Проверка на «не равно», числа с плавающей точкой
- LDR> – Проверка на «больше», числа с плавающей точкой
- LDR< – Проверка на «меньше», числа с плавающей точкой
- LDR>= – Проверка на «больше или равно», числа с плавающей точкой
- LDR<= – Проверка на «меньше или равно», числа с плавающей точкой

[LD= / LDD=]

Инструкция	LD= (16-битная инструкция) LDD= (32-битная инструкция)
Функция	Проверка на «равно» целых чисел
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 16/32-битных значений S1 и S2. Если значения равны, то контакт замкнут.

Содержимое счетчика C0 сравнивается с константой 100.

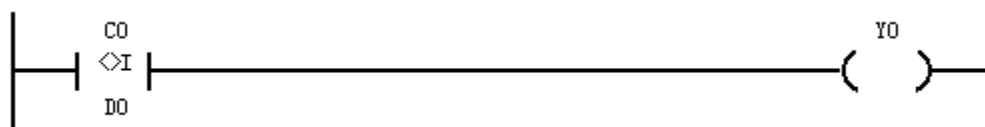
Как только значение счетчика станет равным 100 включится выход Y0.



[LD<> / LDD<>]

Инструкция	LD<> (16-битная инструкция) LDD<> (32-битная инструкция)
Функция	Проверка на «не равно» целых чисел
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 16/32-битных значений S1 и S2. Если значения не равны, то контакт замкнут.

Значение счетчика C0 сравнивается с содержимым регистра D0. Если значения не равны, то Y0 включен.



[LD> / LDD>]

Инструкция	LD> (16-битная инструкция) LDD> (32-битная инструкция)
Функция	Сравнение на «больше»
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 16/32-битных значений S1 и S2. Если S1 > S2, то контакт замкнут.

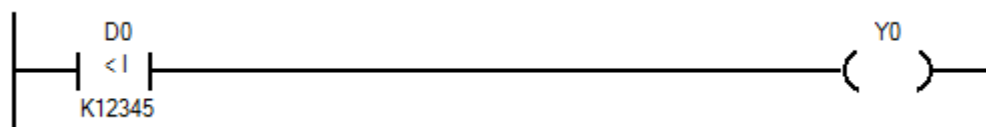
Значение регистра D0 сравнивается с константой 12345. Если D0 больше 12345, то Y0 включен.



[LD< / LDD<]

Инструкция	LD< (16-битная инструкция) LDD< (32-битная инструкция)
Функция	Сравнение на «меньше»
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 16/32-битных значений S1 и S2. Если S1 < S2, то контакт замкнут.

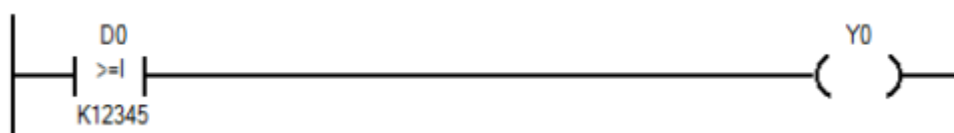
Значение регистра D0 сравнивается с константой 12345. Если D0 меньше 12345, то Y0 включен.



[LD>= / LDD>=]

Инструкция	LD>= (16-битная инструкция) LDD>= (32-битная инструкция)
Функция	Сравнение на «больше или равно»
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 16/32-битных значений S1 и S2. Если S1 >S2 или S1=S2, то контакт замкнут.

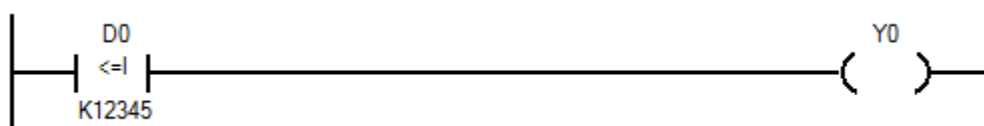
Значение регистра D0 сравнивается с константой 12345. Если содержимое D0 больше или равно 12345, то Y0 включен.



[LD<= / LDD<=]

Инструкция	LD<= (16-битная инструкция) LDD<= (32-битная инструкция)
Функция	Сравнение на «меньше или равно»
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 16/32-битных значений S1 и S2. Если S1 < S2 или S1=S2, то контакт замкнут.

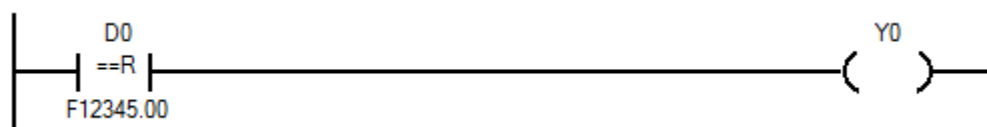
Содержимое регистра D0 сравнивается с константой 12345. Если D0 меньше или равно 12345, то Y0 включен.



[LDR=]

Инструкция	LDR=
Функция	Проверка на «равно» чисел с плавающей точкой
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 32-битных значений S1 и S2. Если значения равны, то контакт замкнут.

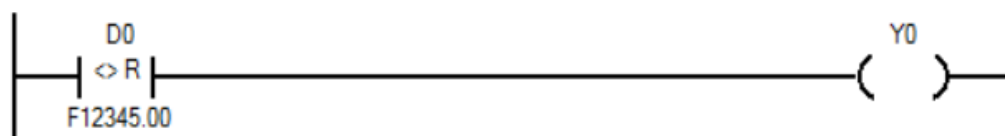
Содержимое регистра D0 сравнивается с константой F12345.00 (константа типа «Float»). Как только значение регистра D0 станет равным 12345.00 включится выход Y0.



[LDR<>]

Инструкция	LDR<>
Функция	Проверка на «не равно» чисел с с плавающей точкой
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 32-битных значений S1 и S2. Если значения не равны, то контакт замкнут.

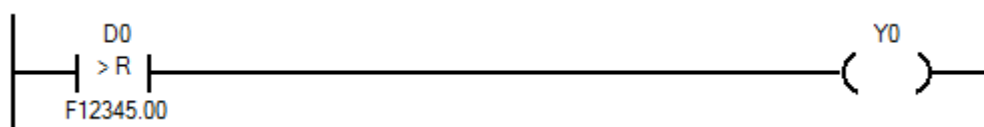
Значение регистра D0 сравнивается с константой 12345.00. Если значения не равны, то Y0 включен.



[LDR>]

Инструкция	LDR>
Функция	Сравнение на «больше» чисел с плавающей точкой
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 32-битных значений S1 и S2. Если $S1 > S2$, то контакт замкнут.

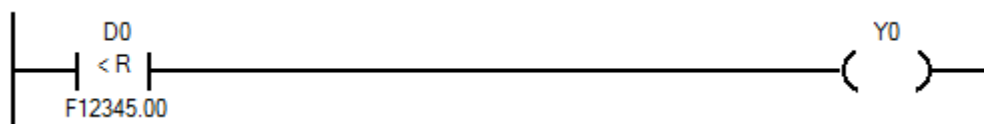
Значение регистра D0 сравнивается с константой 12345.00. Если D0 больше 12345,00, то Y0 включен.



[LDR<]

Инструкция	LDR<
Функция	Сравнение на «меньше» чисел с плавающей точкой
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 32-битных значений S1 и S2. Если $S1 < S2$, то контакт замкнут.

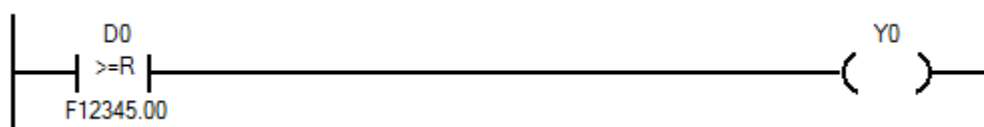
Значение регистра D0 сравнивается с константой 12345.00. Если D0 меньше 12345,00, то Y0 включен.



[LDR>=]

Инструкция	LDR>=
Функция	Сравнение на «больше или равно» чисел с плавающей точкой
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 32-битных значений S1 и S2. Если S1 >S2 или S1=S2, то контакт замкнут.

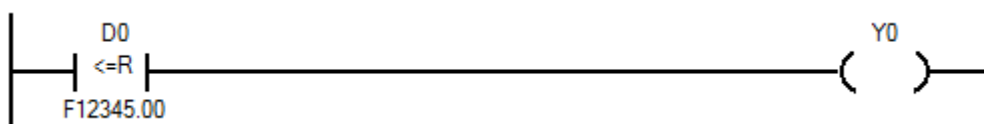
Значение регистра D0 сравнивается с константой 12345.00. Если D0 больше или равно 12345.00 то Y0 включен.



[LDR<=]

Инструкция	LDR<=
Функция	Сравнение на «меньше или равно» чисел с плавающей точкой
Доступные элементы	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	Сравнение двух 32-битных значений S1 и S2. Если S1 < S2 или S1=S2, то контакт замкнут.

Содержимое регистра D0 сравнивается с константой 12345.00. Если D0 меньше или равно 12345.00, то Y0 включен.



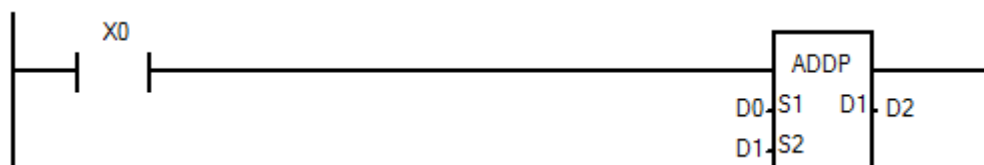
4.6 Математические инструкции (Math Inst)

- ADD / DADD – Сложение двух целых чисел
- SUB / DSUB – Вычитание двух целых чисел
- MUL / DMUL – Умножение двух целых чисел
- DIV / DDIV – Деление двух целых чисел
- INC / DINC – Инкремент целого числа
- DEC / DDEC – Декремент целого числа
- WAND / DAND – Логическое И
- WOR / DOR – Логическое ИЛИ
- WXOR / DXOR – Исключающее ИЛИ
- NEG / DNEG – Логическое отрицание

[ADD / DADD]

Инструкция	ADD (16-битная инструкция) ADDP (16-битная инструкция, импульсное выполнение) DADD (32-битная инструкция) DADDP (32-битная инструкция, импульсное выполнение)		
Функция	Сложение двух целых чисел		
Доступные входы	S1	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Первое слагаемое
	S2		Второе слагаемое
Доступные выходы	D1	T, C, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Сумма
Описание	<p>Инструкция суммирует и сохраняет результат в D1. Формула вычисления: $D1 = S1 + S2$. Если результат расчета равен 0, будет установлен флаг нуля SM20. Если результат вычисления превышает 32767 (16-ти битная операция) или 2147483647 (32-х битная операция), будет установлен флаг переноса SM22. Если результат вычисления меньше -32768 (16-ти битная операция) или -2147483648 (32-х битная операция), будет установлен флаг заимствования SM21.</p>		

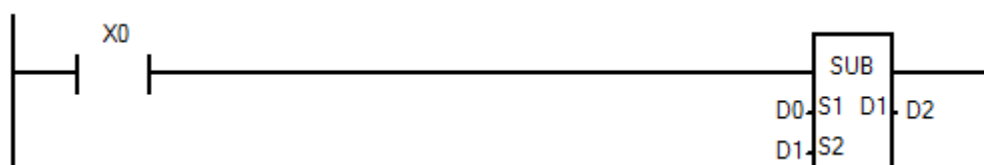
Всё время, пока X0 включен, значения регистров D0 и D1 суммируются и результат помещается в регистр D2.



[SUB / DSUB]

Инструкция	SUB (16-ти битная инструкция) SUBP (16-ти битная инструкция, импульсное исполнение) DSUB (32-х битная инструкция) DSUBP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычитание двух целых чисел		
Доступные входы	S1	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Уменьшаемое
	S2		Вычитаемое
Доступные выходы	D1	T, C, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Разность
Описание	<p>Инструкция вычисляет разность в двоичном формате S1 и S2 и сохраняет результат в D1.</p> <p>Формула вычисления: $D1 = S1 - S2$.</p> <p>Если результат расчета равен 0, будет установлен флаг нуля SM20.</p> <p>Если результат вычисления превышает 32767 (16-ти битная операция) или 2147483647 (32-х битная операция), будет установлен флаг переноса SM22.</p> <p>Если результат вычисления меньше -32768 (16-ти битная операция) или -2147483648 (32-х битная операция), будет установлен флаг заимствования SM21.</p>		

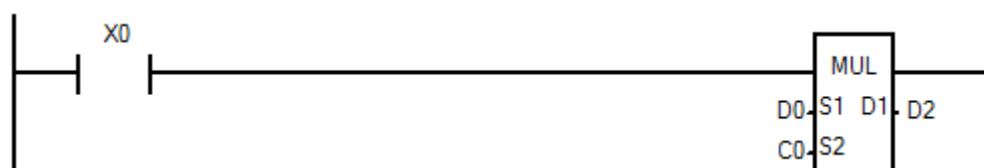
Пока X0 включен, из значения регистра D0 вычитается значение регистра D1 и результат помещается в D2.



[MUL / DMUL]

Инструкция	MUL (16-ти битная инструкция) MULP (16-ти битная инструкция, импульсное исполнение) DMUL (32-х битная инструкция) DMULP (32-х битная инструкция, импульсное исполнение)		
Функция	Умножение двух целых чисел		
Доступные входы	S1	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Первый множитель
	S2		Второй множитель
Доступные выходы	D1	T, C, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Результат умножения
Описание	<p>Инструкция вычисляет произведение чисел S1 и S2 и сохраняет результат в D1.</p> <p>Формула вычисления $D1 = S1 * S2$;</p> <p>Результат может быть только 32-х разрядным, занимает 2 соседних регистра.</p>		

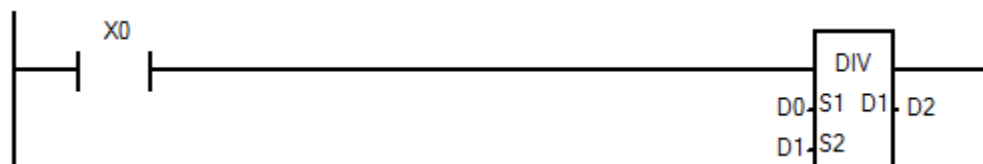
Когда X0 включен, значение регистра D0 умножается на значение счетчика C0 и результат помещается в D2 и D3.



[DIV / DDIV]

Инструкция	DIV (16-ти битная инструкция) DIVP (16-ти битная инструкция, импульсное исполнение) DDIV (32-х битная инструкция) DDIVP (32-х битная инструкция, импульсное исполнение)		
Функция	Деление двух целых чисел		
Доступные входы	S1	T, C, D, SD, V, Z, K, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Делимое
	S2		Делитель
Доступные выходы	D1	T, C, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Частное - результат деления
Описание	<p>При исполнении 16-ти битной инструкции частное помещается в D, а остаток от деления в D+1;</p> <p>При исполнении 32-х битной инструкции результат помещается в D и D+1 а остаток от деления в D+2 и D+3;</p> <p>Если делимое является отрицательным числом, то и остаток от деления так же будет отрицательным;</p> <p>При использовании в качестве регистра хранения результата KnX, KnY, KnM, KnS остаток не будет получен;</p> <p>Если делитель S2 равен 0, то вычисление производиться не будет</p>		

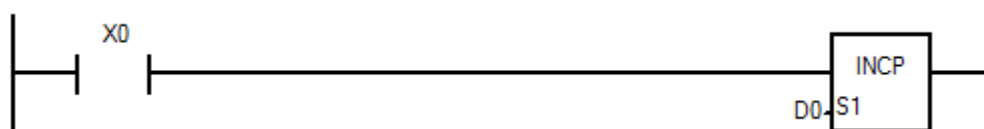
Когда X0 включен, значение регистра D0 делится на значение регистра D1, результат помещается в регистр D2 а остаток от деления в регистр D3.



[INC / DINC]

Инструкция	INC (16-ти битная инструкция) INCP (16-ти битная инструкция, импульсное исполнение) DINC (32-х битная инструкция) DINCP (32-х битная инструкция, импульсное исполнение)	
Функция	Инкремент целого числа	
Доступные входы		
	S1	Т, С, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	<p>Инструкция прибавляет 1 к значению в S1; Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то прибавление будет осуществляться в каждом цикле сканирования; Для 16-ти битной инструкции при достижении значения 32767 и добавлении 1 будет записано значение -32768; Для 32-ти битной инструкции при достижении значения 2147483647 и добавлении 1 будет записано значение -2147483648; Инструкция при выполнении не устанавливает флаг нуля SM20, переноса SM22 или заимствования SM21;</p>	

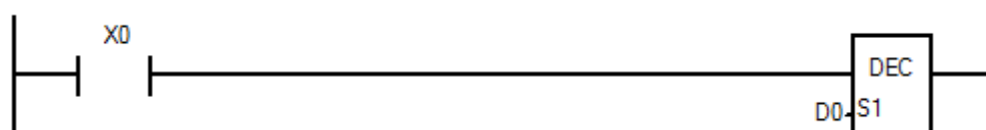
По каждому переднему фронту сигнала X0 к значению регистра D0 прибавляется 1.



[DEC / DDEC]

Инструкция	DEC (16-ти битная инструкция) DECP (16-ти битная инструкция, импульсное исполнение) DDEC (32-х битная инструкция) DDECP (32-х битная инструкция, импульсное исполнение)	
Функция	Декремент целого числа	
Доступные входы		
	S1	Т, С, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS
Описание	<p>Инструкция вычитает 1 из значения в S1; Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то вычитание будет осуществляться в каждом цикле сканирования;</p> <p>Для 16-ти битной инструкции при достижении значения -32768 и вычитании 1 будет записано значение 32767; Для 32-ти битной инструкции при достижении значения -2147483648 и вычитании 1 будет записано значение 2147483647;</p> <p>Инструкция при выполнении не устанавливает флаг нуля SM20, переноса SM22 и заимствования SM21;</p>	

Когда X0 включен, из значения регистра D0 вычитается 1 в каждом цикле сканирования.



[WAND / DAND]

Инструкция	WAND (16-ти битная инструкция) WANDP (16-ти битная инструкция, импульсное исполнение) DAND (32-х битная инструкция) DANDP (32-х битная инструкция, импульсное исполнение)		
Функция	Логическое И		
Доступные входы	S1	T, C, D, SD, V, Z, K, H, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Первое значение
	S2		Второе значение
Доступные выходы	D1	KnY, KnM, KnSM, KnS, T, C, D	Результат
Описание	Инструкция производит операцию «Логическое И» с содержимым регистров S1 и S2 и результат сохраняет в D1.		

Когда X0 включен, над содержимым регистров D0 и D1 производится операция «Логическое И» в каждом цикле сканирования и результат сохраняется в D2.

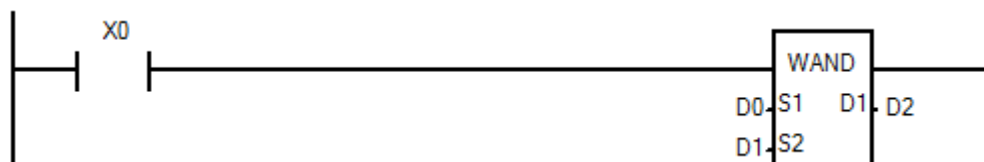


Таблица истинности:

D0	0	1	1	0
D1	0	0	1	1
D2	0	0	1	0

[WOR / DOR]

Инструкция	WOR (16-ти битная инструкция) WORP (16-ти битная инструкция, импульсное исполнение) DOR (32-х битная инструкция) DORP (32-х битная инструкция, импульсное исполнение)		
Функция	Логическое ИЛИ		
Доступные входы	S1	T, C, D, SD, V, Z, K, H, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Первое значение
	S2		Второе значение
Доступные выходы	D1	KnY, KnM, KnSM, KnS, T, C, D	Результат
Описание	Инструкция производит операцию «Логическое ИЛИ» с содержимым регистров S1 и S2 и результат сохраняет в D1.		

Когда X0 включен, над содержимым регистров D0 и D1 производится операция «Логическое ИЛИ» и результат сохраняется в D2.

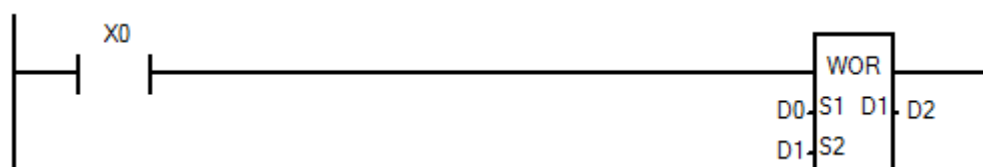


Таблица истинности:

D0	0	1	1	0
D1	0	0	1	1
D2	0	1	1	1

[WXOR / DXOR]

Инструкция	WXOR (16-ти битная инструкция) WXORP (16-ти битная инструкция, импульсное исполнение) DXOR (32-х битная инструкция) DXORP (32-х битная инструкция, импульсное исполнение)		
Функция	Исключающее ИЛИ		
Доступные входы	S1	T, C, D, SD, V, Z, K, H, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Первое значение
	S2		Второе значение
Доступные выходы	D1	KnY, KnM, KnSM, KnS, T, C, D	Результат
Описание	Инструкция производит операцию «Исключающее ИЛИ» с содержимым регистров S1 и S2 и результат сохраняет в D1.		

Когда X0 включен, над содержимым регистров D0 и D1 производится операция «Исключающее ИЛИ» и результат сохраняется в D2.

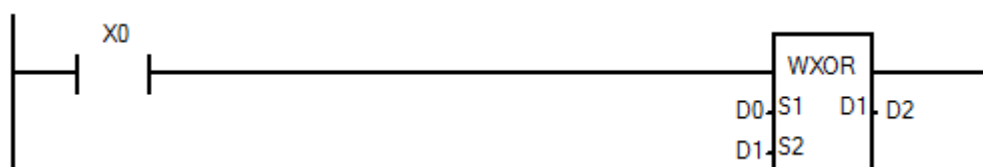


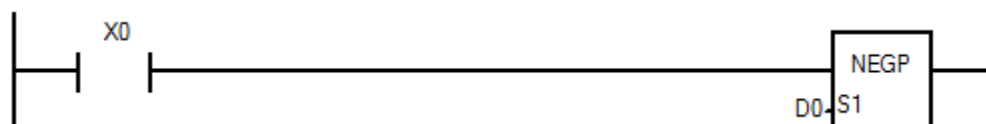
Таблица истинности:

D0	0	1	1	0
D1	0	0	1	1
D2	0	1	0	1

[NEG / DNEG]

Инструкция	NEG (16-ти битная инструкция) NEGP (16-ти битная инструкция, импульсное исполнение) DNEG (32-х битная инструкция) DNEGP (32-х битная инструкция, импульсное исполнение)		
Функция	Логическое отрицание		
Доступные входы			
	S1	T, C, D, SD, V, Z, K, H, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Число
Описание	Инструкция производит операцию «Логическое НЕ» с содержимым регистра S1, затем прибавляется 1 и результат сохраняется в S1. Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то логическая операция будет осуществляться в каждом цикле сканирования.		

Когда X0 включен, над содержимым регистра D0 производится операция «Логическое НЕ», затем прибавляется 1. Инструкцию можно применять для получения абсолютного значения из отрицательного числа.



Пример: до применения операции NEG в регистре D0 хранится десятичное число -2. После применения операции NEG в регистре D0 хранится значение 2.

Таблица истинности:

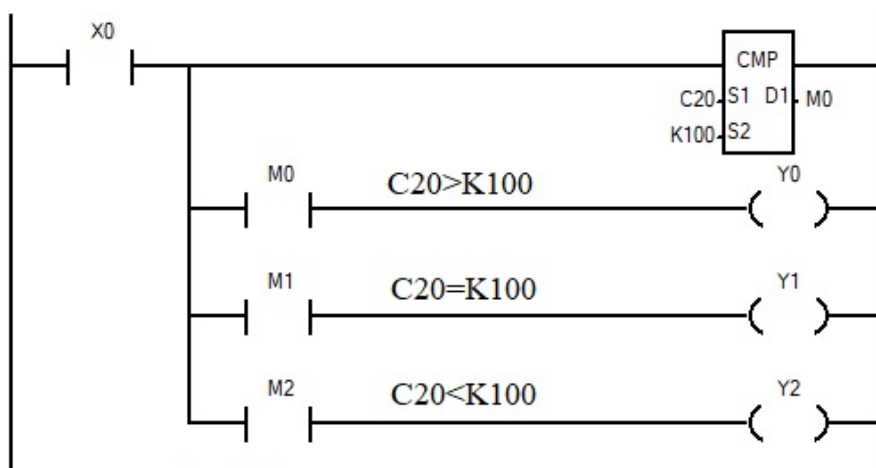
D0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
NEGP D0																
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

4.7 Инструкции сравнения и передачи данных (Transfer and Compare Inst)

- **CMP / DCMP** – Сравнение
- **ZCP / DCPP** – Зонное сравнение
- **MOV / DMOV** – Пересылка данных
- **CML / DCML** – Пересылка данных с инвертированием
- **VMOV / VMOV P** – Групповая пересылка данных
- **FMOV / DFMOV** – Пересылка данных в несколько адресов.
- **XCH / DXCH** – Обмен данными
- **BCD / DBCD** – Пересылка данных с преобразованием в BCD - формат
- **BIN / DBIN** – Пересылка данных с преобразованием в BIN - формат
- **DEMOV / DEMOV P** – Пересылка данных с преобразованием из FLOAT в BIN - формат

[CMP / DCMP]

Инструкция	CMP (16-битная инструкция) CMP (16-битная инструкция, импульсное исполнение) DCMP (32-битная инструкция) DCMP (32-битная инструкция, импульсное исполнение)		
Функция	Сравнение		
Доступные входы	S1	T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY,	Первое сравниваемое значение
	S2	KnM, KnSM, KnS	Второе сравниваемое значение
Доступные выходы	D1	Y, M, SM, S	Результат сравнения
Описание	<p>При выполнении инструкции сравниваются значения в S1 и S2 и результат помещается в D1. Элемент D1 занимает три последовательных значения D1, D1+1, D1+2:</p> <ul style="list-style-type: none"> - если $S1 > S2$ то включается D1; - если $S1 = S2$, то включается D1+1; - если $S1 < S2$, то включается D1+2. 		

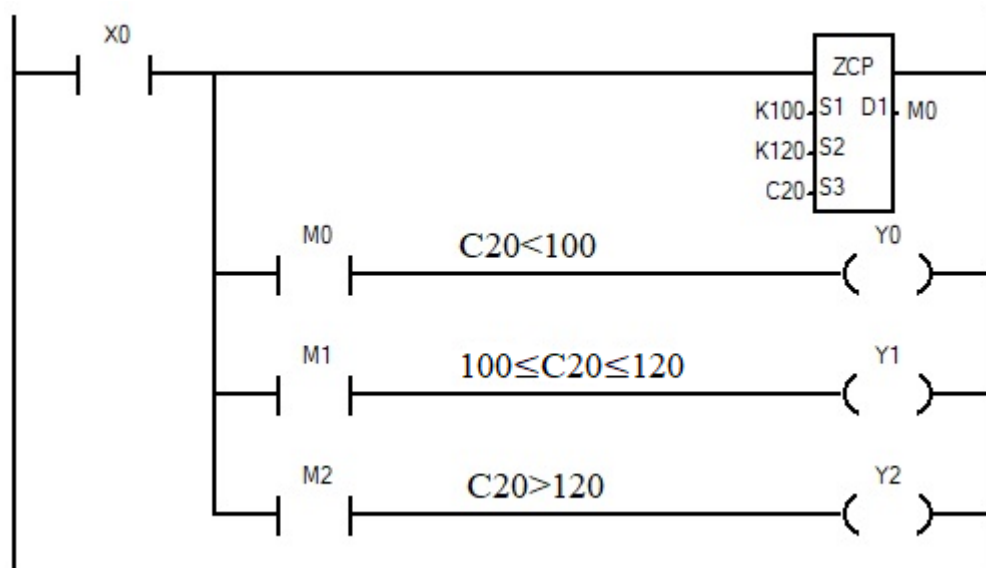


Когда X0 включен, происходит сравнение значения счетчика C20 с числом 100, результат помещается в M0:

- если значение C20 больше 100, то включается M0 и выход Y0;
- если значение C20 равно 100, то включается M1 и выход Y1;
- если значение C20 меньше 100, то включается M2 и выход Y2.

[ZCP / DСPP]

Инструкция	ZCP (16-битная инструкция) ZCPP (16-битная инструкция, импульсное исполнение) DZCP (32-битная инструкция) DZCPP (32-битная инструкция, импульсное исполнение)		
Функция	Зонное сравнение		
Доступные входы	S1	T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS	Начальное значение диапазона
	S2		Конечное значение диапазона
	S3		Число внутри диапазона
	Доступные выходы		
D1	Y, M, SM, S	Результат сравнения	
Описание	<p>При выполнении инструкции сравниваются значения в S3 со значением интервала, заданными S1 и S2, результат помещается в D1. S1 должен быть меньше S2. Элемент D1 занимает три последовательных значения D1, D1+1, D1+2:</p> <ul style="list-style-type: none"> - если $S3 < S1$ то включается D1; - если $S1 \leq S3 \leq S2$, то включается D1+1; - если $S3 > S2$, то включается D1+2. 		

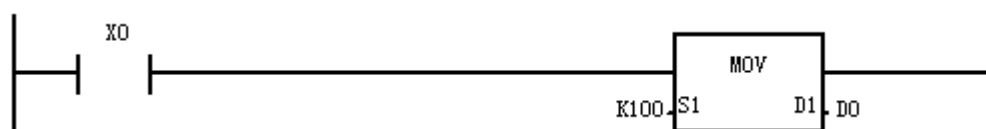


Когда X0 включен, происходит сравнение значения счетчика C20 с числами 100 и 120, результат помещается в M0:

- если значение C20 меньше 100, то включается M0 и выход Y0;
- если значение C20 больше или равно 100 но меньше либо равно 120, то включается M1 и выход Y1;
- если значение C20 больше 120, то включается M2 и выход Y2.

[MOV / DMOV]

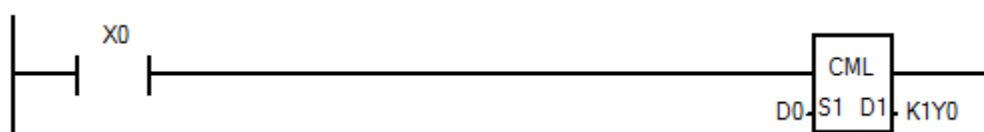
Инструкция	MOV (16-битная инструкция) MOVP (16-битная инструкция, импульсное исполнение) DMOV (32-битная инструкция) DMOVP (32-битная инструкция, импульсное исполнение)	
Функция	Пересылка данных	
Доступные входы	S1	Пересылаемое значение T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS
Доступные выходы	D1	Приемник T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS
Описание	При выполнении инструкции данные из источника S1 пересылаются в приемник D1.	



Когда X0 включен, в регистр D0 пересылается число 100.

[CML / DCML]

Инструкция	CML (16-битная инструкция) CMLP (16-битная инструкция, импульсное исполнение) DCML (32-битная инструкция) DCMLP (32-битная инструкция, импульсное исполнение)	
Функция	Пересылка данных с инвертированием	
Доступные входы	S1	Т, С, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS
	Входное значение	
Доступные выходы	D1	Т, С, D, SD, V, Z, AQ, KnY, KnM, KnSM, KnS
	Приемник данных	
Описание	<p>При выполнении инструкции данные из источника S1 переводятся в двоичный формат и пересылаются в приемник данных D1, при этом все биты инвертируются. Если в качестве приёмника данных использовать регистры KnY, KnM, KnSM, KnS, то количество инвертируемых битов определяется константой Kn в полубайтах.</p> <p>K1 – 4 первых бита K2 – 8 первых битов K3 – 12 первых битов и т.п</p>	



Когда X0 включен, первые 4 бита регистра D0 инвертируются и включаются соответствующие выходы Y.

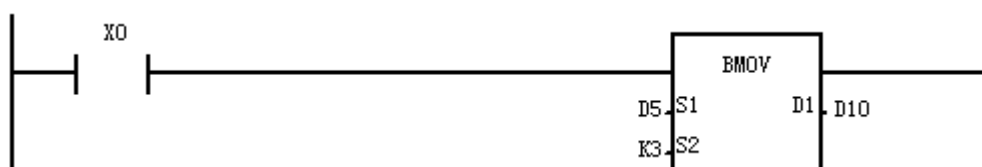
Например, если D0=6:

D0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
CML D0 K1Y0																	
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

После выполнения инструкции CML будут включены выходы Y0 и Y3.

[BMOV]

Инструкция	BMOV (16-битная инструкция) BMOVР (16-битная инструкция, импульсное исполнение)		
Функция	Групповая пересылка данных		
Доступные входы	S1	T, C, D, SD, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Начальный адрес пересылаемого регистра
	S2	D, SD, K	Количество регистров
Доступные выходы	D1	T, C, D, SD, KnY, KnM, KnSM, KnS	Начальный адрес регистра приемника
Описание	При выполнении инструкции данные из регистров с начальным номером S1 в количестве S2 пересылаются в регистры с начальным номером D1.		



Когда X0 включен, данные из регистров D5, D6, D7 пересылаются в регистры D10, D11, D12 следующим образом:

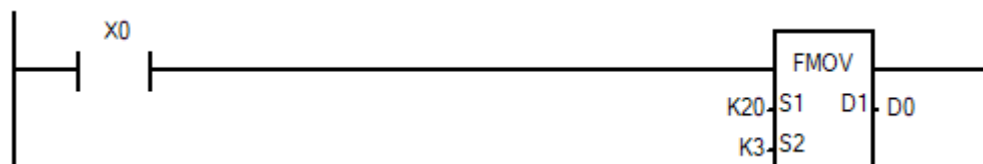
D5 → D10

D6 → D11

D7 → D12

[FMOV / DFMOV]

Инструкция	FMOV (16-битная инструкция) FMOV _P (16-битная инструкция, импульсное исполнение) DFMOV (32-битная инструкция) DFMOV _P (32-битная инструкция, импульсное исполнение)		
Функция	Пересылка данных в несколько адресов		
Доступные входы	S1	T, C, D, SD, AI, AQ, KnX, KnY, KnM, KnSM, KnS	Пересылаемое значение
	S2	K	Количество регистров приемника с начальным адресом D1
Доступные выходы	D1	T, C, D, SD, KnY, KnM, KnSM, KnS	Начальный адрес регистра приемника
Описание	При выполнении инструкции данные из S1 пересылаются в регистры с начальным номером D1 и количеством S2.		



Когда X0 включен, число 20 пересылается в регистры D0, D1, D2 следующим образом:

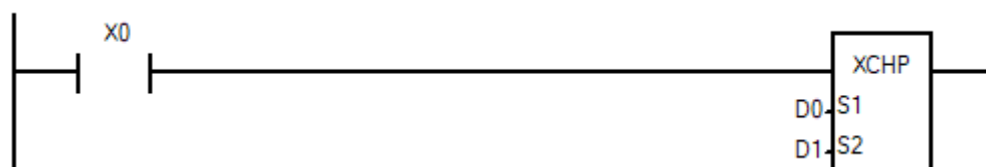
K20 → D0

K20 → D1

K20 → D2

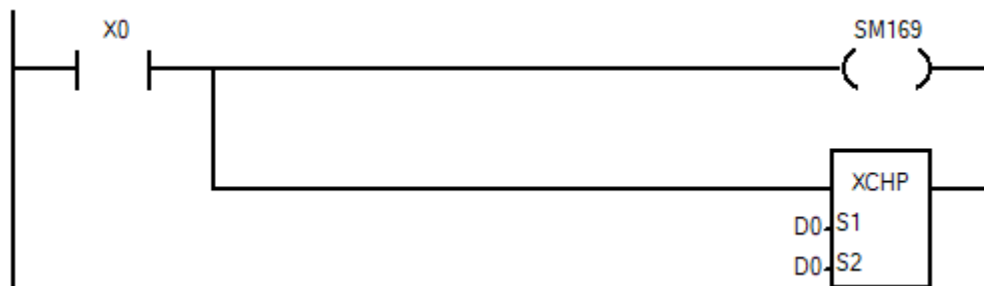
[XCH / DXCH]

Инструкция	XCH (16-битная инструкция) XCHP (16-битная инструкция, импульсное исполнение) DXCH (32-битная инструкция) DXCHP (32-битная инструкция, импульсное исполнение)		
Функция	Обмен данными		
Доступные входы	S1	T, C, D, V, Z, KnY, KnM, KnSM, KnS	Первый регистр обмена
	S2		Второй регистр обмена
Доступные выходы	D1	T, C, D, SD, KnY, KnM, KnSM, KnS	Начальный адрес регистра приемника
Описание	<p>При выполнении инструкции данные в регистрах S1 и S2 меняются местами.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то обмен данными будет осуществляться в каждом цикле сканирования;</p> <p>Если S1=S2, и включено специальное внутреннее реле SM169, то:</p> <ul style="list-style-type: none"> -при выполнении 16-ти битной инструкции старшие и младшие 8 бит меняются местами. -при выполнении 32-х битной инструкции старшие и младшие 16 бит меняются местами. <p>Если S1 не равен S2 и включено специальное реле SM169, то включится флаг ошибки SM67 и инструкция не будет выполнена.</p>		



Когда X0 включен, содержимое регистров D0 и D1 меняется местами.

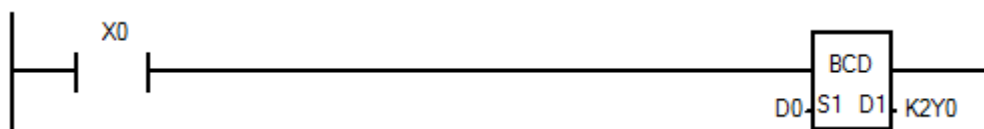
D0 ↔ D1



Когда X0 включен младшие и старшие 8 бит в регистре D0 меняются местами.

[BCD / DBCD]

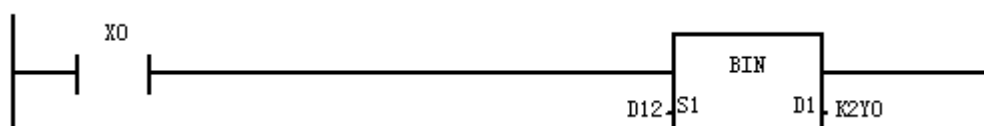
Инструкция	BCD (16-битная инструкция) BCDP (16-битная инструкция, импульсное исполнение) DBCDC (32-битная инструкция) DBCDCP (32-битная инструкция, импульсное исполнение)		
Функция	Пересылка данных с преобразованием в BCD - формат		
Доступные входы			
	S1	T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS	Передаваемое значение
Доступные выходы			
	D1	T, C, D, SD, V, Z, AI, AQ, KnY, KnM, KnSM, KnS	Приемник
Описание	<p>При выполнении инструкции данные из источника S1 переводятся в формат BCD и пересылаются в D1.</p> <p>Для 16-ти битной инструкции результат преобразования должен находиться в диапазоне 0-9999.</p> <p>Для 32-х битной инструкции результат преобразования должен находиться в диапазоне 0-99999999. Выход за указанные диапазоны приведет к возникновению ошибки, при этом установится флаг SM67.</p>		



Когда X0 включен, значение регистра D0 переводятся в формат BCD и значение присваивается выходам Y.

[BIN / DBIN]

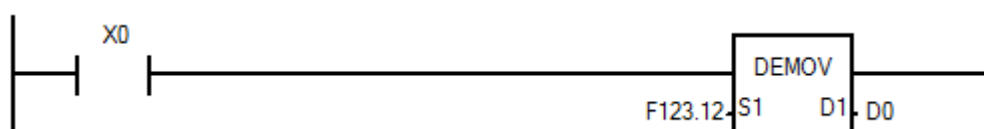
Инструкция	BIN (16-битная инструкция) BINP (16-битная инструкция, импульсное исполнение) DBIN (32-битная инструкция) DBINP (32-битная инструкция, импульсное исполнение)		
Функция	Пересылка данных с преобразованием в BIN - формат		
Доступные входы			
	S1	T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS	Передаваемое значение
Доступные выходы			
	D1	T, C, D, SD, V, Z, AI, AQ, KnY, KnM, KnSM, KnS	Приемник
Описание	<p>При выполнении инструкции данные из источника S1 в BCD-формате переводятся в формат BIN и пересылаются в D1.</p> <p>Для 16-ти битной инструкции значение S1 должно быть в BCD- формате и находится в диапазоне 0-9999.</p> <p>Для 32-х битной инструкции значение S1 должно быть в BCD- формате и находится в диапазоне 0-99999999.</p> <p>Если S1 заданно не в BCD- формате, то возникнет ошибка выполнения и установятся флаги SM67 и SM68.</p>		



Когда X0 включен, значение регистра D12 переводятся в формат BIN и значение присваивается выходам Y ПЛК.

[DEMOV / DEMOVP]

Инструкция	DEMOV (стандартная инструкция) DEMOVP (импульсное исполнение)	
Функция	Пересылка данных с преобразованием из FLOAT в BIN - формат	
Доступные входы		
	S1	T, C, D, SD, V, Z, AI, AQ, K, H, KnX, KnY, KnM, KnSM, KnS Передаваемое значение типа FLOAT
Доступные выходы		
	D1	T, C, D, SD, V, Z, AI, AQ, KnY, KnM, KnSM, KnS Приемник
Описание	При выполнении инструкции данные из источника S1 в формате числа с плавающей точкой переводятся в формат BIN и пересылаются в D1.	



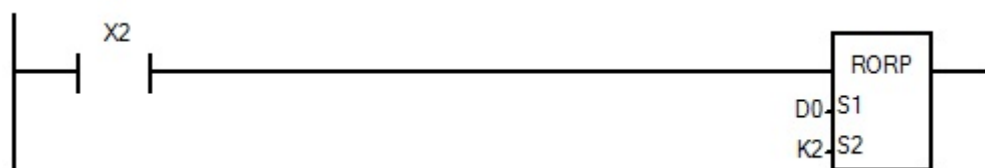
Когда X0 включен, число с плавающей точкой 123.12 переводится в формат BIN и пересылается в регистр D0.

4.8 Инструкции сдвига (Shift Inst)

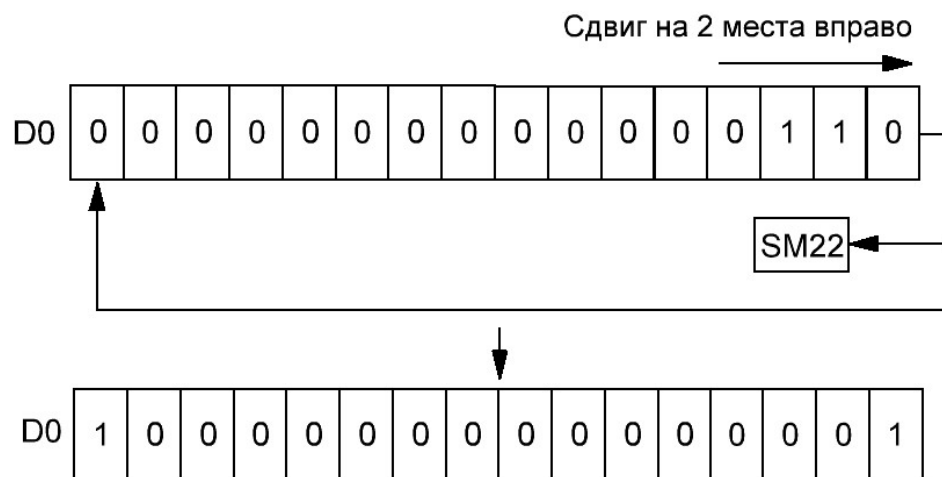
- ROR / DROR – Кольцевой сдвиг вправо
- ROL / DROL – Кольцевой сдвиг влево
- RCR / DRCR – Кольцевой сдвиг вправо с использованием флага переноса
- RCL / DRCL – Кольцевой сдвиг влево с использованием флага переноса
- SFTR / SFTRP – Групповой сдвиг битовых значений вправо
- SFTL / SFTLP – Групповой сдвиг битовых значений влево
- WSFR / WSFRP – Пословный сдвиг регистров вправо
- WSFL / WSFLP – Пословный сдвиг регистров влево
- SFWR / SFWRP – Запись данных в стек
- SFRD / SFRDP – Чтение данных из стека

[ROR / DROR]

Инструкция	ROR (16-ти битная инструкция) RORP (16-ти битная инструкция, импульсное исполнение) DROR (32-х битная инструкция) DRORP (32-х битная инструкция, импульсное исполнение)		
Функция	Кольцевой сдвиг вправо		
Доступные входы	S1	KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Данные
	S2	K(0...16)	Количество сдвигаемых бит
Описание	<p>При выполнении инструкции данные в S1 сдвигаются вправо на количество мест S2; Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования; Значение последнего сдвигаемого бита копируется в регистр SM22.</p> <p>Если в качестве S1 используются KnX, KnY, KnM, KnSM, KnS то S2 может быть не более 4 для 16-ти битной инструкции и не более 8 для 32-х битной инструкции.</p>		

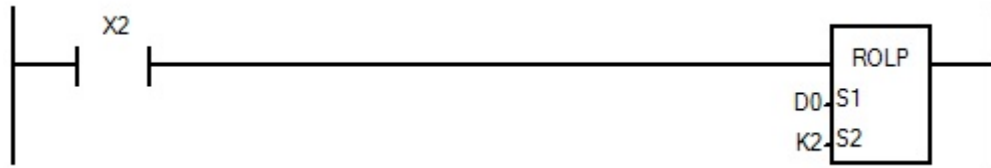


Когда X2 изменяет состояние с OFF на ON, биты в регистре D0 сдвигаются на 2 места вправо. При этом значение последнего сдвигаемого бита копируется в регистр SM22. В данном примере последний сдвигаемый бит равен 1, поэтому после выполнения инструкции SM22 = 1.

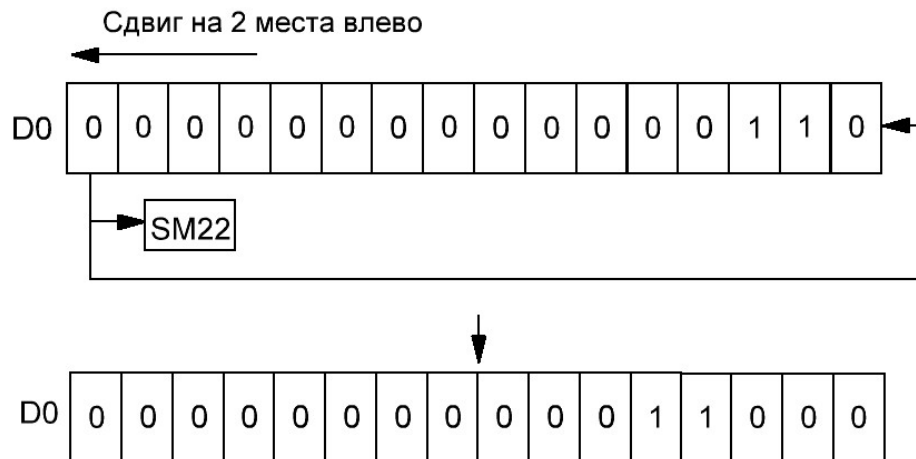


[ROL / DROL]

Инструкция	ROL (16-ти битная инструкция) ROLP (16-ти битная инструкция, импульсное исполнение) DROL (32-х битная инструкция) DROLP (32-х битная инструкция, импульсное исполнение)		
Функция	Кольцевой сдвиг влево		
Доступные входы	S1	KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Данные
	S2	K(0...16)	Количество сдвигаемых бит
Описание	<p>При выполнении инструкции данные в S1 сдвигаются влево на количество мест S2;</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования; Значение последнего сдвигаемого бита копируется в регистр SM22.</p> <p>Если в качестве S1 используются KnX, KnY, KnM, KnSM, KnS то S2 может быть не более 4 для 16-ти битной инструкции и не более 8 для 32-х битной инструкции.</p>		

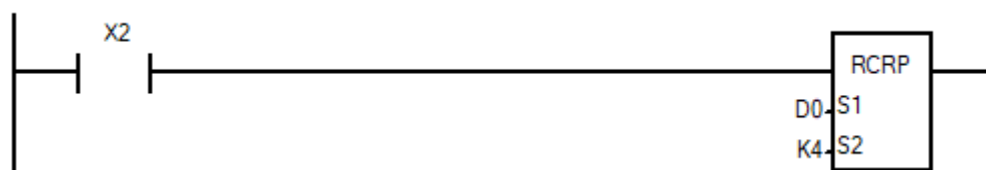


Когда X2 изменяет состояние с OFF на ON, биты в регистре D0 сдвигаются на 2 места влево. При этом значение последнего сдвигаемого бита копируется в регистр SM22. В данном примере последний сдвигаемый бит равен 0, поэтому после выполнения инструкции SM22 = 0.

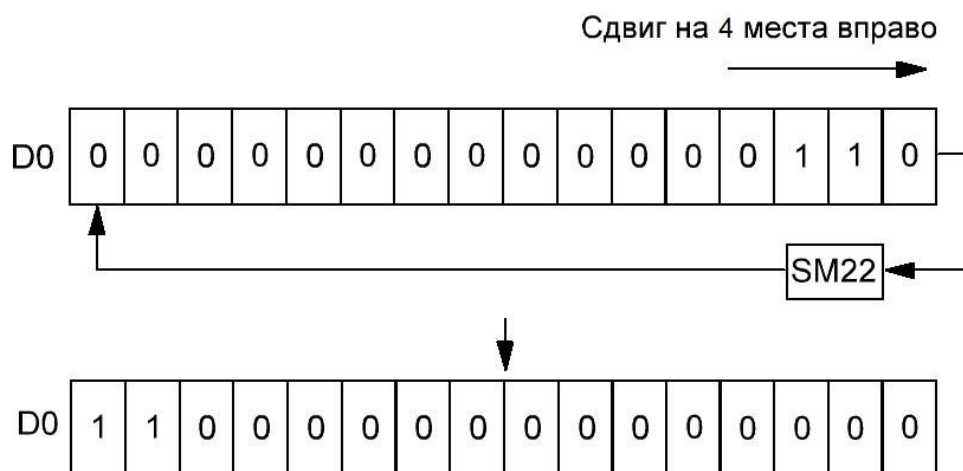


[RCR / DRCR]

Инструкция	RCR (16-ти битная инструкция) RCRP (16-ти битная инструкция, импульсное исполнение) DRCR (32-х битная инструкция) DRCRP (32-х битная инструкция, импульсное исполнение)		
Функция	Кольцевой сдвиг вправо с использованием флага переноса		
Доступные входы			
	S1	KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Данные
	S2	K(0...16)	Количество сдвигаемых бит
Описание	<p>При выполнении инструкции данные в S1 сдвигаются вправо на количество мест S2 через флаг переноса; Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования; Значение SM22 воспринимается совместно со сдвигаемым регистром, т.е. как 17 бит.</p> <p>Если в качестве S1 используются KnX, KnY, KnM, KnSM, KnS то S2 может быть не более 4-х для 16-ти битной инструкции и не более 8-ми для 32-х битной инструкции.</p>		

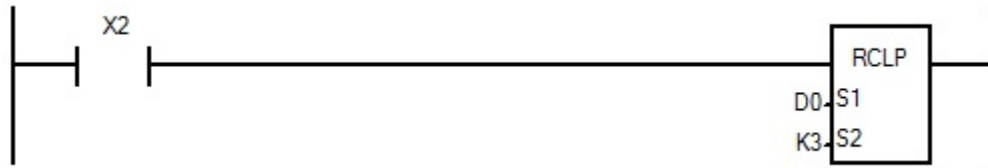


Когда X2 изменяет состояние с OFF на ON, биты в регистре D0 вместе с флагом переноса SM22 (всего 17 бит) сдвигаются на 4 места вправо. В SM22 остается значение последнего сдвигаемого бита, в данном примере SM22=0. Примечание. Похоже, что в данном примере SM22=0, а сдвиг произошел на 4 места вправо.

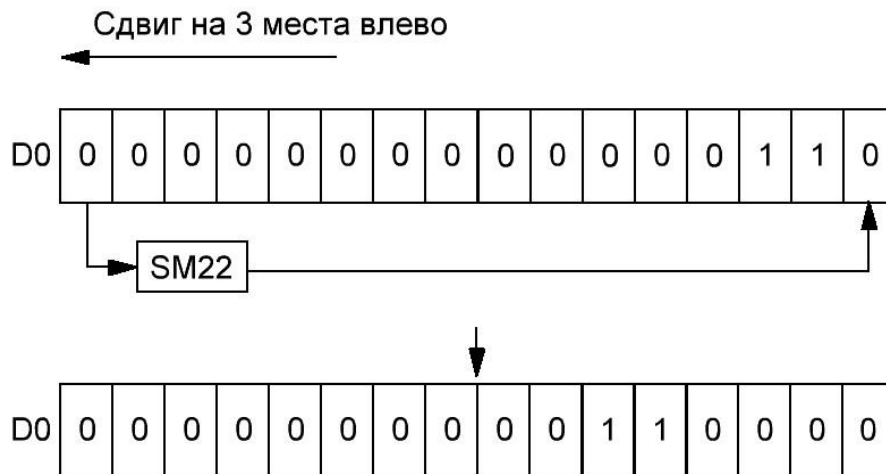


[RCL / DRCL]

Инструкция	RCL (16-ти битная инструкция) RCLP (16-ти битная инструкция, импульсное исполнение) DRCL (32-х битная инструкция) DRCLP (32-х битная инструкция, импульсное исполнение)		
Функция	Кольцевой сдвиг влево с использованием флага переноса		
Доступные входы	S1	KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Данные
	S2	K(0...16)	Количество сдвигаемых бит
Описание	<p>При выполнении инструкции данные в S1 сдвигаются влево на количество мест S2 через флаг переноса; Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования; Значение SM22 воспринимается совместно со сдвигаемым регистром, т.е. как 17 бит; Если в качестве S1 используются KnX, KnY, KnM, KnSM, KnS то S2 может быть не более 4-х для 16-ти битной инструкции и не более 8-ми для 32-х битной инструкции.</p>		

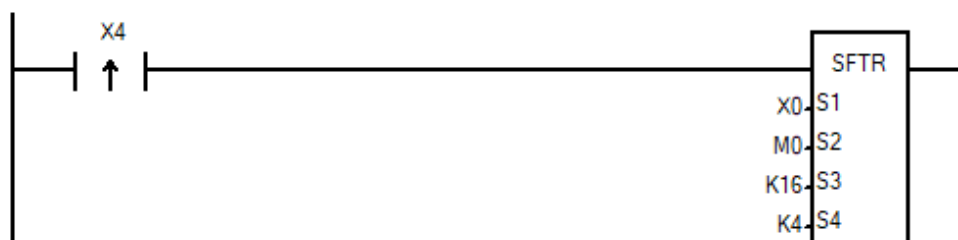


Когда X0 изменяет состояние с OFF на ON, биты в регистре D0 вместе с флагом переноса SM22 (всего 17 бит) сдвигаются на 3 места влево. В SM22 остается значение последнего сдвигаемого бита, в данном примере SM22=0.

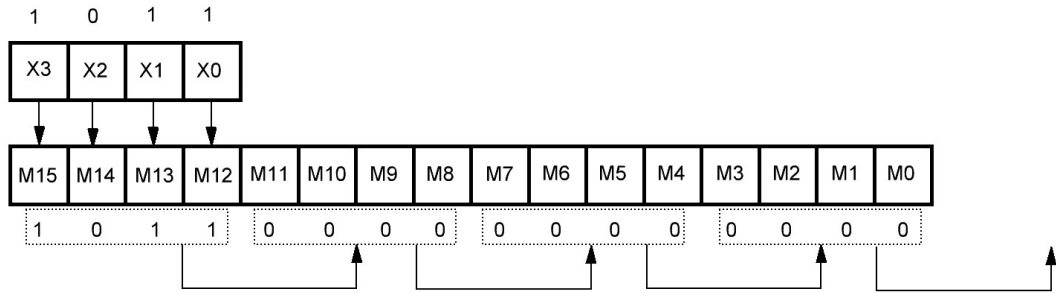


[SFTR]

Инструкция	SFTR (16-ти битная инструкция) SFTRP (16-ти битная инструкция, импульсное исполнение)		
Функция	Групповой сдвиг битовых значений вправо.		
Доступные входы	S1	X, Y, M, SM, S	Начальный адрес источника данных
	S2	Y, M, SM, S	Начальный адрес приемника данных
	S3	K(0~1024)	Количество сдвигаемых бит
	S4	K(0~1024)	Число позиций сдвига
Описание	<p>При выполнении данной инструкции происходит сдвиг бит вправо в количестве S3 на число позиций S4 и из источника с начальным адресом S1 переносятся данные в приемник с начальным адресом S2.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования.</p>		



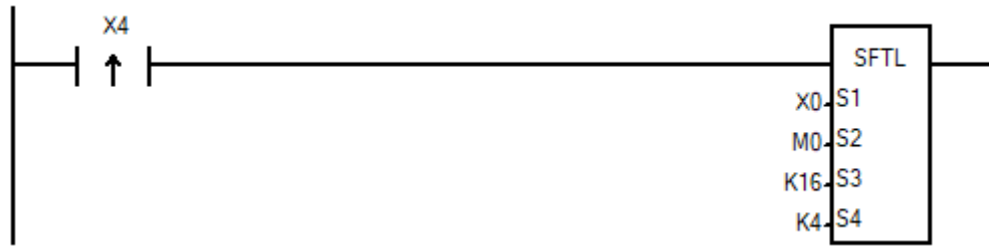
Когда X4 изменяет состояние с OFF на ON, биты M15-M0 сдвигаются на 4 позиции вправо, а значение X3-X0 помещается в биты M15-M12.



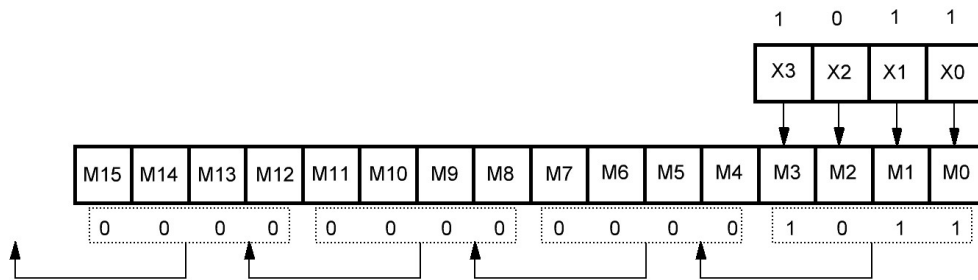
Данные в M15-M0 сдвигаются на 4-ре позиции вправо и значения X3-X0 помещаются в M15-M12

[SFTL]

Инструкция	SFTL (16-ти битная инструкция) SFTLP (16-ти битная инструкция, импульсное исполнение)		
Функция	Групповой сдвиг битовых значений влево.		
Доступные входы	S1	X, Y, M, SM, S	Начальный адрес источника данных
	S2	Y, M, SM, S	Начальный адрес приемника данных
	S3	K(0~1024)	Количество сдвигаемых бит
	S4	K(0~1024)	Число позиций сдвига
	Описание	<p>При выполнении данной инструкции происходит сдвиг бит влево в количестве S3 на число позиций S4 и из источника с начальным адресом S1 переносятся данные в приемник с начальным адресом S2.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования.</p>	



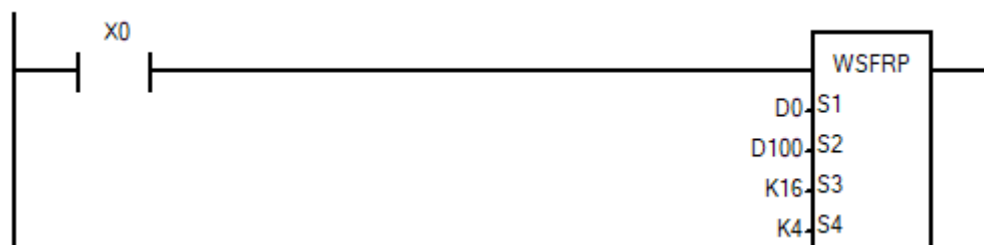
Когда X4 изменяет состояние с OFF на ON, биты M15-M0 сдвигаются на 4 позиции влево, а значения X3-X0 помещаются в биты M3-M0.



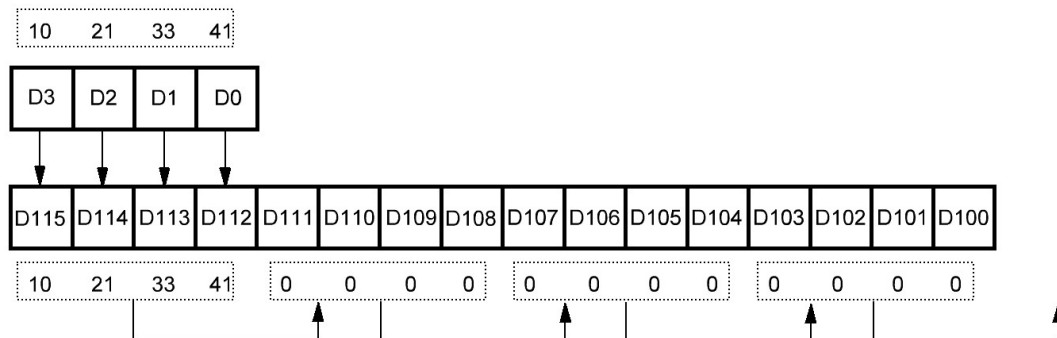
Данные в M0-M15 сдвигаются на 4-ре позиции влево и значения X3-X0 помещаются в M3-M0

[WSFR]

Инструкция	WSFR (16-ти битная инструкция) WSFRP (16-ти битная инструкция, импульсное исполнение)		
Функция	Пословный сдвиг регистров вправо		
Доступные входы			
	S1	T, C, D, SD	Начальный адрес источника данных
	S2	T, C, D, SD	Начальный адрес приемника данных
	S3	K(0~1024)	Количество сдвигаемых бит
	S4	K(0~1024)	Число позиций сдвига
Описание	<p>При выполнении инструкции регистры в количестве S3 сдвигаются вправо на число позиций S4 и из источника с начальным адресом S1 переносятся данные в приемник с начальным адресом S2.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования.</p>		



Когда X0 изменяет состояние с OFF на ON, данные в регистрах D100 – D115 сдвигаются на четыре позиции вправо и содержимое D3-D0 помещается в регистры D115-D112.

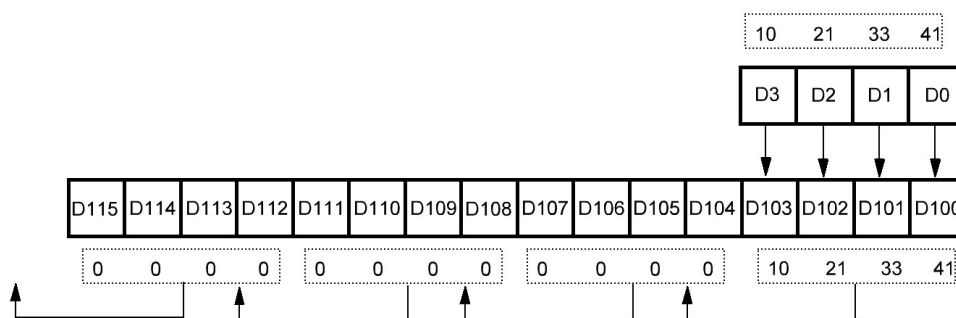


[WSFL]

Инструкция	WSFL (16-ти битная инструкция) WSFLP (16-ти битная инструкция, импульсное исполнение)		
Функция	Пословный сдвиг регистров влево		
Доступные входы	S1	T, C, D, SD	Начальный адрес источника данных
	S2	T, C, D, SD	Начальный адрес приемника данных
	S3	K(0~1024)	Количество сдвигаемых бит
	S4	K(0~1024)	Число позиций сдвига
	Описание	<p>При выполнении инструкции регистры в количестве S3 сдвигаются влево на число позиций S4 и из источника с начальным адресом S1 переносятся данные в приемник с начальным адресом S2.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то сдвиг будет осуществляться в каждом цикле сканирования.</p>	

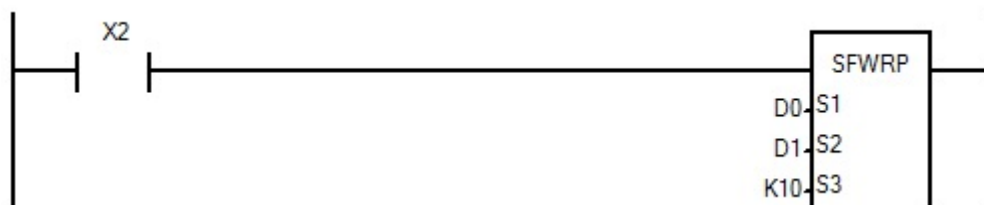


Когда X0 изменяет состояние с OFF на ON, данные в регистрах D100 – D115 сдвигаются на четыре позиции влево и содержимое D3-D0 помещается в регистры D103-D100.

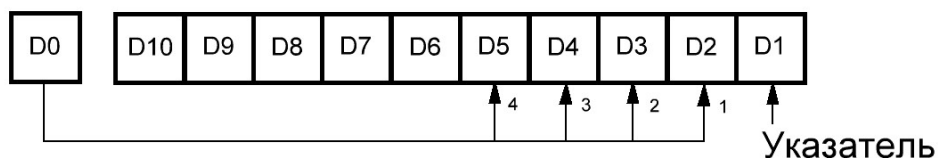


[SFWR]

Инструкция	SFWR (16-ти битная инструкция) SFWRP (16-ти битная инструкция, импульсное исполнение)		
Функция	Запись данных в стек		
Доступные входы			
	S1	T, C, D, SD, V, Z	Источник данных
	S2	T, C, D, SD	Начальный адрес стека данных
	S3	K(0~1024)	Глубина стека
Описание	<p>При выполнении инструкции происходит запись данных из источника S1 в приемник данных S2 длиной S3. Порядок сохранения данных FIFO (первый пришел – первый вышел); В первом элементе S2 хранится указатель стека.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то запись будет осуществляться в каждом цикле сканирования;</p> <p>Инструкция применяется совместно с SFRD.</p>		

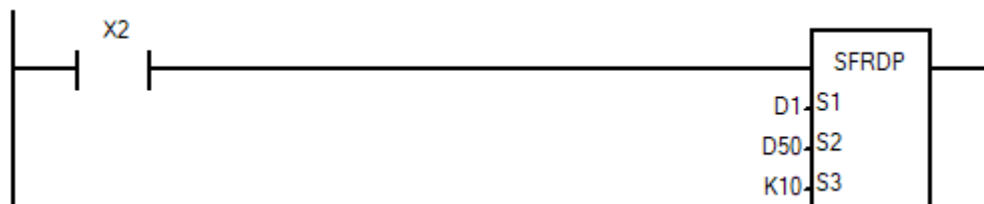


Когда X2 изменяет состояние с OFF на ON, содержимое D0 записывается в D2, а значение D1 принимает значение 1. Если содержимое D0 изменяется, то при следующей изменении состояния X0 с OFF на ON содержимое D0 записывается в D3 а значение D1 увеличивается на 1.

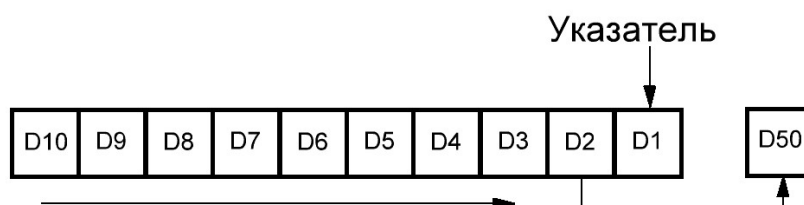


[SFRD]

Инструкция	SFRD (16-ти битная инструкция) SFRDP (16-ти битная инструкция, импульсное исполнение)		
Функция	Чтение данных из стека		
Доступные входы			
	S1	T, C, D, SD, V, Z	Начальный адрес стека данных
	S2	T, C, D, SD	Приемник данных
	S3	K(0~1024)	Глубина стека
Описание	<p>При выполнении инструкции происходит чтение данных из стековой памяти с начальным адресом S1 и длиной S3 в приемник S2. В первом элементе S1 хранится указатель стека. Если указатель стека станет равным 0, то включится флаг SM20;</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то чтение будет осуществляться в каждом цикле сканирования;</p>		



Когда X2 изменяет состояние с OFF на ON, значение из регистра D2 считывается в регистр D50, регистры D3...D10 смещаются вправо а значение D1 уменьшается на 1.

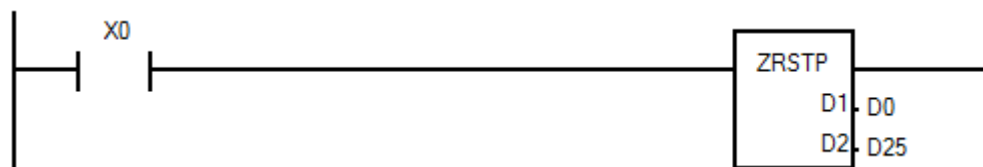


4.9 Инструкции обработки данных (Data Process Inst)

- ZRST / ZRSTP – Групповой сброс
- ENCO / ENCOF – Кодирование данных
- SUM / DSUM – Сумма активных битов
- BON / DBON - Проверка состояния бита
- MEAN / DMEAN – Среднее арифметическое
- ANS – Установка сигнала тревоги с задержкой включения
- ANR – Сброс сигнала тревоги
- SQR / DSQR – Вычисление квадратного корня
- FLT / DFLT – Преобразование целого числа в число с плавающей точкой
- SWAP / DSWAP – Перестановка байт в регистре
- SCALE / DSCALE - Пропорциональное масштабирование

[ZRST]

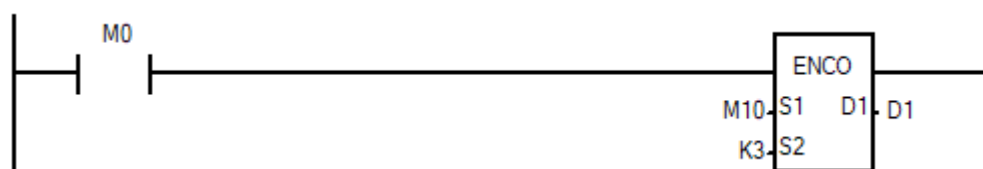
Инструкция	ZRST (16-ти битная инструкция) ZRSTP (16-ти битная инструкция, импульсное исполнение)	
Функция	Групповой сброс	
Доступные выходы	D1	Начальный элемент диапазона
	D2	Конечный элемент диапазона
Описание	<p>При выполнении инструкции происходит сброс состояния элементов в диапазоне от D1 до D2;</p> <p>Элементы D1 и D2 должны быть одного типа и одной разрядности;</p> <p>Должно выполняться условие $D1 < D2$;</p> <p>Обычно инструкцию используют в импульсном режиме.</p> <p>Если инструкция применяется не в импульсном режиме, то сброс будет осуществляться в каждом цикле сканирования.</p>	



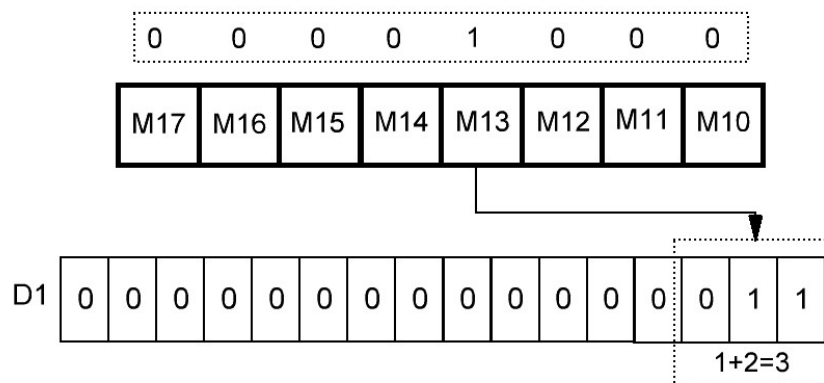
Когда X0 изменяет состояние с OFF на ON происходит сброс значений группы регистров начиная с D0 и заканчивая D25.

[ENCO]

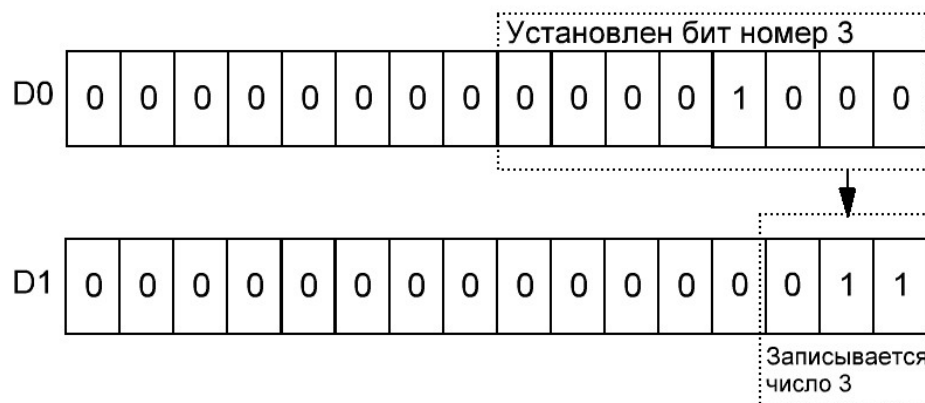
Инструкция	ENCO (16-ти битная инструкция) ENCOP (16-ти битная инструкция, импульсное исполнение)		
Функция	Кодирование данных		
Доступные входы			
	S1	K, X, Y, M, SM, S, T, C, D, SD, V, Z	Начальный элемент диапазона
	S2	K: 0...8	Количество элементов
Доступные выходы			
	D1	Y, M, SM, S, T, C, D, SD, AQ	Результат декодирования
Описание	<p>При выполнении инструкции проверяется порядковый номер установленного бита в источнике данных S1. Размерность источника данных задается в виде 2^{S2}.</p> <p>Если установлено несколько битов, то в приемник данных D1 записывается последний установленный бит.</p> <p>Если ни один из битов не установлен, на выходе инструкции значение 0. Такое же значение будет, если выбран 0 бит. Если установлено несколько битов, то на выходе инструкции номер старшего бита. Для исключения неопределенности в этих случаях рекомендуется использование совместно с инструкцией SUM.</p>		



Если в качестве S1 используются битовые значения и S1=M10, S2=3, D1=1, то в источнике данных с начальным адресом M10 в количестве 8-ми элементов ($2^3 = 8$) проверяется порядковый номер установленного бита и номер бита записывается в приемник D1. На примере ниже установлен бит M13(бит номер 3) и при выполнении инструкции в регистр D1 запишется число 3.

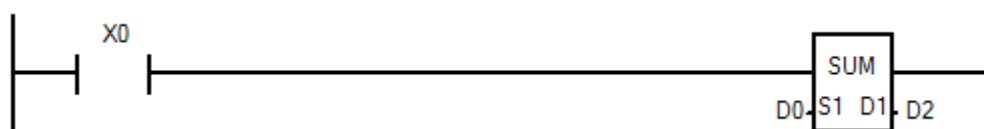


Если в качестве S1 используется регистр, то в приемник запишется номер активного бита в регистре источника данных S1.



[SUM / DSUM]

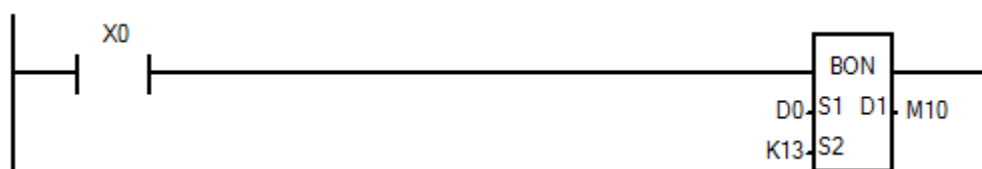
Инструкция	SUM (16-ти битная инструкция) SUMP (16-ти битная инструкция, импульсное исполнение) DSUM (32-х битная инструкция) DSUMP (32-х битная инструкция, импульсное исполнение)	
Функция	Сумма активных битов	
Доступные входы	S1	Т, С, D, SD, V, Z, K, KnX, KnY, KnM, KnSM, KnS
	Начальный адрес элемента	
Доступные выходы	D1	Y, M, SM, S, T, C, D, SD, AQ
	Количество активных битов	
Описание	При выполнении инструкции происходит подсчет суммы активных битов в S1 и результат помещается в D1. Биты, которые равны «0» на результат не влияют.	



Когда X0 включен, происходит подсчет битов равных "1" в регистре D0 и сумма помещается в D2.

[BON / DBON]

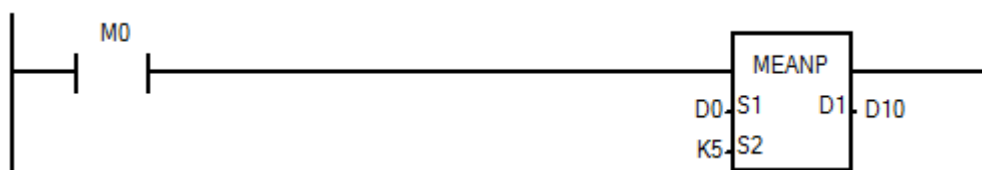
Инструкция	BON (16-ти битная инструкция) BONP (16-ти битная инструкция, импульсное исполнение) DBON (32-х битная инструкция) DBONP (32-х битная инструкция, импульсное исполнение)		
Функция	Проверка состояния бита		
Доступные входы	S1	T, C, D, SD, V, Z, K, KnX, KnY, KnM, KnSM, KnS	Проверяемый регистр
	S2	K	Номер проверяемого бита: 0..15 для 16-битной; 0..31 для 32-битной;
Доступные выходы	D1	Y, M, SM, S, T, C, D, SD, AQ	Состояние бита
Описание	При выполнении инструкции происходит проверка состояния бита номер S2 в регистре S1. Результат (состояние) помещается в D1.		



Когда X0 включен, происходит проверка состояния 13-го бита в регистре D0, результат помещается в M10.

[MEAN / DMEAN]

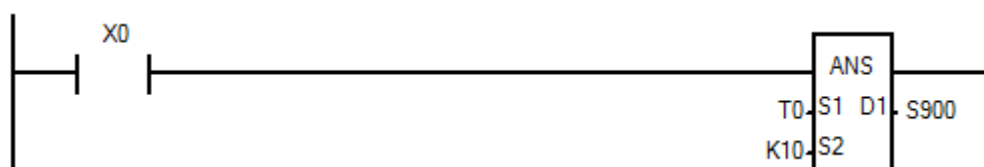
Инструкция	MEAN (16-ти битная инструкция) MEANP (16-ти битная инструкция, импульсное исполнение) DMEAN (32-х битная инструкция) DMEANP (32-х битная инструкция, импульсное исполнение)	
Функция	Среднее арифметическое	
Доступные входы	S1	T, C, D, SD, AI, AQ, KnX, KnY, KnM, KnSM, KnS Начальный адрес элемента
	S2	K: 1...64 Количество элементов
Доступные выходы	D1	T, C, D, SD, V, Z, AI, AQ, KnX, KnY, KnM, KnSM, KnS Среднеарифметическое значение
Описание	При выполнении инструкции происходит расчет среднеарифметического значения содержимого последовательных регистров начиная с S1 и количеством S2, результат помещается в D1. Результатом выполнения будет являться целое число, остаток от деления отбрасывается.	



Когда X0 включен, происходит расчет среднеарифметического значения регистров с D0 по D4, результат помещается в D10.

[ANS]

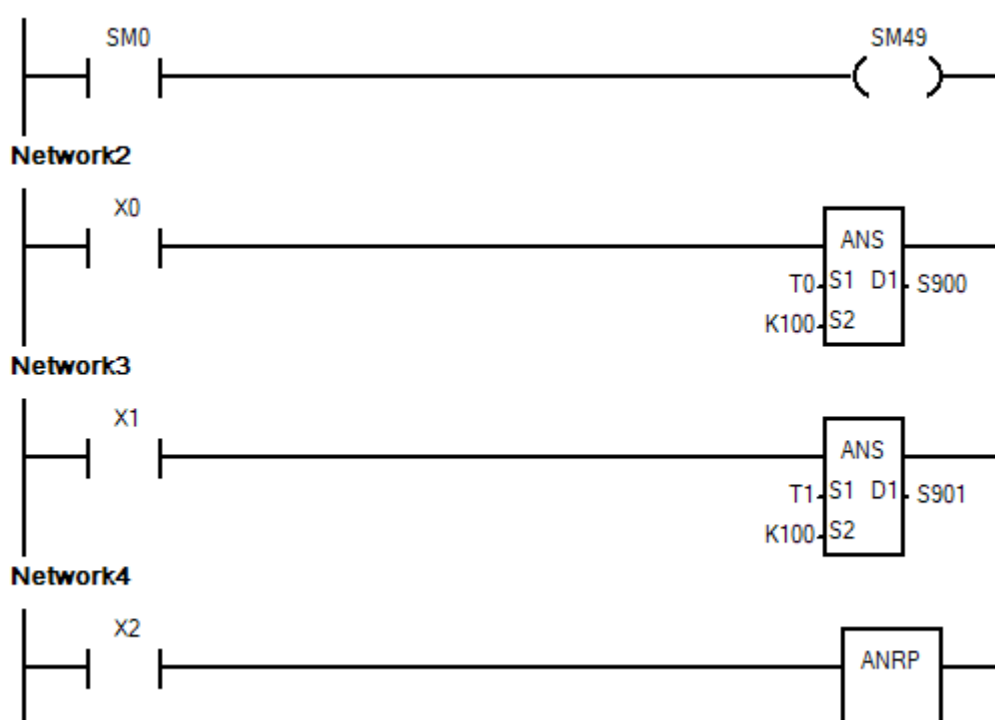
Инструкция	ANS (16-ти битная инструкция)		
Функция	Установка сигнала тревоги с задержкой включения		
Доступные входы	S1	T	Таймер задержки
	S2	K: -32768...65535	Время задержки таймера
Доступные выходы	D1	S	Сигнал тревоги
Описание	<p>Инструкция включает сигнал тревожной сигнализации D1 с задержкой. Задержка обеспечивается таймером ,указанным в S1 на время в S2.</p> <p>Если специальное реле управления сигнализацией SM49 включено, то при срабатывании любой из сигнализаций включится специальное реле SM48 а в регистре SD49 будет сохранен номер сигнализации с младшим номером.</p> <p>Инструкция обычно применяется совместно с инструкцией ANR.</p>		



Если X0 будет включен на время более 1 секунды (шаг таймера T0=100мс=0,1с), то установится сигнал S900. Когда X0 отключится, S900 останется включенным.

[ANR]

Инструкция	ANR (16-ти битная инструкция) ANRP(16-ти битная инструкция, импульсное исполнение)
Функция	Сброс сигнала тревоги
Доступные элементы	Отсутствуют
Описание	При выполнении инструкции происходит поочередный сброс сигналов тревоги установленных инструкцией ANS. Сначала сбрасывается тревога с меньшим номером. Обычно инструкцию используют в импульсном режиме.

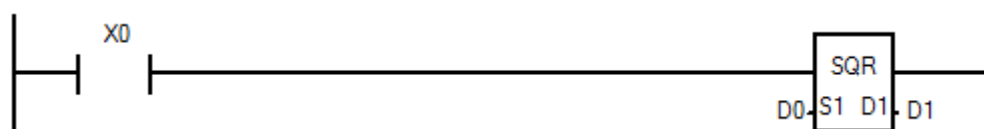


Если X0 будет включен более чем на 10 секунд, установится сигнал тревоги S900. Если X1 будет включен более чем на 10 секунд, установится сигнал тревоги S901.

При включении X2 сбросится сигнал тревоги S900. При повторном включении X2 сбросится сигнал тревоги S901.

[SQR / DSQR]

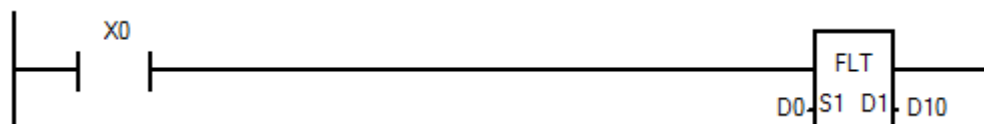
Инструкция	SQR (16-ти битная инструкция) SQRP(16-ти битная инструкция, импульсное исполнение) DSQR(32-х битная инструкция) DSQRP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычисление квадратного корня		
Доступные входы			
	S1	D, SD, AI, AQ, K	Число
Доступные выходы			
	D1	D, SD, AI, AQ	Квадратный корень из числа
Описание	<p>При выполнении инструкции происходит вычисление квадратного корня из числа в S1, результат помещается в D1.</p> <p>Выполнение возможно только если S1 положительное. Если S1 отрицательное, то включится флаг SM67 и инструкция не будет выполнена.</p> <p>Результатом операции может быть только целое число. Если результат не является целым числом, то знаки после запятой будут отброшены и включится флаг SM21.</p> <p>Если результат равен 0, то включится флаг SM20.</p>		



Когда X_0 включен происходит операция извлечения квадратного корня из значения в D_0 и результат помещается в D_1 .

[FLT / DFLT]

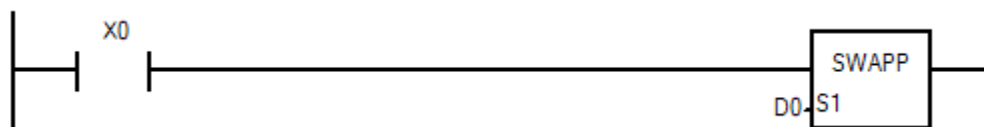
Инструкция	FLT (16-ти битная инструкция) FLTP (16-ти битная инструкция, импульсное исполнение) DFLT (32-х битная инструкция) DFLTP (32-х битная инструкция, импульсное исполнение)		
Функция	Преобразование целого числа в число с плавающей точкой		
Доступные входы			
	S1	D, SD, AI, AQ	Целое число (16 или 32-битное)
Доступные выходы			
	D1	D, SD, AI, AQ	Число с плавающей точкой (32-битное)
Описание	<p>При выполнении инструкции происходит преобразование двоичного целого числа S1 в двоичное число с плавающей точкой и результат помещается в S2;</p> <p>Константы K (DEC) и H (HEX) автоматически преобразуются в нужный формат если используется инструкция для работы с числами с плавающей точкой, поэтому в этой инструкции их использовать нельзя;</p> <p>Обратное преобразование осуществляется инструкцией INT.</p>		



Когда X0 включен, происходит преобразование целого числа регистра D0 в число с плавающей точкой и результат помещается в D10.

[SWAP / DSWAP]

Инструкция	SWAP (16-ти битная инструкция) SWAPP (16-ти битная инструкция, импульсное исполнение) DSWAP (32-х битная инструкция) DSWAPP (32-х битная инструкция, импульсное исполнение)	
Функция	Перестановка байт в регистре	
Доступные входы		
	S1	Т, С, D, SD, V, Z, AI, AQ, KnY, KnM, KnSM, KnS
Описание	<p>При выполнении 16-ти битной инструкции старший и младший байт в регистре меняются местами;</p> <p>При выполнении 32-х битной инструкции в каждом из 2-х регистров старшие и младшие байты меняются местами.</p> <p>Обычно инструкцию используют в импульсном режиме. Если инструкция применяется не в импульсном режиме, то перестановка будет осуществляться в каждом цикле сканирования.</p>	



При изменении состояния X0 с OFF на ON в регистре D0 (16 бит) младшие и старшие 8 бит меняются местами.

[SCALE / DSCALE]

Инструкция	SCALE (16-битная инструкция) SCALEP (16-битная инструкция, импульсное выполнение) DSCALE (32-битная инструкция) DSCALEP (32-битная инструкция, импульсное выполнение) DESCALE (32-битная инструкция для чисел с плавающей точкой) DESCALEP (32-битная инструкция для чисел с плавающей точкой, импульсное выполнение)		
Функция	Пропорциональное преобразование		
Доступные входы	S1	KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Входное значение
	S2		Нижний предел входного значения
	S3		Верхний предел входного значения
	S4		Нижний предел выходного значения
	S5		Верхний предел выходного значения
	Доступные выходы		
D1	T, C, D, SD, V, Z, AQ	Выходное значение	
Описание	Преобразует входной сигнал в требуемый диапазон по формуле $D1 = (S1-S2) / (S3-S2) * (S5-S4) + S4$		



Когда X0 включен инструкция производит преобразование входного значения A10, которое может изменяться в диапазоне от 4000 до 20000, в

пользовательский диапазон от 0 до 1000. Преобразованное значение записывается в регистр D0.

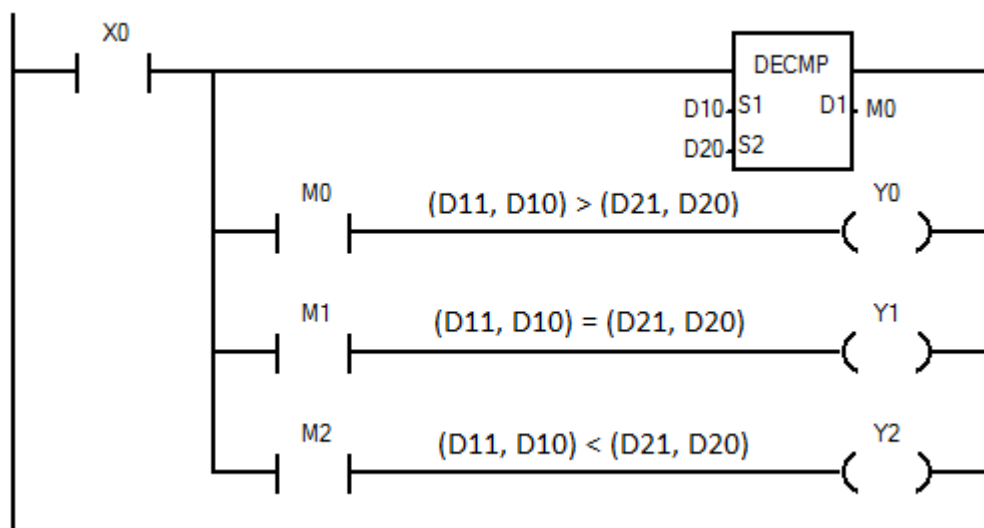
Данная инструкция является полезной при преобразовании сигналов с аналоговых датчиков в пользовательский формат. К примеру, перевод сигнала 4...20 мА в диапазон 0...10 бар.

4.10 Инструкции для чисел с плавающей точкой (Float Process Inst)

- **DECMP** – Сравнение двух чисел с плавающей точкой
- **DEZCP** – Зонное сравнение двух чисел с плавающей точкой.
- **DEBCD** – Преобразование двоичного числа с плавающей точкой в десятичное с плавающей точкой
- **DEBIN** – Преобразование десятичного числа с плавающей точкой в двоичное с плавающей точкой
- **DEADD** – Сложение чисел с плавающей точкой
- **DESUB** – Вычитание чисел с плавающей точкой
- **DEMUL** – Умножение чисел с плавающей точкой
- **DEDIV** – Деление чисел с плавающей точкой
- **DESOR** – Вычисление квадратного корня в формате числа с плавающей точкой
- **INT / DINT** – Преобразование числа с плавающей точкой в целое
- **DESIN** – Вычисление синуса в формате числа с плавающей точкой
- **DECOS** – Вычисление косинуса в формате числа с плавающей точкой
- **DETAIL** – Вычисление тангенса в формате числа с плавающей точкой
- **DEPOW** – Возведение числа в степень в формате с плавающей точкой

[DECMP]

Инструкция	DECMP (32-х битная инструкция) DECMPР (32-х битная инструкция, импульсное исполнение)	
Функция	Сравнение двух чисел с плавающей точкой.	
Доступные входы	S1	D, SD, K
	S2	
Доступные выходы	D1	Y, M, SM, S
	Результат сравнения	
Описание	<p>При выполнении инструкции сравниваются числа с плавающей точкой в S1 и S2 и результат помещается в D. Элемент D занимает три последовательных значения - D, D+1, D+2.</p> <p>Если $S1 > S2$ то включается D; Если $S1 = S2$, то включается D+1; Если $S1 < S2$, то включается D+2;</p>	

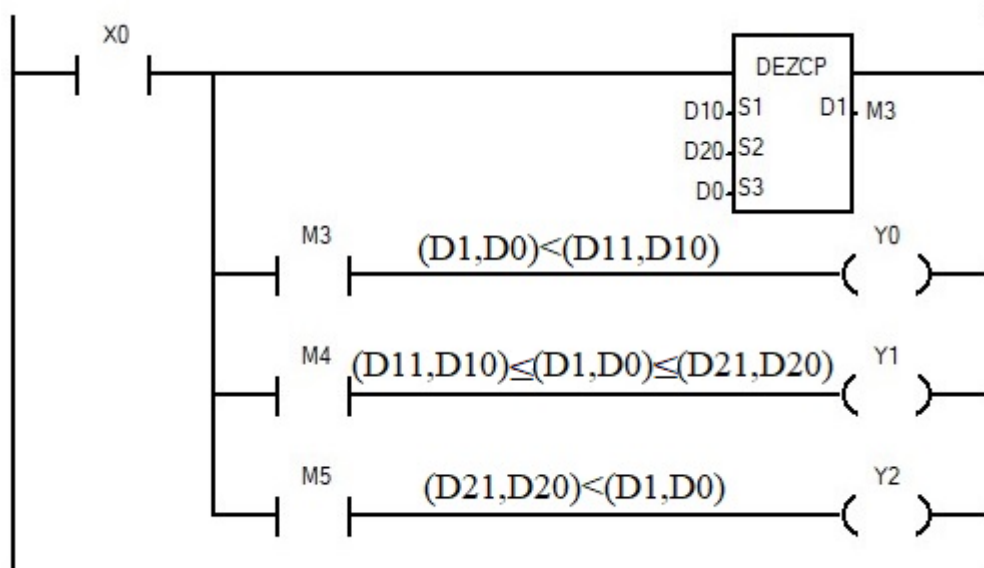


Когда X0 включен, то происходит сравнение двух чисел с плавающей точкой (D11, D10) и (D21, D20):

- если $(D11, D10) > (D21, D20)$, то включится выход Y0;
- если $(D11, D10) = (D21, D20)$, то включится выход Y1;
- если $(D11, D10) < (D21, D20)$, то включится выход Y2.

[DEZCP]

Инструкция	DEZCP (32-х битная инструкция) DEZCPP (32-х битная инструкция, импульсное исполнение)		
Функция	Зонное сравнение двух чисел с плавающей точкой		
Доступные входы	S1	D, SD, K	Начальное значение интервала
	S2		Конечное значение интервала
	S3		Число для сравнения
Доступные выходы	D1	Y, M, SM, S	Результат сравнения
	<p>Описание</p> <p>При выполнении инструкции сравниваются значения в S3 со значением интервала, заданными S1 и S2, результат помещается в D1. Элемент D1 занимает три последовательных значения D1, D1+1, D1+2.</p> <p>S1 должен быть меньше S2;</p> <ul style="list-style-type: none"> - если $S3 < S1$ то включается D1; - если $S1 \leq S3 \leq S2$, то включается D1+1; - если $S3 > S2$, то включается D1+2; 		

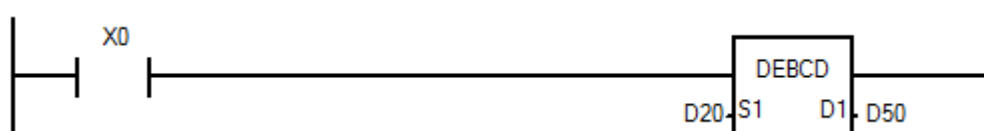


Когда X0 включен, происходит сравнение значения числа с плавающей точкой в (D1, D0) с числами с плавающей точкой в (D11, D10) и (D21, D20):

- если $(D1, D0) < (D11, D10)$, то включится выход Y0;
- если $(D11, D10) \leq (D1, D0) \leq (D21, D20)$, то включится выход Y1;
- если $(D21, D20) < (D1, D0)$, то включится выход Y2.

[DEBCD]

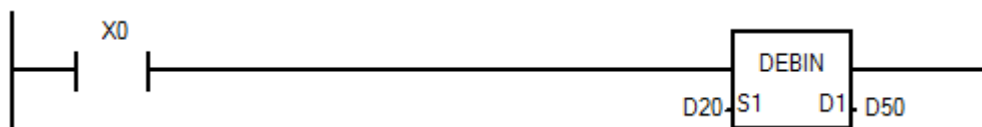
Инструкция	DEBCD (32-х битная инструкция) DEBCDP (32-х битная инструкция, импульсное исполнение)	
Функция	Преобразование двоичного числа с плавающей точкой в десятичное с плавающей точкой	
Доступные входы		
	S1	D, SD
Доступные выходы		
	D1	D, SD
Описание	При выполнении инструкции происходит преобразование двоичного целого числа с плавающей точкой S1 в десятичное число с плавающей точкой и результат помещается в D1.	



Когда X0 включен, двоичное число с плавающей точкой в D20 преобразуется в десятичное число с плавающей точкой и помещается в регистр D50.

[DEBIN]

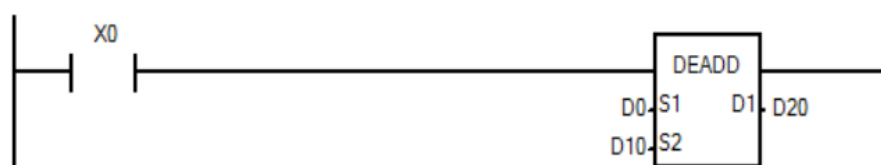
Инструкция	DEBIN (32-х битная инструкция) DEBINP (32-х битная инструкция, импульсное исполнение)	
Функция	Преобразование десятичного числа с плавающей точкой в двоичное с плавающей точкой	
Доступные входы	S1	Десятичное число с плавающей точкой
	D, SD	
Доступные выходы	D1	Двоичное число с плавающей точкой
	D, SD	
Описание	При выполнении инструкции происходит преобразование десятичного числа с плавающей точкой S1 в двоичное число с плавающей точкой и результат помещается в D1	



Когда X0 включен, десятичное число с плавающей точкой в формате $D1 \times 10^{(D1+1)}$ в D20, D21 преобразуется в двоичное число с плавающей точкой и помещается в регистр D50, D51.

[DEADD]

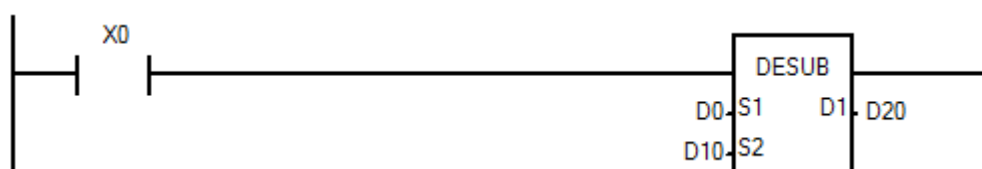
Инструкция	DEADD (32-х битная инструкция) DEADDP (32-х битная инструкция, импульсное исполнение)		
Функция	Сложение чисел с плавающей точкой		
Доступные входы	S1	D, SD, K	Первое слагаемое
	S2		Второе слагаемое
Доступные выходы	D1	D, SD	Сумма
Описание	<p>Инструкция суммирует в двоичном формате S1 и S2 и сохраняет результат в D1.</p> <p>Если результат расчета равен 0, будет установлен флаг нуля SM20;</p> <p>Если результат вычисления превышает максимальное значение для числа с плавающей точкой, будет установлен флаг переноса SM22;</p> <p>Если результат вычисления меньше минимального значения для числа с плавающей точкой, будет установлен флаг заимствования SM21;</p>		



Когда X0 включен, числа с плавающей точкой в D0 и D10 суммируются, результат помещается в регистр D20.

[DESUB]

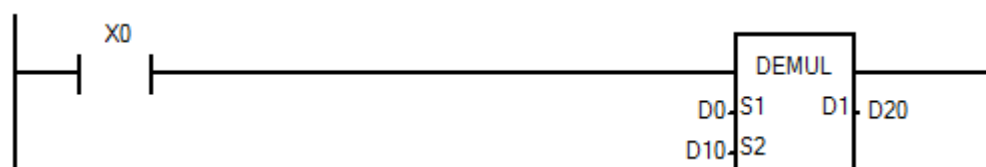
Инструкция	DESUB (32-х битная инструкция) DESUBP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычитание чисел с плавающей точкой		
Доступные входы	S1	D, SD, K	Уменьшаемое
	S2		Вычитаемое
Доступные выходы	D1	D, SD	Разность
Описание	<p>Инструкция вычисляет разность чисел с плавающей точкой S1 и S2 и сохраняет результат в D1.</p> <p>Если результат расчета равен 0, будет установлен флаг нуля SM20;</p> <p>Если результат вычисления превышает максимальное значение для числа с плавающей точкой, будет установлен флаг переноса SM21;</p> <p>Если результат вычисления меньше минимального значения для числа с плавающей точкой, будет установлен флаг заимствования SM22;</p>		



Когда X0 включен, из числа с плавающей точкой D0 вычитается число с плавающей точкой D10 и результат помещается в регистр D20.

[DEMUL]

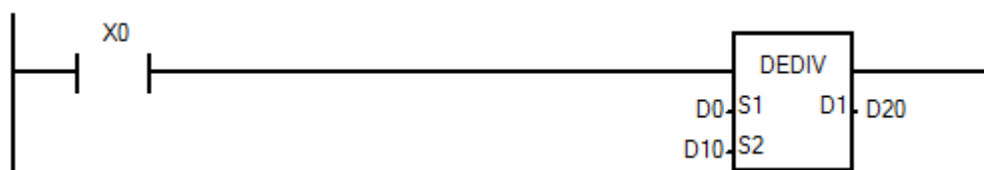
Инструкция	DEMUL (32-х битная инструкция) DEMULP (32-х битная инструкция, импульсное исполнение)		
Функция	Умножение чисел с плавающей точкой		
Доступные входы	S1	D, SD, K	Первый множитель
	S2		Второй множитель
Доступные выходы	D1	D, SD	Произведение
Описание	<p>Инструкция вычисляет произведение чисел с плавающей точкой S1 и S2 и сохраняет результат в D1. Если значение результата превышает максимальное значение числа с плавающей точкой, то включается флаг переноса SM22. Если значение результата меньше минимального значения числа с плавающей точкой, то включается флаг заимствования SM21. Если результат равен 0, включится флаг SM20.</p>		



Когда X0 включен, первое число с плавающей точкой D0 умножается на второе число с плавающей точкой D10 и результат помещается в регистр D20.

[DEDIV]

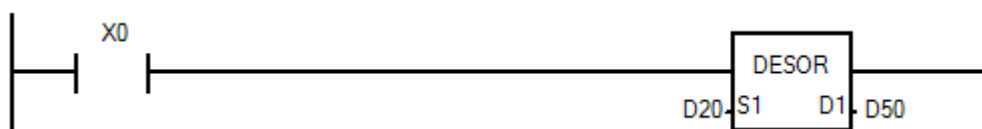
Инструкция	DEDIV (32-х битная инструкция) DEDIVP (32-х битная инструкция, импульсное исполнение)		
Функция	Деление чисел с плавающей точкой		
Доступные входы	S1	D, SD, K	Делимое
	S2		Делитель
Доступные выходы	D1	D, SD	Частное
Описание	<p>Инструкция осуществляет операцию деления числа с плавающей точкой S1 на число с плавающей точкой S2 и сохраняет результат в D1;</p> <p>Если результат расчета равен 0, будет установлен флаг нуля SM20;</p> <p>Если результат вычисления превышает максимальное значение для числа с плавающей точкой, будет установлен флаг переноса SM22;</p> <p>Если результат вычисления меньше минимального значения для числа с плавающей точкой, будет установлен флаг заимствования SM21;</p> <p>Если S2 равен 0, то возникает ошибка и инструкция не выполняется.</p>		



Когда X0 включен, число с плавающей точкой D0 делится на число с плавающей точкой D10, результат помещается в регистр D20.

[DESOR]

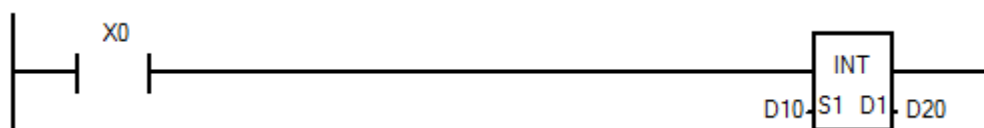
Инструкция	DESOR (32-х битная инструкция) DESORP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычисление квадратного корня в формате числа с плавающей точкой		
Доступные входы			
	S1	D, SD, K	Число
Доступные выходы			
	D1	D, SD	Квадратный корень из числа
Описание	<p>При выполнении инструкции происходит вычисление квадратного корня из числа с плавающей точкой S1, результат помещается в S2;</p> <p>Выполнение возможно только если S1 положительное. Если S1 отрицательное, то включится флаг SM67 и инструкция не будет выполнена;</p>		



Когда X0 включен происходит операция извлечения квадратного корня из числа с плавающей точкой D20, результат помещается в D50.

[INT / DINT]

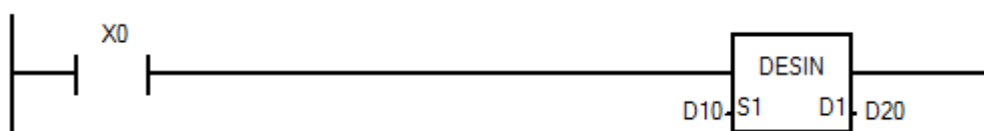
Инструкция	INT (16-ти битная инструкция) INTP (16-ти битная инструкция, импульсное исполнение) DINT (32-х битная инструкция) DINTP (32-х битная инструкция, импульсное исполнение)		
Функция	Преобразование числа с плавающей точкой в целое		
Доступные входы			
	S1	D, SD, K	Число с плавающей точкой (32-бит)
Доступные выходы			
	D1	D, SD	Целое число 16/32-битное
Описание	<p>При выполнении инструкции происходит преобразование двоичного числа с плавающей точкой S1 в десятичное D1; Результатом преобразования всегда будет целое число, знаки после запятой отбрасываются;</p> <p>Если результат преобразования равен 0, будет установлен флаг нуля SM20;</p> <p>Если результат преобразования выходит за диапазоны: для 16-ти битной инструкции -32768~32767 для 32-х битной инструкции -2147483648~2147483647 то будет включен флаг переноса SM22;</p> <p>Если в результате преобразования были отброшены десятичные знаки, то будет установлен флаг заимствования SM21;</p>		



Когда X0 включен, двоичное число с плавающей точкой в D10 преобразуется в целое десятичное число и помещается в D20.

[DESIN]

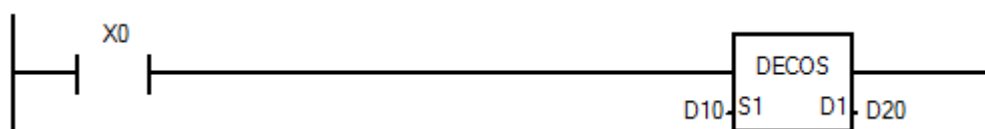
Инструкция	DESIN (32-х битная инструкция) DESINP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычисление синуса в формате числа с плавающей точкой		
Доступные входы			
	S1	D, SD, K	Радианы
Доступные выходы			
	D1	D, SD	Значение синуса
Описание	При выполнении инструкции происходит вычисление синуса в формате с плавающей точкой. Значение задается в радианах(RAD). $RAD = \alpha * \pi / 180^\circ$.		



Когда X0 включен, происходит вычисление синуса из значения в D10, результат помещается в D20.

[DECOS]

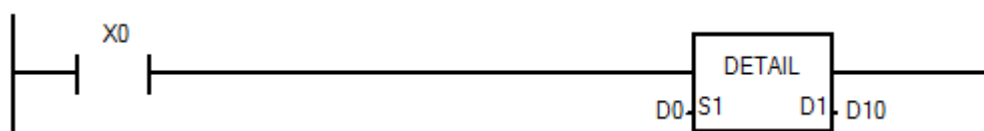
Инструкция	DECOS (32-х битная инструкция) DECOSP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычисление косинуса в формате числа с плавающей точкой		
Доступные входы			
	S1	D, SD, K	Радианы
Доступные выходы			
	D1	D, SD	Значение косинуса
Описание	При выполнении инструкции происходит вычисление косинуса в формате с плавающей точкой. Значение задается в радианах(RAD). $RAD = \alpha * \pi / 180^\circ$.		



Когда X0 включен, происходит вычисление косинуса из значения в D10, результат помещается в D20.

[DETAIL]

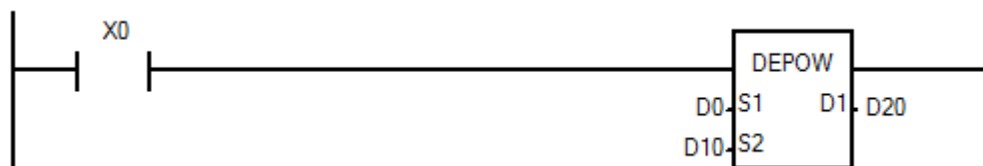
Инструкция	DETAIL (32-х битная инструкция) DETAILP (32-х битная инструкция, импульсное исполнение)		
Функция	Вычисление тангенса в формате числа с плавающей точкой		
Доступные входы			
	S1	D, SD, K	Радианы
Доступные выходы			
	D1	D, SD	Тангенс
Описание	При выполнении инструкции происходит вычисление тангенса в формате с плавающей точкой. Значение задается в радианах(RAD). $RAD = \alpha * \pi / 180^\circ$.		



Когда X0 включен, происходит вычисление тангенса из значения в D0, результат помещается в D10.

[DEPOW]

Инструкция	DEPOW (32-х битная инструкция) DEPOWP (32-х битная инструкция, импульсное исполнение)		
Функция	Возведение числа в степень в формате с плавающей точкой		
Доступные входы	S1	D, SD, K	Число
	S2		Степень
Доступные выходы	D1	D, SD	Результат возведения в степень
Описание	При выполнении инструкции происходит возведение числа S1 в степень S2; При возведении числа S1 в степень 0 результат будет равен 1.		



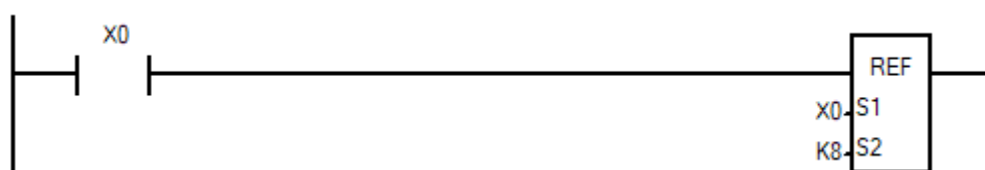
Когда X0 включен, число с плавающей точкой D0 возводится в степень D10, результат помещается в D20.

4.11 Высокоскоростные инструкции (High-speed inst)

- REF – Обновление состояния входов и выходов
- REFF / REFFP – Изменение времени фильтрации входов
- SPD – Подсчет количества импульсов за интервал времени
- PLSY / DPLSY – Формирование заданного количества импульсов с заданной частотой
- PWM – Широтно-импульсная модуляция
- PLSR / DPLSR – Импульсный выход с ускорением и замедлением
- PLSV / DPLSV – Импульсный выход с выбором направления движения
- PWMR – Широтно-импульсная модуляция с заданным коэффициентом заполнения
- PWMY / DPWMY – Выдача заданного количество импульсов с широтно-импульсной модуляцией и с заданным коэффициентом заполнения
- ZNR / DZNR – Возврат в нулевую точку
- DHSCT – Прерывание по совпадению счетчика
- DSZR – Возврат в нулевую точку с поиском DOG (сигнала зоны остановки)
- DRVI / DDRVI – Односкоростное позиционирование в относительных координатах
- DRVA / DDRVA – Односкоростное позиционирование в абсолютных координатах
- PLSTR / DPLSTR – Выдача импульсов с ускорением и замедлением
- PSTOP – Остановка работы импульсного выхода

[REF]

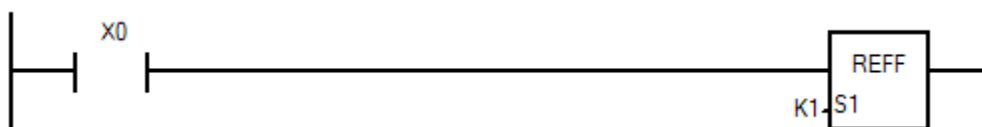
Инструкция	REF (16-ти битная инструкция) REFP (16-ти битная инструкция, импульсное исполнение)		
Функция	Обновление состояния входов и выходов		
Доступные входы			
	S1	X, Y	Начальный номер входа/выхода
	S2	K (0~1024)	Количество входов/выходов
Доступные выходы			
	D1	Y, M, SM, S	Результат сравнения
Описание	<p>При выполнении инструкции происходит обновление состояния входов/выходов S1 в количестве S2 вне зависимости от цикла сканирования.</p> <p>Инструкцию можно использовать между инструкциями FOR~NEXT и между инструкциями CJ, в подпрограмме прерывания;</p> <p>Фактическая задержка изменения состояния входов зависит от времени фильтрации и устанавливается в диапазоне 0~60 мс (инструкция REFF), другие порты ввода-вывода фильтруются аппаратно, время фильтрации составляет 10 мс;</p> <p>Фактическая задержка изменения состояния выходов определяется временем срабатывания выходного элемента.</p>		



Когда X0 включен, состояние входов X0...X7 обновляются сразу, не дожидаясь начала нового цикла программы.

[REFF]

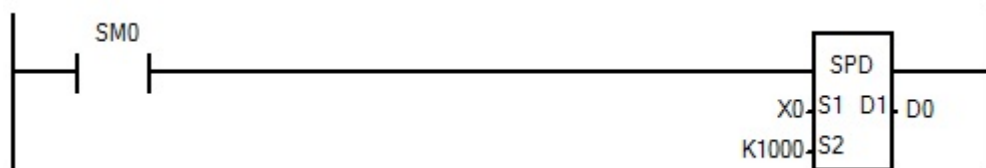
Инструкция	REFF (16-ти битная инструкция) REFFP (16-ти битная инструкция, импульсное исполнение)		
Функция	Изменение времени фильтрации входов		
Доступные входы			
	S1	K (0~60)	Время фильтрации
Описание	При выполнении инструкции время фильтрации всех дискретных входов становится равным S1, значение задается в миллисекундах; Если инструкция не используется, то время фильтрации равно 10 мс; Если используются прерывания по входу, высокоскоростные счетчики или инструкция SPD, то для задействованных входов инструкция REFF не действует. Время фильтрации так же можно изменить записав необходимое значение в специальный регистр SD20.		



Когда X0 замкнут, время фильтрации всех дискретных входов становится равным 1 мс.

[SPD]

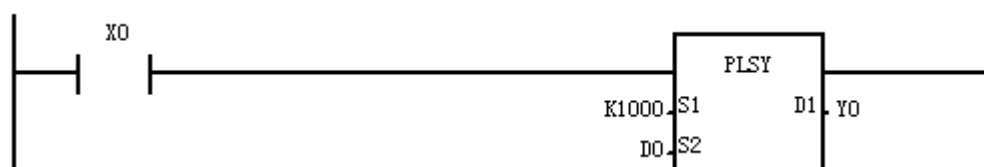
Инструкция	SPD (16-ти битная инструкция)		
Функция	Подсчет количества импульсов за интервал времени		
Доступные входы	S1	X	Вход для приема импульсов
	S2	K, T, C, D, SD, V, Z, KnX, KnY, KnM, KnSM, KnS	Интервал времени
Доступные выходы	D1	T, C, D, SD, V, Z	Количество импульсов за интервал
Описание	<p>При выполнении инструкции происходит подсчет количества импульсов на S1 за время S2, результат сохраняется в D1. Время задается в миллисекундах. Регистр хранения результата занимает 3 последовательных элемента: D1, D1+1, D1+2.</p> <p>D1 – количество импульсов за заданное время; D1+1- сумма импульсов внутри интервала времени. После истечения времени подсчета результат сохраняется в D1 а D1+1 сбрасывается; D1+2- отображение оставшегося времени до конца интервала.</p>		



Когда ПЛК находится в режиме «Работа» производится подсчет количества импульсов на входе X0 за 1 секунду (т. е. частота импульсов), результат сохраняется в D0.

[PLSY / DPLSY]

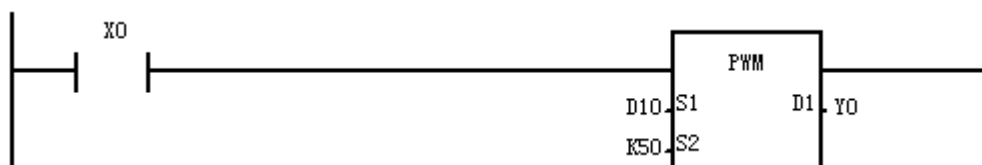
Инструкция	PLSY (16-ти битная инструкция) DPLSY (32-х битная инструкция)		
Функция	Формирование заданного количества импульсов с заданной частотой		
Доступные входы	S1	K, T, C, D, SD, V, Z, KnX, KnY, KnM, KnSM, KnS	Частота импульсов, Гц
	S2		Количество импульсов
Доступные выходы	D1	Y	Используемый выход
Описание	Инструкция формирует количество импульсов S2 с частотой S1 на выходе D1. Доступны выходы Y0...Y1; Импульсы формируются с коэффициентом заполнения 0,5; Выполнение инструкции не зависит от времени цикла.		



Когда X0 замкнут, на выходе Y0 формируются импульсы с частотой 1кГц, количество импульсов задается в D0.

[PWM]

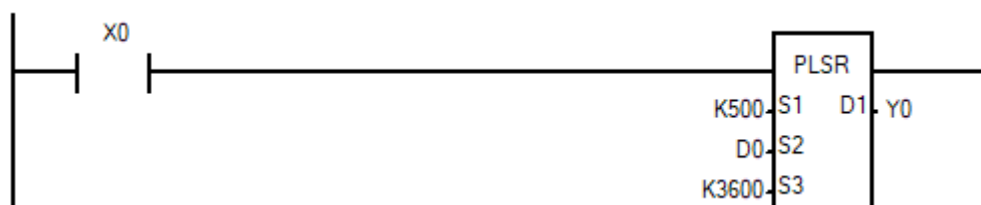
Инструкция	PWM (16-ти битная инструкция)		
Функция	Широтно-импульсная модуляция		
Доступные входы			
	S1	K, T, C, D, SD, V, Z, KnX, KnY, KnM, KnSM, KnS	Ширина импульса
	S2		Период
Доступные выходы			
	D1	Y	Используемый выход
Описание	<p>При выполнении инструкции на выходе D1 происходит формирование импульсов шириной S1 и периодом S2; Должно выполняться условие $S1 \leq S2$; Время задается в миллисекундах в диапазоне 0...32767. В качестве выходов могут использоваться только Y0 и Y1.</p>		



Когда X0 замкнут на выходе Y0 формируются импульсы шириной D10, и периодом ШИМ в 50 мс. Если период ШИМ 50 мс, то заданием в D10 числа в диапазоне 0-50 можно изменять ширину импульса от 0 до 100%.

[PLSR / DPLSR]

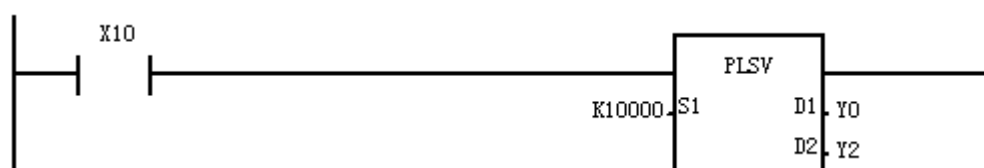
Инструкция	PLSR (16-ти битная инструкция) DPLSR (32-х битная инструкция)		
Функция	Импульсный выход с ускорением/замедлением		
Доступные входы	S1	K, T, C, D, SD, V, Z, KnX, KnY, KnM, KnSM, KnS	Заданная частота
	S2		Количество выдаваемых импульсов
	S3		Время разгона и замедления
Доступные выходы	D1	Y	Используемый выход
Описание	<p>Инструкция формирует на выходе D1 импульсы с максимальной частотой S1 в количестве S2. Частота плавно изменяется до S1 за время S3. При достижении максимальной частоты происходит плавное снижение частоты за время S3.</p> <p>S1- максимальная частота, диапазон от 10 Гц до 100 кГц; S2- количество выдаваемых импульсов, диапазон от 110 до 32767 для 16-ти битной инструкции и от 110 до 2147483647 для 32-х битной инструкции; S3- время разгона и замедления, диапазон 50...5000 миллисекунд;</p> <p>Инструкция должна использоваться только для ПЛК с транзисторными выходами. В качестве выходов могут использоваться только Y0 и Y1.</p>		



Когда X0 включен, на выходе Y0 формируются импульсы с плавно нарастающей частотой до 500 Гц за 3600 миллисекунд, затем происходит плавное снижение за тоже время. Количество импульсов задается в D0.

[PLSV / DPLSV]

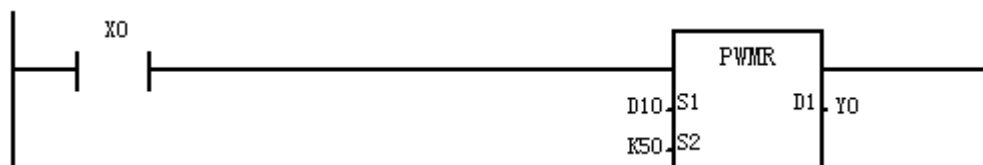
Инструкция	PLSV (16-ти битная инструкция) DPLSV (32-х битная инструкция)		
Функция	Импульсный выход с выбором направления движения.		
Доступные входы			
	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Заданная частота, Гц
Доступные выходы			
	D1	Y	Используемый выход
	D2	Y, M, SM, S	Индикация направления
Описание	<p>При выполнении инструкции происходит выдача импульсов с частотой S1 на выходе D1. Если частота - положительное число то D2 включен. Если частота – отрицательное число то D2 выключен;</p> <p>Диапазон задания частоты в S1 для 16-ти битной инструкции 1...32767 и -1...-32767;</p> <p>Диапазон задания частоты в S1 для 32-х битной инструкции 1...100000 и -1...-100000;</p> <p>В качестве D1 могут использоваться только выходы Y0 и Y1;</p> <p>Инструкция должна использоваться только для ПЛК с транзисторными выходами.</p>		



Когда X10 замкнут, на выходе Y0 генерируются импульсы с частотой 10 кГц, при этом Y2 включен, тем самым показывая, что текущее направление - вперед.

[PWMR]

Инструкция	PWMR (16-ти битная инструкция)		
Функция	Широтно-импульсная модуляция с заданным коэффициентом заполнения.		
Доступные входы	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Частота импульсов, Гц
	S2		Коэффициент заполнения
Доступные выходы	D1	Y	Используемый выход
	<p>Описание</p> <p>При выполнении инструкции происходит генерация импульсов на выходе D1 с частотой S1 и коэффициентом заполнения S2. S1- частота импульсов, диапазон 0...32767; S2- коэффициент заполнения S2/1000, диапазон 0...1000; Инструкция должна использоваться только для ПЛК с транзисторными выходами. В качестве D1 можно использовать только Y0 и Y1.</p>		

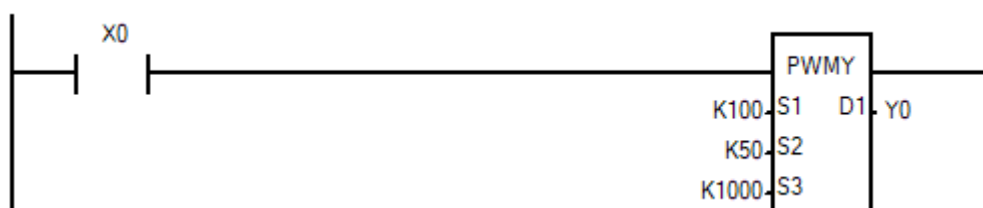


Когда X0 замкнут на выходе Y0 генерируются импульсы с частотой, заданной в регистре D10 с коэффициентом заполнения 50.

Коэффициент заполнения это отношение длительности импульса к периоду следования импульсов. В данном случае это безразмерная величина, но удобнее пользоваться процентами – так, значение S2=50, означает 5% заполнения, т. е. длина импульса по времени составляет 5% от периода. Если S2=500, то длительность импульса равна половине периода.

[PWMY / DPWMY]

Инструкция	PWMY (16-ти битная инструкция) DPWMY (32-х битная инструкция)		
Функция	Выдача заданного количество импульсов с широтно-импульсной модуляцией и с заданным коэффициентом заполнения.		
Доступные входы	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Частота импульсов, Гц
	S2		Коэффициент заполнения
	S3		Заданное количество импульсов
Доступные выходы	D1	Y	Используемый выход
Описание	<p>При выполнении инструкции происходит генерация импульсов на выходе D1 в количестве S3, с частотой S1 и коэффициентом заполнения S2.</p> <p>S1- частота импульсов. Диапазон задания частоты в S1 для 16-ти битной инструкции 0...65535 Гц; Диапазон задания частоты в S1 для 32-х битной инструкции 0...2147483647 Гц; S2- коэффициент заполнения S2/1000, диапазон 0...1000; S3- количество импульсов. Если S3=0 то импульсы генерируются постоянно. Диапазон задания количества импульсов в S3 для 16-ти битной инструкции 0...65535; Диапазон задания количества импульсов в S3 для 32-х битной инструкции 0...2147483647; Инструкция должна использоваться только для ПЛК с транзисторными выходами. В качестве D1 можно использовать только Y0 и Y1</p>		



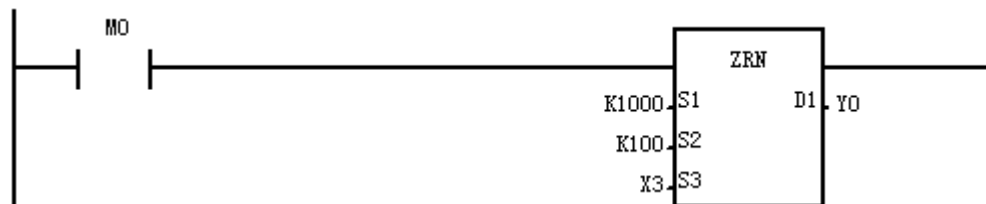
Когда X0 замкнут, на выходе Y0 генерируются 1000 импульсов с частотой 100 Гц и коэффициентом заполнения 50. Работа инструкции аналогична PWMR, но с возможностью задать количество выдаваемых импульсов.

[ZRN / DZRN]

Инструкция	ZNR (16-ти битная инструкция) DZNR (32-х битная инструкция)		
Функция	Возврат в нулевую точку		
Доступные входы			
	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Частота импульсов при возврате в нулевую точку (основная скорость)
	S2		Пониженная частота после поступления сигнала DOG (ползучая скорость)
	S3	X, Y, M, SM, S	Элемент для подачи сигнала нулевой точки (DOG)
Доступные выходы			
	D1	Y	Импульсный выход
Описание	<p>Когда выполняются инструкции управления относительной позицией DRVI и управления абсолютной позицией DRVA контроллер вычисляет количество импульсов вращения вперед или назад и сохраняет их в регистрах [SD141, SD140] (Y0) и [SD143, SD142] (Y1). Но данные в этих регистрах не сохраняются при отключении питания, поэтому инструкция возврата в нулевую точку ZRN должна быть выполнена при включении питания.</p> <p>S1- Начальная частота импульсов при возврате в нулевую точку. Диапазон задания частоты для 16-ти битной инструкции 10...32767 Гц; Диапазон задания частоты для 32-х битной инструкции 10...100000 Гц;</p> <p>S2- Пониженная частота после поступления сигнала DOG (DOG- сигнал входа в зону останова). Диапазон задания 10...32767;</p> <p>S3- Элемент для подачи сигнала нулевой точки (DOG). Наилучшее быстродействие обеспечивается когда в качестве сигнала используется физический вход ПЛК(X);</p> <p>D1- Импульсный выход, могут использоваться только Y0...Y1.</p>		

Системные регистры, задействованные при выполнении инструкции:

- SD141, SD140: регистры текущего выходного значения Y0 (32 бита);
- SD143, SD142: регистры текущего выходного значения Y1 (32 бита);
- SM145: завершение работы импульсного выхода Y0;
- SM146: завершение работы импульсного выхода Y1;
- SM147: состояние импульсного выхода Y0 (Занято/Готово);
- SM148: состояние импульсного выхода Y1 (Занято/Готово);



Когда M0 включен, на выходе Y0 формируются импульсы с частотой 1000 Гц. Когда X3 включится, то частота уменьшится до 100 Гц. Выдача импульсов прекращается, когда X3 выключается.

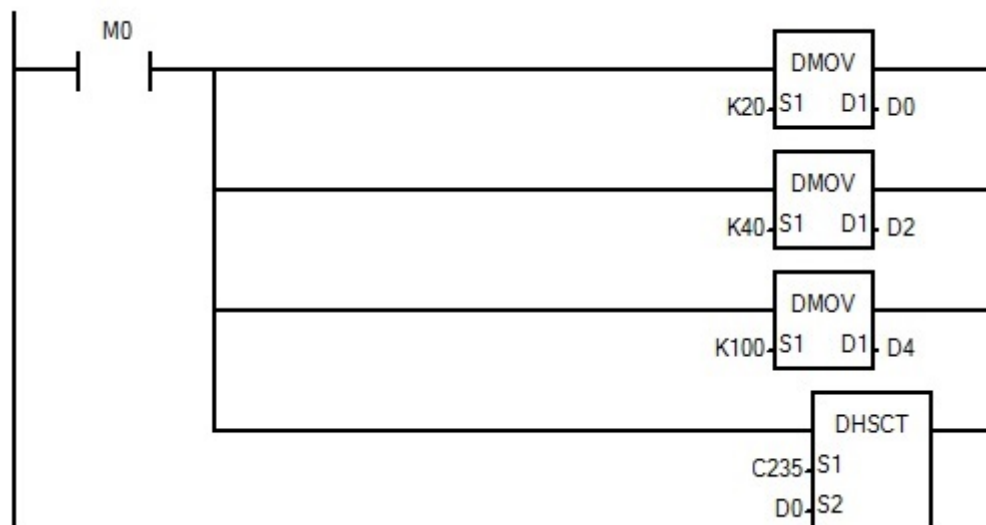
[DHSCT]

Инструкция	DHSCT (32-х битная инструкция)																
Функция	Прерывание по совпадению счетчика																
Доступные входы	S1	C (C235...C255)	Используемый счетчик														
	S2	D	Начальный адрес регистра данных уставок счетчика														
Описание	<p>При выполнении инструкции происходит подсчет поступающих импульсов счетчиком, указанным в S1 и при совпадении количества импульсов с данными в регистрах, начальный адрес которых указан в S2, вызывается подпрограмма прерывания. Всего можно использовать до 16 значений уставки для счетчика. Например, если начальный адрес S2=D0, то первый вызов подпрограммы прерывания произойдет при совпадении значения счетчика со значением в (D1,D0). Второй при совпадении с (D3,D2) и т. д. Если данные в регистре 0 - завершение цикла. Номер прерывания в зависимости от использованного счетчика показан в таблице ниже:</p>																
	<table border="1"> <thead> <tr> <th>Счетчик</th> <th>Номер прерывания</th> <th>Количество прерываний</th> </tr> </thead> <tbody> <tr> <td>C235</td> <td>I35</td> <td>SD235</td> </tr> <tr> <td>C236</td> <td>I36</td> <td>SD236</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>C255</td> <td>I55</td> <td>SD255</td> </tr> </tbody> </table> <p>В программе можно использовать не более 6-ти инструкций DHSCT и для каждой инструкции должны использоваться разные счетчики.</p>			Счетчик	Номер прерывания	Количество прерываний	C235	I35	SD235	C236	I36	SD236	C255	I55
Счетчик	Номер прерывания	Количество прерываний															
C235	I35	SD235															
C236	I36	SD236															
...															
C255	I55	SD255															

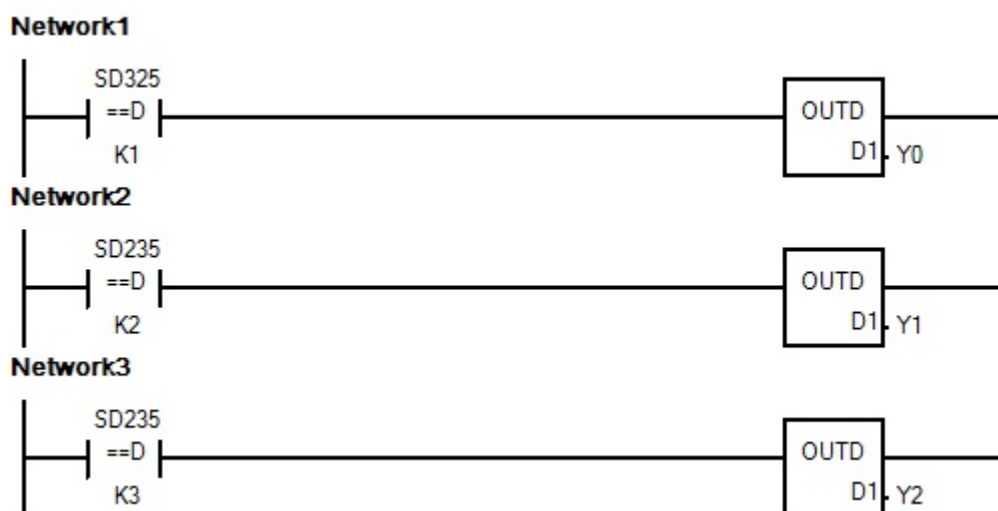
Для применения необходимо:

1. Разместить константы в регистры D0, D2, D4.

2. В инструкции DHSCT указать счетчик C235 и начальный адрес регистров для уставок D0.



3. Создать подпрограмму обработки прерывания I35 (Program Block> Insert INT Program)



При каждом срабатывании прерывания, содержимое регистра SD235 будет увеличиваться на единицу. В подпрограмме обработки прерывания происходит сравнение значения регистра с константой и срабатывает соответствующий выход.

В результате выполнения программы произойдет поочередное включение Y0-Y2 при достижении уставок счетчика C235 – 20(включится Y0), 40(включится Y1), и 100(включится Y2).

[DSZR]

Инструкция	DSZR (16-ти битная инструкция)													
Функция	Возврат в нулевую точку с поиском DOG (сигнала зоны остановки)													
Доступные входы	S1	X, Y, M, SM, S	Вход сигнала начала зоны DOG											
	S2	X	Вход сигнала окончания зоны DOG											
Доступные выходы	D1	Y	Импульсный выход											
	D2	Y	Выход направления движения											
Описание	<p>S1 - Сигнал ближней точки DOG. Логика сигнала определяется битом флага инверсии, как показано в таблице ниже:</p> <table border="1"> <thead> <tr> <th>Входной сигнал</th> <th>Бит инверсии</th> <th rowspan="5">0 - Если входной сигнал ON, то сигнал ближней точки ON. 1- Если входной сигнал OFF, то сигнал ближней точки ON.</th> </tr> </thead> <tbody> <tr> <td>X0</td> <td>SM345</td> </tr> <tr> <td>X1</td> <td>SM355</td> </tr> <tr> <td>X2</td> <td>SM365</td> </tr> <tr> <td>X3</td> <td>SM375</td> </tr> </tbody> </table>			Входной сигнал	Бит инверсии	0 - Если входной сигнал ON, то сигнал ближней точки ON. 1- Если входной сигнал OFF, то сигнал ближней точки ON.	X0	SM345	X1	SM355	X2	SM365	X3	SM375
	Входной сигнал	Бит инверсии	0 - Если входной сигнал ON, то сигнал ближней точки ON. 1- Если входной сигнал OFF, то сигнал ближней точки ON.											
X0	SM345													
X1	SM355													
X2	SM365													
X3	SM375													
<p>S2 - Сигнал окончания зоны DOG. Логика сигнала определяется битом флага инверсии, как показано в таблице ниже:</p> <table border="1"> <thead> <tr> <th>Входной сигнал</th> <th>Бит инверсии</th> <th rowspan="3">0 - Если входной сигнал ON, то сигнал окончания зоны ON. 1- Если входной сигнал OFF, то сигнал окончания зоны</th> </tr> </thead> <tbody> <tr> <td>X0</td> <td>SM346</td> </tr> <tr> <td>X1</td> <td>SM356</td> </tr> </tbody> </table>			Входной сигнал	Бит инверсии	0 - Если входной сигнал ON, то сигнал окончания зоны ON. 1- Если входной сигнал OFF, то сигнал окончания зоны	X0	SM346	X1	SM356					
Входной сигнал	Бит инверсии	0 - Если входной сигнал ON, то сигнал окончания зоны ON. 1- Если входной сигнал OFF, то сигнал окончания зоны												
X0	SM346													
X1	SM356													

X2	SM366	ON
X3	SM376	

D1 - Адрес импульсного выхода, могут использоваться только Y0...Y1.

D2 - Сигнал направления движения. ON - движение вперед, OFF - движение назад.

Направление возврата в нулевую точку определяется значением специальных бит согласно таблице ниже:

Выходной сигнал	Бит направления	0 - Возврат в нулевую точку происходит при движении в обратном направлении. 1- Возврат в нулевую точку происходит при движении в прямом направлении.
Y0	SM342	
Y1	SM352	

Очистка выходного сигнала.

С помощью этой команды сигнал очистки может быть активирован после остановки в нулевой точке. Функция может быть включена путем размыкания бита очистки, как показано в двух таблицах ниже:

A. Функция, обозначенная программным элементом сигнала очистки не используется, т.е. программный элемент установлен по умолчанию:

Выходной сигнал	Бит флага сигнала очистки	Действующая функция, обозначенная программным элементом сигнала очистки	Программный элемент сигнала очистки
Y0	SM341=ON	SM464=OFF	Y4
Y1	SM351=ON	SM465=OFF	Y5

B. Функция сигнала очистки назначается пользователем:

Выходной сигнал	Бит флага сигнала	Действующая функция,	Программный элемент
-----------------	-------------------	----------------------	---------------------

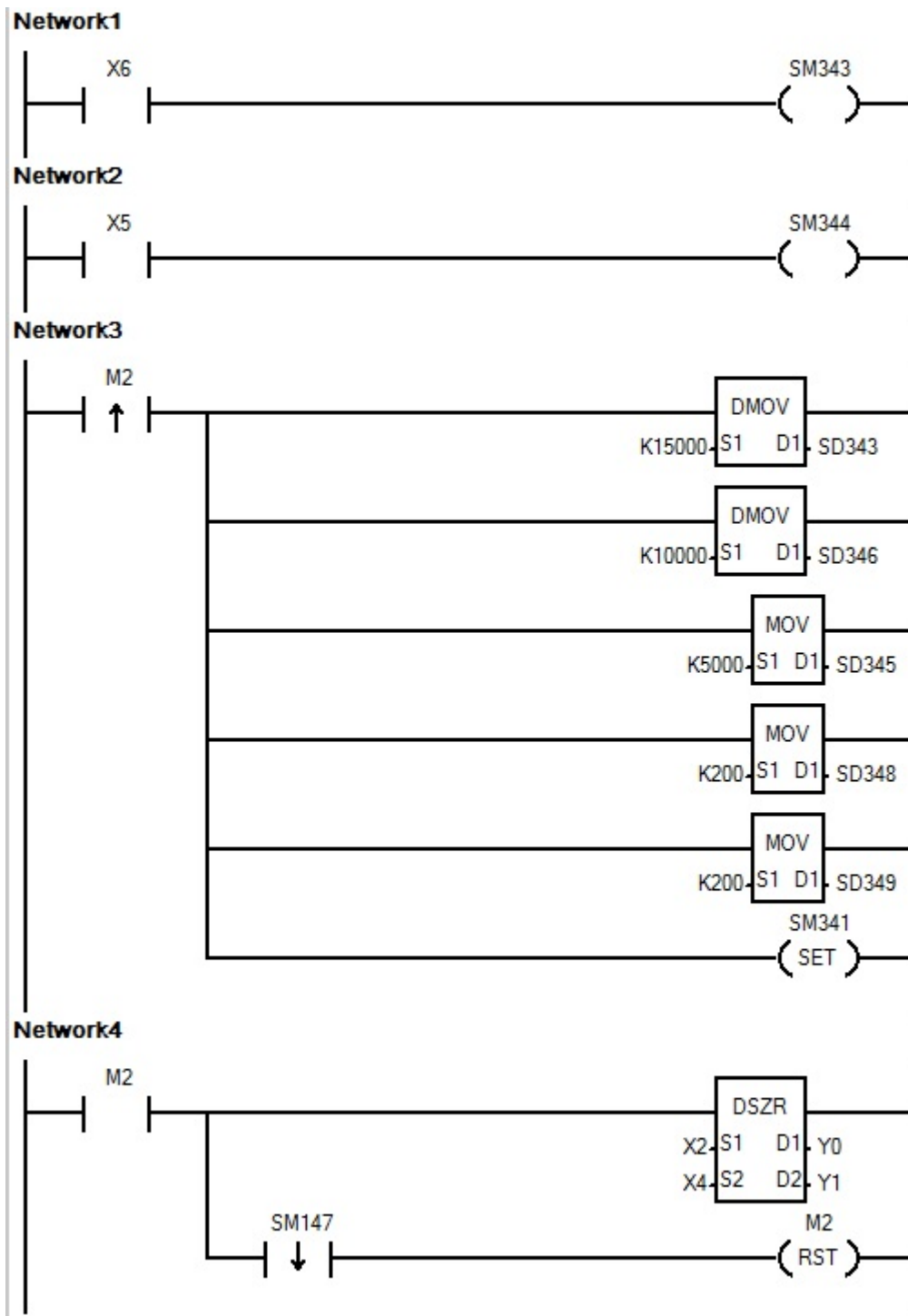
	очистки	обозначенная программным элементом сигнала очистки	сигнала очистки
Y0	SM341=ON	SM464=ON	SD464
Y1	SM351=ON	SM465=ON	SD465

Скорость возврата в нулевую точку (по умолчанию 50 кГц):
При задании скоростей необходимо соблюдать условие:
базовая скорость ≤ скорость возврата ≤ максимальная скорость.

Выходной сигнал	Базовая скорость	Скорость возврата	Максимальная скорость
Y0	SD342	SD346	SD343
Y1	SD352	SD356	SD353

Скорость подхода к целевому положению (по умолчанию 1кГц):

Выходной сигнал	Базовая скорость	Скорость подхода	Максимальная скорость
Y0	SD342	SD345	SD343
Y1	SD352	SD355	SD353



К входу X6 подключен концевой выключатель, ограничивающий движение вперед.

К входу X5 подключен концевой выключатель, ограничивающий движение назад.

К входу X2 подключен концевой выключатель начала зоны DOG

К входу X4 подключен концевой выключатель окончания зоны DOG

При срабатывании M2, происходит задание скоростей:

SD343 - максимальная скорость движения

SD346 - скорость возврата в нулевую точку

SD345 - скорость движения в зоне DOG

SD348 - время ускорения

SD349 - время замедления

При включении M2 на выходе Y0 происходит выдача импульсов с частотой 10 кГц (SD346). При поступлении сигнала начала зоны DOG (X2) частота импульсов на выходе изменяется на 5 кГц (SD345). При поступлении сигнала окончания зоны DOG (X4) генерация импульсов прекращается – возврат в начальную точку завершен. Изначально направление движения – назад (Y1 – выключен). При срабатывании концевого выключателя ограничения движения назад направление движения изменяется на противоположное (Y1- включен). Таким образом происходит поиск сигнала начала зоны остановки (X2). Если после изменения направления движения сработал конечной выключатель движения вперед то генерация импульсов прекращается.

[DRVI / DDRVI]

Инструкция	DRVI (16-ти битная инструкция) DDRVI (32-х битная инструкция)		
Функция	Односкоростное позиционирование в относительных координатах		
Доступные входы			
	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Количество импульсов
	S2		Частота импульсов, Гц
Доступные выходы			
	D1	Y	Импульсный выход
	D2	Y, M, SM, S	Выход направления движения
Описание	<p>При выполнении инструкции происходит генерация выходных импульсов в количестве S1 с частотой S2 на выходе D1. D2 изменяет состояние в зависимости от значения, заданного в S1- при положительном значении (движение вперед) D2=ON, при отрицательном значении D2=OFF;</p> <p>Диапазоны значений: S1- для 16-ти битной инструкции - 32768...32 767. Для 32-х битной инструкции -2147483648...2147483647; S2- для 16-ти битной инструкции 10...32767 Гц. Для 32-х битной инструкции 0...100000 Гц; В качестве импульсного выхода в D1 могут использоваться только Y0 или Y1; Текущее количество выданных импульсов находится : Для выхода Y0 – в регистрах SD141(старший байт), SD140(младший байт) при использовании 32-х разрядной инструкции; Для выхода Y1 – в регистрах SD143(старший байт), SD142(младший байт) при использовании 32-х разрядной инструкции; При изменении значения S1 и S2 во время выполнения инструкции не приведет к немедленному изменению параметров. Новые значения будут использованы только после завершения работы инструкции; Если условия выполнения инструкции будет снято в процессе работы, то произойдет выдача импульсов с замедлением и остановка, при этом флаг завершения</p>		

выполнения инструкции SM29 не сработает;
Если заданная частота S2 меньше базовой частоты SD145(по умолчанию 5 кГц) то выдача импульсов будет производится с частотой SD145.
Время ускорения и замедления задается в регистре SD148(по умолчанию 100 мс).

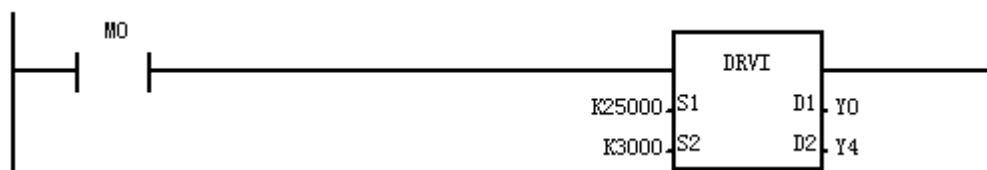
Флаги:

SM147 – импульсный выход Y0 занят;

SM148 – импульсный выход Y1 занят;

SM149 – импульсный выход Y2 занят;

SM150 – импульсный выход Y3 занят;



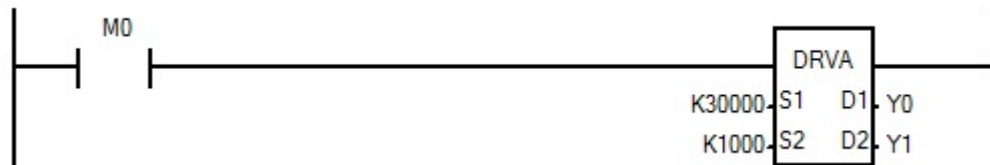
Когда M0 включен, на выходе Y0 генерируется 25000 импульсов с частотой 3000 Гц. При этом Y4 будет включен, т. к. значение S1 положительное число (движение вперед).

[DRVA / DDRVA]

Инструкция	DRVA (16-ти битная инструкция) DDRVA (32-х битная инструкция)		
Функция	Однокоростное позиционирование в абсолютных координатах		
Доступные входы			
	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z	Количество импульсов
	S2		Частота импульсов, Гц
Доступные выходы			
	D1	Y	Импульсный выход
	D2	Y, M, SM, S	Выход направления движения
Описание	<p>При выполнении инструкции происходит генерация выходных импульсов в количестве S1 с частотой S2 на выходе D1. D2 изменяет состояние в зависимости от значения, заданного в S1- при положительном значении (движение вперед) D2=ON, при отрицательном значении D2=OFF;</p> <p>Диапазоны значений: S1- для 16-ти битной инструкции - 32768...32 767. Для 32-х битной инструкции -2147483648...2147483647; S2- для 16-ти битной инструкции 10...32767 Гц. Для 32-х битной инструкции 0...100000 Гц; В качестве импульсного выхода в D1 могут использоваться только Y0 или Y1; Текущее количество выданных импульсов находится : Для выхода Y0 – в регистрах SD141(старший байт), SD140(младший байт) при использовании 32-х разрядной инструкции; Для выхода Y1 – в регистрах SD143(старший байт), SD142(младший байт) при использовании 32-х разрядной инструкции; При изменении значения S1 и S2 во время выполнения инструкции не приведет к немедленному изменению параметров. Новые значения будут использованы только после завершения работы инструкции; Если условия выполнения инструкции будет снято в процессе работы, то произойдет выдача импульсов с замедлением и остановка, при этом флаг завершения</p>		

выполнения инструкции SM29 не сработает;
Если заданная частота S2 меньше базовой частоты SD145(по умолчанию 5 кГц) то выдача импульсов будет производится с частотой SD145.
Время ускорения и замедления задается в регистре SD148(по умолчанию 100 мс).

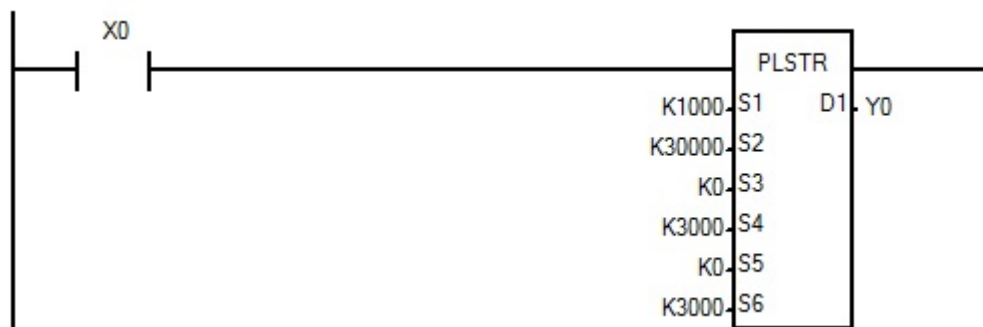
Флаги:
SM147 – импульсный выход Y0 занят;
SM148 – импульсный выход Y1 занят;



Когда M0 включен, идет процесс абсолютного позиционирования – в качестве заданной позиции выступает значение регистра SD140-SD141 – количество выданных импульсов на выходе Y0. Например, если текущее значение SD140-141 равно 80000, то при выполнении инструкции на выход Y0 будут подаваться импульсы с частотой 1 кГц до тех пор, пока значение SD140-SD141 не станет равным 30000. При этом выход Y1 будет выключен, так как идет движение назад (изначальная абсолютная позиция больше чем заданная).

[PLSTR / DPLSTR]

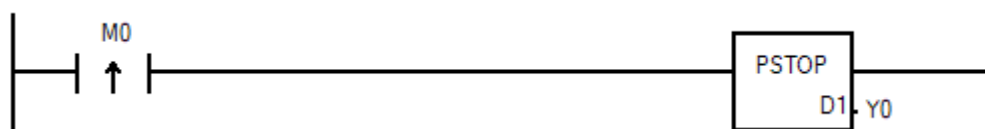
Инструкция	PLSTR (16-ти битная инструкция) DPLSTR (32-х битная инструкция)		
Функция	Выдача импульсов с ускорением и замедлением (трапецевидный график движения)		
Доступные входы	S1	K, KnX, KnY, KnM, KnSM, KnS, T, C, D, SD, V, Z, AI, AQ	Основная частота, Гц
	S2		Количество выходных импульсов
	S3		Начальная частота, Гц
	S4		Время ускорения, мс
	S5		Конечная частота, Гц
	S6		Время замедления, мс
Доступные выходы	D1	Y	Импульсный выход
Описание	<p>S1 - Основная частота выходных импульсов. Диапазон для 16-ти битной инструкции -32768...32767. Для 32-х битной инструкции -2147483648...2147483647;</p> <p>S2 - Количество выходных импульсов. Диапазон для 16-ти битной инструкции -32768...32767. Для 32-х битной инструкции -2147483648...2147483647;</p> <p>S3 - Начальная частота выходных импульсов. Диапазон для 16-ти битной инструкции -32768...32767. Для 32-х битной инструкции -2147483648...2147483647;</p> <p>S4 - Время ускорения на начальном участке. Диапазон 50...5000 миллисекунд;</p> <p>S5- Конечная частота выходных импульсов. Диапазон для 16-ти битной инструкции -32768...32767. Для 32-х битной инструкции -2147483648...2147483647;</p> <p>S6 - Время замедления на конечном участке. Диапазон 50...5000 миллисекунд;</p> <p>D1 - Импульсный выход;</p> <p>Инструкция должна использоваться только для ПЛК с транзисторными выходами;</p> <p>В качестве импульсного выхода в D1 могут использоваться только Y0 или Y1;</p>		



Когда X0 замкнут, происходит плавное ускорение от начальной частоты S3 (0 Гц) до основной частоты S1 (1000 Гц) за время S4 (3 секунды). После работы на основной частоте происходит снижение до конечной частоты S5 (0 Гц) за время S6 (3 секунды). Всего за время работы инструкции выдается количество импульсов заданным в S2.

[PSTOP]

Инструкция	PSTOP		
Функция	Остановка работы импульсного выхода		
Доступные выходы			
	D1	Y	Импульсный выход
Описание	При выполнении инструкции импульсный выход немедленно останавливается.		



При включении M0 выполнение импульсной инструкции на выходе Y0 останавливается.

4.12 Инструкции связи с внешними устройствами (External Device Inst)

Данный раздел содержит инструкции по работе с внешними устройствами по протоколу Modbus RTU, а также инструкции ПИД-регулирования.

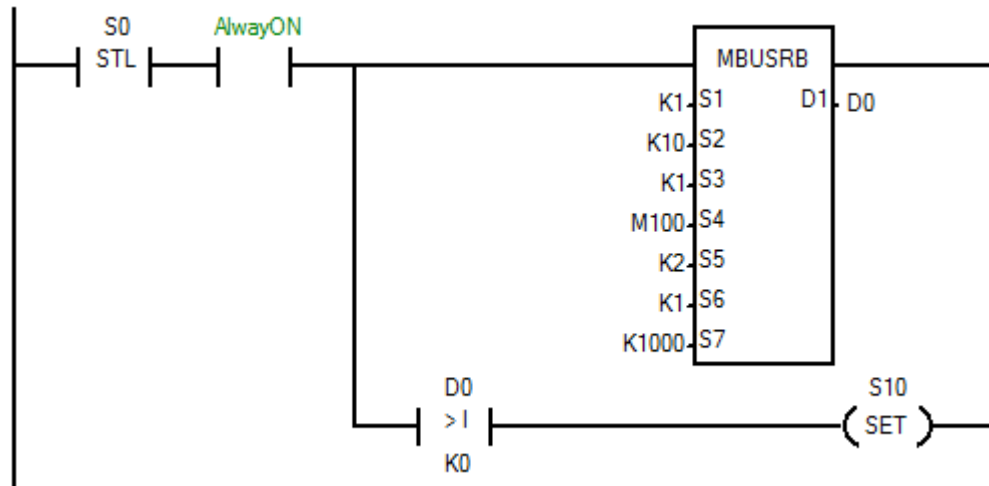
Для добавления нескольких инструкций чтения / записи Modbus в тело программы требуется использование шаговых инструкций(STL).

- **MBUSRB** – Чтение битовых регистров
- **MBUSWB** – Запись битовых регистров
- **MBUSRW** – Чтение регистров типа WORD
- **MBUSWW** – Запись регистров типа WORD
- **PIDT** – Автонастройка ПИД-регулятора
- **PIDR** – ПИД-регулятор

[MBUSRB]

Инструкция	MBUSRB		
Функция	Чтение битовых регистров по Modbus		
Доступные входы	S1	D, K: 1..2	Номер порта (COM1/COM2)
	S2	D, K: 1..255	Адрес slave-устройства
	S3	D, K	Инструкция чтения: 1- чтение DO; 2- чтение DI.
	S4	Y, M	Регистр для сохранения состояния считанного бита
	S5	D, K: 0..65535	Адрес считываемого регистра (логический). При использовании физического адреса указать значение на 1 больше.
	S6	D, K: 1..65535	Количество считываемых регистров
	S7	K: 1..65535	Максимальное время ожидания ответа, мс
	Доступные выходы		
D1		D	Код состояния: 0 – не выполняется / в процессе выполнения; 1 – команда выполнена успешно; 2 – превышение времени ожидания ответа; 3 – неверный адрес Slave-устройства; 4 – ошибка кода функции; 5 – ошибка тестирования (testing fault); 101 – принятый код функции не может быть обработан; 102 – адрес данных, указанный в запросе, недоступен.

Описание	Производится чтение регистра S5 и его запись в битовый регистр S4 с Modbus-устройства с адресом S2 по порту S1 функцией S3. На выходе D1 формируется код состояния.



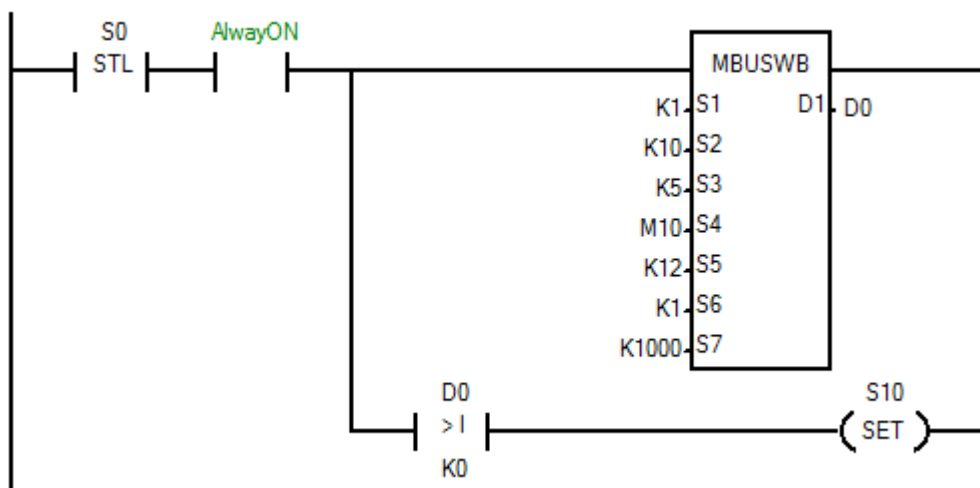
На шаге S0 происходит чтение командой 1 (S3=K1) одного бита (S6=K1) из Modbus-устройства с адресом 10 (S2=K10) из регистра с физическим адресом 1 (логический адрес 000002, S5=K2). Состояние прочитанного бита записывается в регистр M100.

При любом исходе завершения чтения (удачном или нет), программа переходит на следующий шаг.

[MBUSWB]

Инструкция	MBUSWB		
Функция	Запись битовых регистров		
Доступные входы	S1	D, K: 1..2	Номер порта (COM1/COM2)
	S2	D, K: 1..255	Адрес slave-устройства
	S3	D, K	Инструкция записи: 5 - запись одного регистра; 15 - запись нескольких регистров.
	S4	X, Y, M	Внутренний битовый регистр, состояние которого необходимо записать в slave-устройство
	S5	D, K: 0..65535	Адрес записываемого регистра (логический). При использовании физического адреса указать значение на 1 больше.
	S6	D, K: 1..65535	Количество записываемых регистров
	S7	K: 1..65535	Максимальное время ожидания ответа, мс
	Доступные выходы	D1	D

		102 – адрес данных, указанный в запросе, недоступен.
Описание	Производится запись значения битового регистра S4 в регистр S5 Modbus-устройства с адресом S2 по порту S1 функцией S3. На выходе D1 формируется код состояния.	



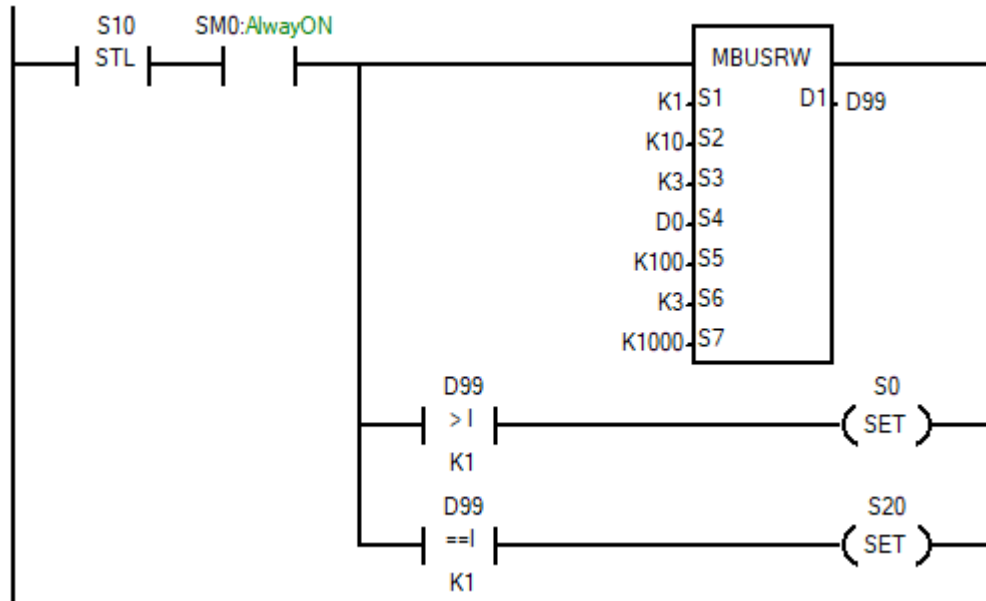
На шаге S0 происходит запись командой 5 (S3=K5) одного бита (S6=K1) в Modbus-устройства с адресом 10 (S2=K10) в регистр с физическим адресом 11 (логический адрес 000012, S5=K12). Состояние прочитанного бита записывается в регистр M100.

При любом исходе завершения записи (удачном или нет), программа переходит на следующий шаг.

[MBUSRW]

Инструкция	MBUSRW		
Функция	Чтение 16-битных регистров		
Доступные входы	S1	D, K: 1..2	Номер порта (COM1/COM2)
	S2	D, K: 1..255	Адрес slave-устройства
	S3	D, K	Инструкция чтения: 3 - чтение Holding Register; 4 - чтение Input Register.
	S4	D	Регистр для сохранения считанного значения
	S5	D, K: 0..65535	Адрес считываемого регистра (логический). При использовании физического адреса указать значение на 1 больше.
	S6	D, K: 1..65535	Количество считываемых регистров
	S7	K: 1..65535	Максимальное время ожидания ответа, мс
	Доступные выходы		
D1		D	Код состояния: 0 – не выполняется / в процессе выполнения; 1 – команда выполнена успешно; 2 – превышение времени ожидания ответа; 3 – неверный адрес Slave-устройства; 4 – ошибка кода функции; 5 – ошибка тестирования (testing fault); 101 – принятый код функции не может быть обработан; 102 – адрес данных, указанный в запросе, недоступен.

Описание	Производится чтение регистра S5 и его запись в регистр S4 с Modbus-устройства с адресом S2 по порту S1 функцией S3. На выходе D1 формируется код состояния.



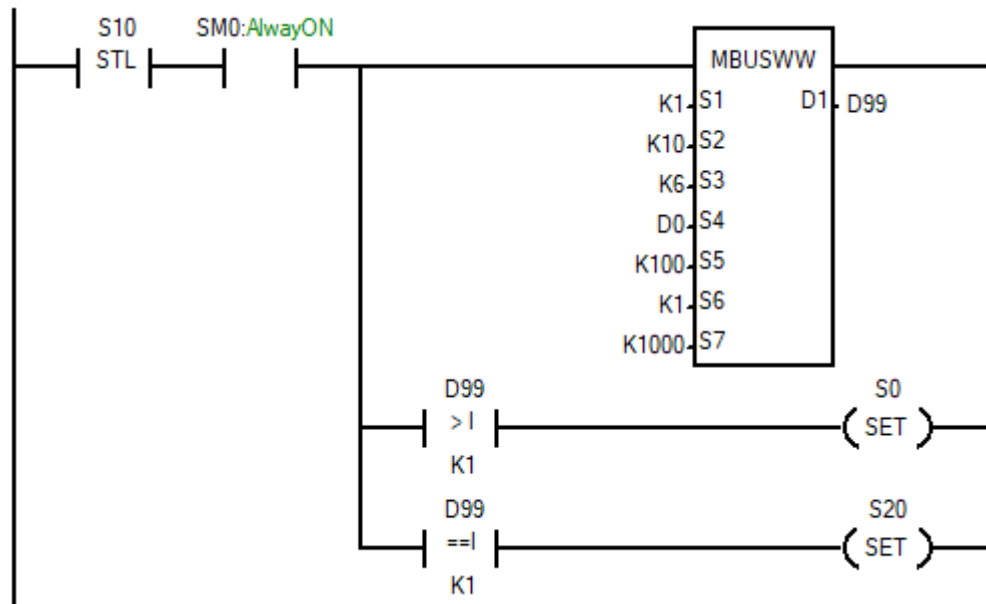
На шаге S10 происходит чтение командой 3 (S3=K3) трёх регистров (S6=K3) из Modbus-устройства с адресом 10 (S2=K10) из регистра с физическим адресом 99 (логический адрес 400100, S5=K100). Прочитанное значение записывается в регистр D0.

В случае возникновения ошибки при чтении, программа возвращается на предыдущий шаг S0. При успешном завершении чтения (D99 = 1), программа переходит на следующий шаг S20.

[MBUSWW]

Инструкция	MBUSWW		
Функция	Запись 16-битных регистров		
Доступные входы	S1	D, K: 1..2	Номер порта (COM1/COM2)
	S2	D, K: 1..255	Адрес slave-устройства
	S3	D, K	Инструкция записи: 6 - запись одного регистра; 16 - запись нескольких регистров.
	S4	D	Внутренний регистр, значение которого необходимо записать в slave-устройство
	S5	D, K: 0..65535	Адрес записываемого регистра (логический). При использовании физического адреса указать значение на 1 больше.
	S6	D, K: 1..65535	Количество записываемых регистров
	S7	K: 1..65535	Максимальное время ожидания ответа, мс
	Доступные выходы	D1	D

		102 – адрес данных, указанный в запросе, недоступен.
Описание	Производится запись значения регистра S4 в регистр S5 Modbus-устройства с адресом S2 по порту S1 функцией S3. На выходе D1 формируется код состояния.	



На шаге S10 происходит запись командой 6 (S3=K6) одного регистра (S6=K1) в Modbus-устройства с адресом 10 (S2=K10) в регистр с физическим адресом 99 (логический адрес 400100, S5=K100). Записываемое значение берётся из регистра D0.

В случае возникновения ошибки при записи, программа возвращается на предыдущий шаг S0. При успешном завершении записи (D99 = 1), программа переходит на следующий шаг S20.

[PIDT]

Инструкция	PIDT		
Функция	Автонастройка ПИД-Регулятора		
Доступные входы	S1	K, T, C, D, SD, V, Z, AI	Текущее значение регулируемого параметра
	S2		Уставка регулятора
	S3		Выбор режима автонастройки
	S4		Временной интервал между расчетами ПИД-регулятора (x10 мс)
Доступные выходы	D1	T, C, D, SD, V, Z, AQ	Выходная мощность ПИД-регулятора
	D2		Коэффициент пропорциональности
	D3		Время интегрирования (x100 мс)
	D4		Время дифференцирования (x100 мс)
	D5		Зона накопления интегральной составляющей
	D6		Код состояния. Когда D6=10 автонастройка успешно завершена.
Описание	<p>При выполнении инструкции происходит автонастройка ПИД-регулятора. Выбор режима автонастройки (S3) задается в шестнадцатеричном формате. Значение формируется следующим образом:</p>		

4 бит

4 бит

4 бит

4 бит

D	C	B	A
---	---	---	---

где:

A – выбор выходной мощности выдаваемой в процессе настройки ПИД-регулятора, диапазон от 0 до A – 0-100%;

B – выбор способа автонастройки ПИД-регулятора:

0 - Настройка по колебаниям,

1 - Настройка по переходной характеристике;

C – выбор ограничений при автонастройке:

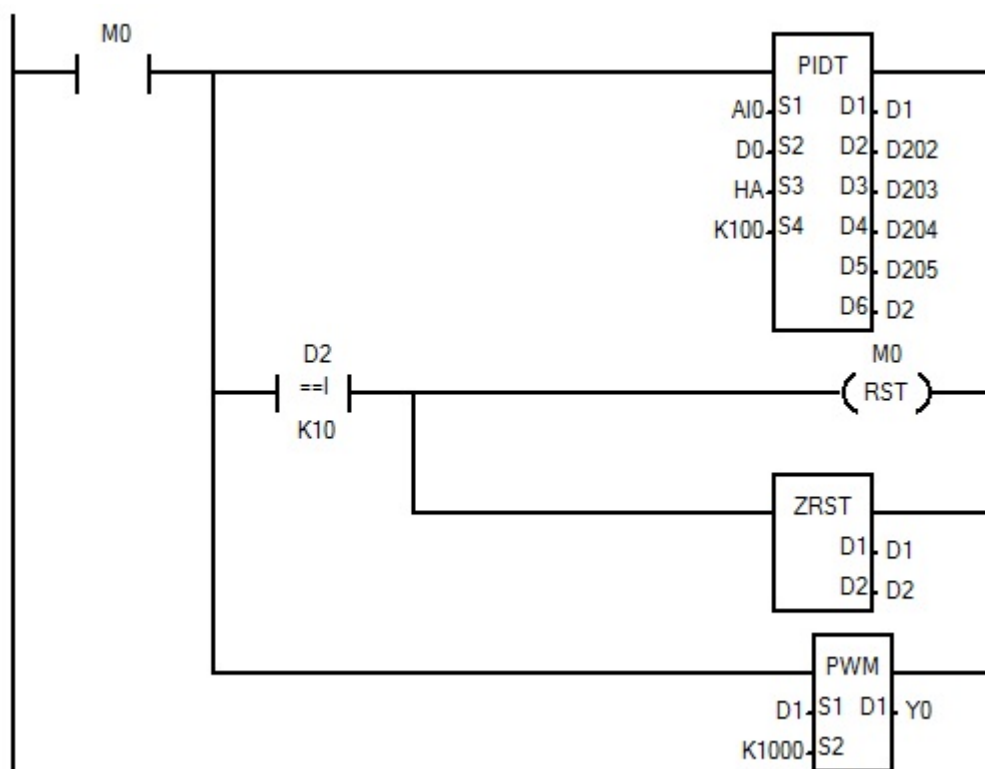
0 - Без ограничений;

1 - При автонастройке допустимо небольшое перерегулирование;

2 - При автонастройке перерегулирование недопустимо.

D – не используется.

Например, для задания режима автонастройки по переходной характеристике со 100% выходной мощностью в S3 необходимо записать значение 1A (hex).



При включении M0 запускается процесс автонастройки. После завершения автонастройки M0 сбросится, а в регистры D202-D205 сохранятся рассчитанные коэффициенты ПИД-регулятора:

D202 - Коэффициент пропорциональности

D203 - Время интегрирования

D204 - Время дифференцирования

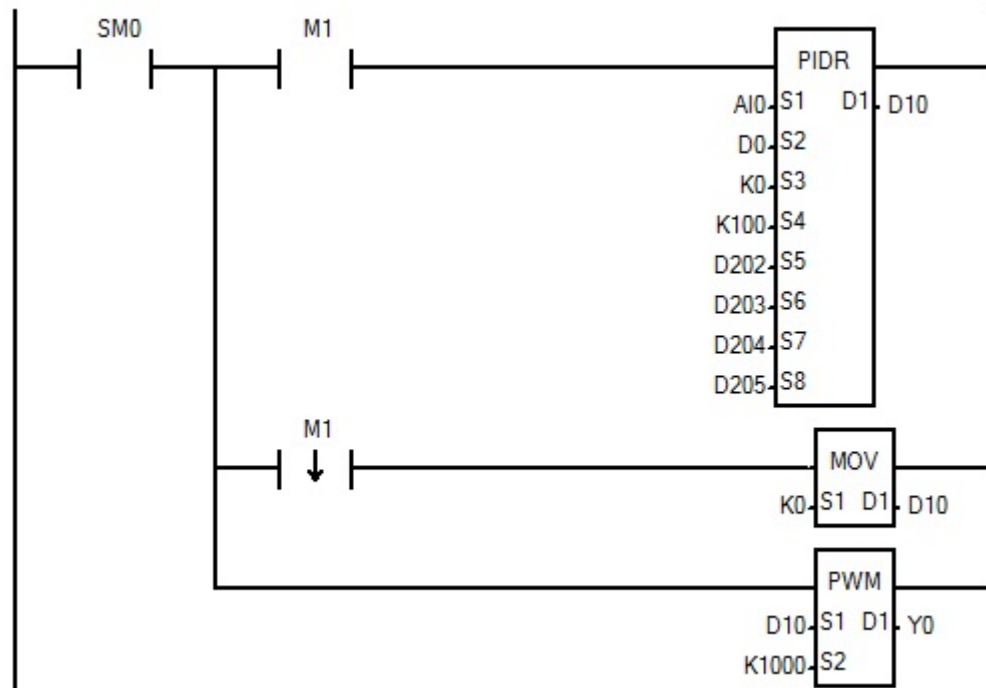
D205 - Зона накопления интегральной составляющей

Y0 - дискретный выход ПЛК.

Если в качестве выхода регулятора используется дискретный выход, то для преобразования сигнала выходной мощности, рассчитанную ПИД-регулятором необходимо использовать инструкцию PWM-регулятора.

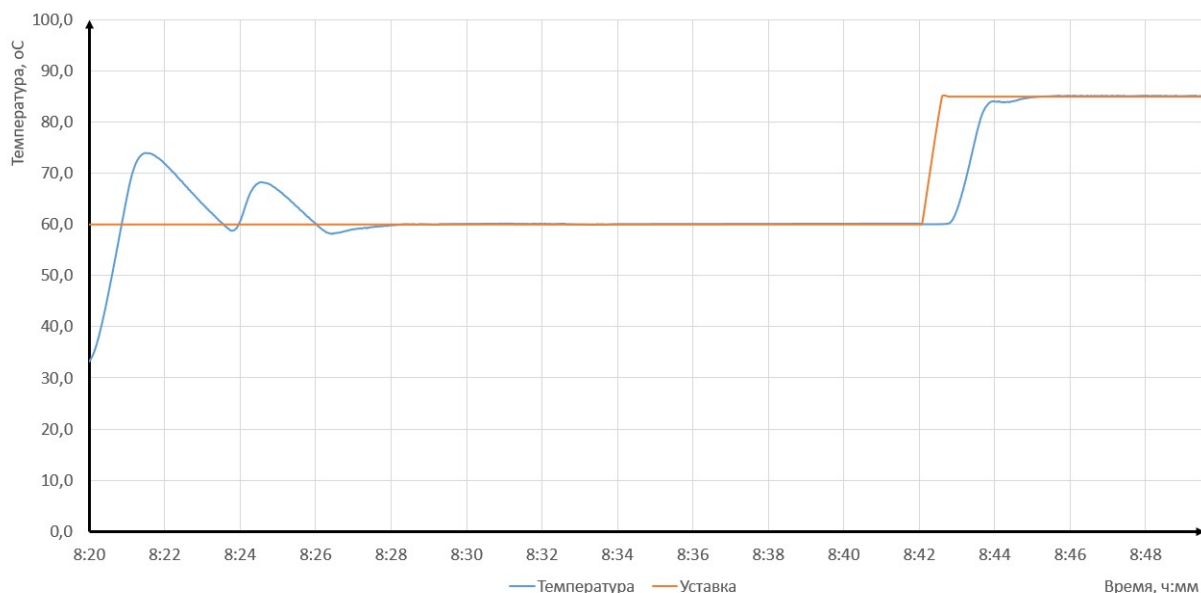
[PIDR]

Инструкция	PIDR		
Функция	ПИД-Регулятор		
Доступные входы	S1	K, T, C, D, SD, V, Z, AI	Текущее значение регулируемого параметра
	S2		Уставка регулятора
	S3		Константа, должна быть равна 0
	S4		Временной интервал между расчетами ПИД-регулятора (x10 мс)
	S5		Коэффициент пропорциональности
	S6		Время интегрирования (x100 мс)
	S7		Время дифференцирования (x100 мс)
	S8		Зона накопления интегральной составляющей
Доступные выходы	D1	T, C, D, SD, V, Z, AQ	Выходная мощность ПИД-регулятора
Описание	Инструкция ПИД-регулятора		



При включении M1 начинает выполняться инструкция ПИД-регулирования. Текущее значение регулируемого параметра поступает со входа A10, уставка регулятора задается в D0. D202-D205 – коэффициенты ПИД-регулятора, Y0 – управляющий выход регулятора.

На рисунке ниже приведен график автонастройки по колебаниям и последующего регулирования со сменой уставки с 60 до 85 С:



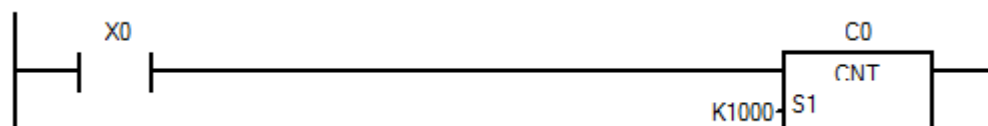
5 ТИПОВЫЕ ЗАДАЧИ

5.1 Подключение кнопки и дребезг контактов

Подключение кнопки является одной из самых базовых и простых задач при работе с ПЛК. Подключение кнопки осуществляется в соответствии со схемой, представленной в паспорте на устройство.

Для работы с кнопками в ПЛК LogicOn используются нормально-открытые, нормально-закрытые контакты, а также детекторы фронтов и некоторые инструкции.

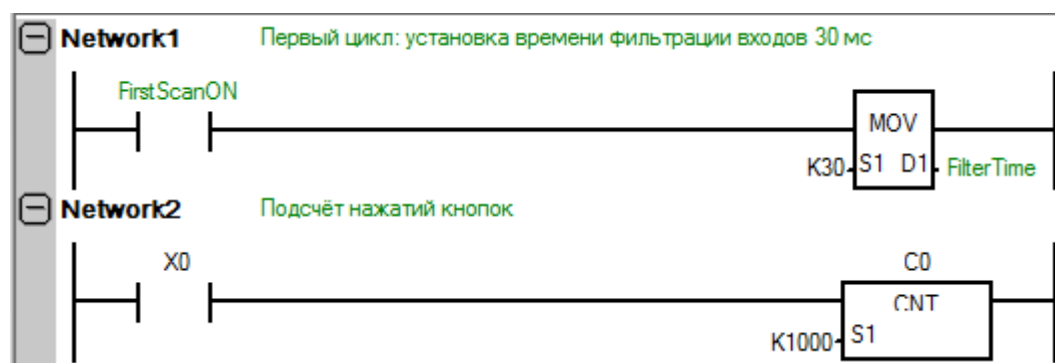
Ниже представлен пример программы, в которой происходит подсчет количества нажатий кнопки, подключенной к входу X0.



Подсчет количества нажатий кнопки

Для исключения влияния дребезга контактов в ПЛК предусмотрена программная фильтрация входов. По умолчанию её величина равна 10 мс, что является достаточным для большинства кнопок. Дополнительно, данное значение можно изменить, записав требуемое время фильтрации в системный регистр SD20.

В следующей программе осуществляется установка времени фильтрации входов 30 мс и дальнейший подсчет нажатий кнопки, подключенной к входу X0. Установка значения фильтрации происходит только в первом цикле контроллера с помощью системного регистра FirstScanOn (SM2). Описание всех системных регистров приведено в разделе «Программные элементы ПЛК».



Установка времени фильтрации входов и подсчет количества нажатий кнопки

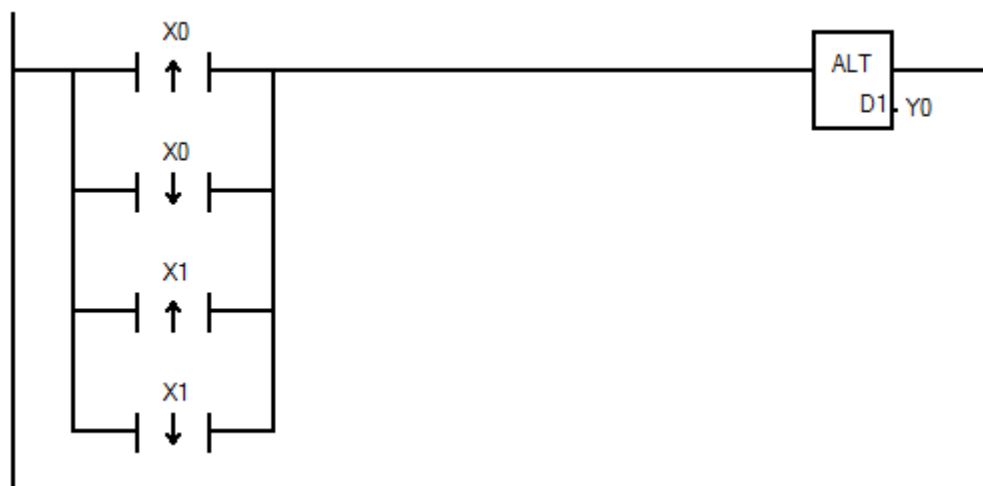
При использовании высокоскоростных инструкций с входами X0...X5, дополнительно не требуется уменьшать время фильтрации входов — это происходит автоматически.

5.2 Управление освещением. Функционал шагового (импульсного) реле

При использовании ПЛК в задаче управления освещением, довольно часто приходится реализовывать функционал шагового реле программно.

Задача: к входам контроллера X0 и X1 подключены два выключателя, выходу Y0 управляет освещением. При изменении состояния любого из выключателей X0 и X1, должно происходить включение или выключение освещения.

Один из самых простых способов решения данной задачи приведен ниже.



Изменение состояние выхода при изменении состояния любой кнопки

Инструкция ALT изменяет состояние выхода Y0 на противоположное, при поступлении переднего или заднего фронта импульса от любого контакта X0 или X1.

При большем количестве выключателей (3 и более) достаточно добавить по 2 сигнала каждого выключателя (детекторы переднего и заднего фронтов) параллельно входным контактам.

5.3 Подключение модулей расширения

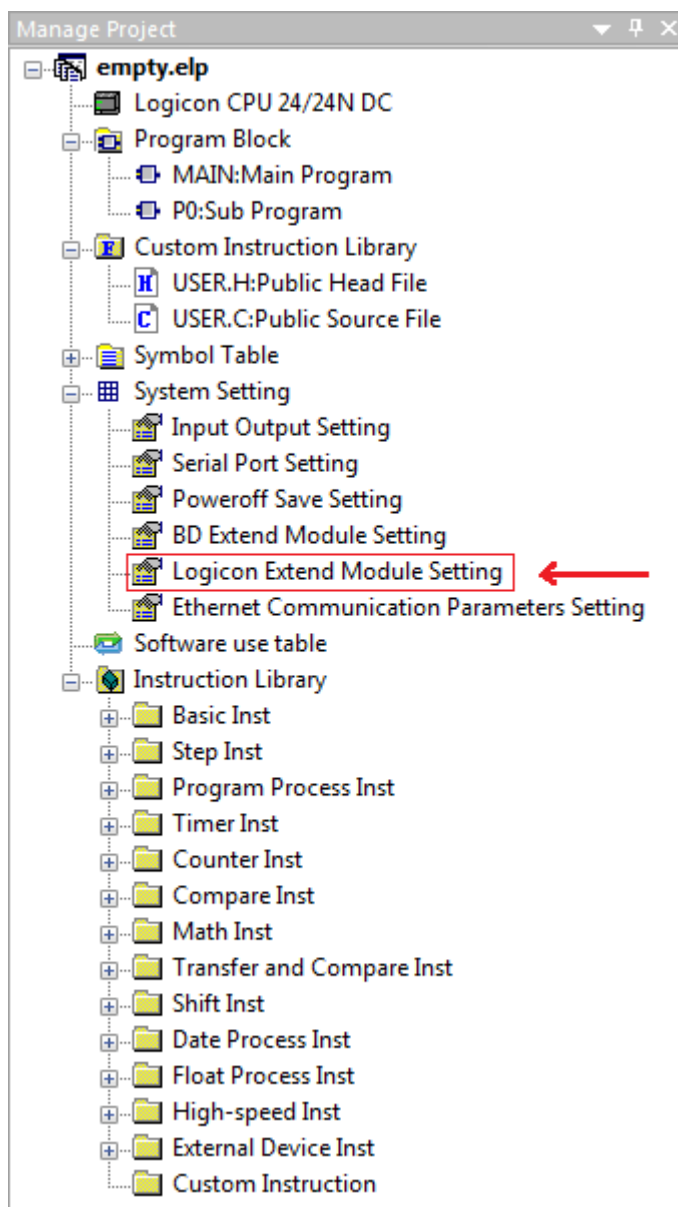
Подключение и отключение модулей расширения обязательно производится при отключенном питании контроллера, а также при отключенном кабеле USB. Контроллер проводит инициализацию модулей расширения после подачи питания, поэтому подавать питание необходимо только после того, как модули будут подключены.

В случае, если кол-во, тип и последовательность подключенных модулей не соответствует программе, будет включен светодиод ошибки «Err.». При обновлении конфигурации модулей может понадобиться обновить ядро ПЛК (Force to Update PLC System).

Модули подключаются к ПЛК по правосторонней шине. Порядок подключения описан в паспорте на модули расширения. Модули питаются от ПЛК по внутренней шине и не требуют дополнительного внешнего питания.

После физического подключения модулей расширения к ПЛК необходимо добавить модули в программу.

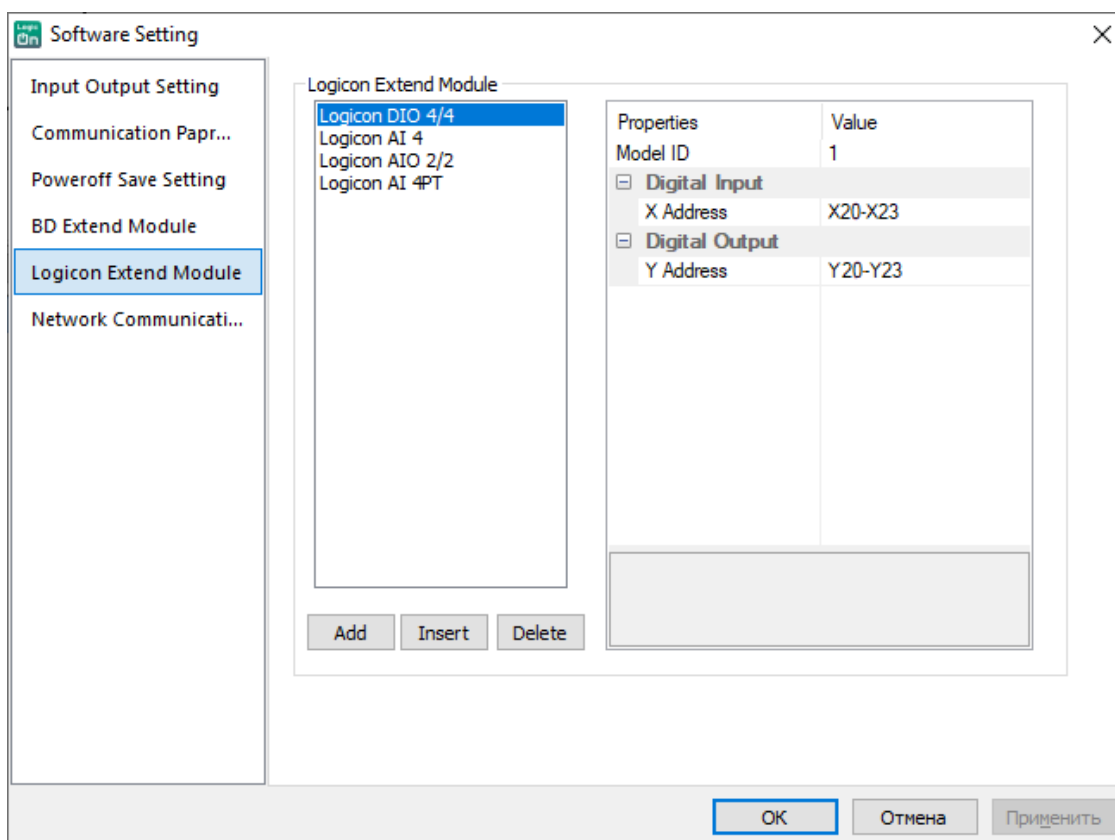
Для добавления необходимо перейти на вкладку «LogicOn Extend Module Setting» в дереве проекта.



Вкладка управления модулями расширения

После перехода на вкладке добавляются требуемые модули расширения. Кнопка «Add» добавляет модуль в конце списка, кнопка «Insert» добавляет модуль перед выбранным.

При добавлении модулей адресация их входов и выходов указывается в окне рядом. После добавления модулей соответствующие индексы можно использовать в программе.



Аналоговые сигналы 4...20мА и 0...10В представляются в виде целых значений в мкА и мкВ соответственно.

Т.е. значение 4000 получаемое от модуля расширения в токовом режиме соответствует физическому значению 4000 мкА или 4.0 мА.

Аналоговые сигналы от датчиков температуры представляются в виде целых значений, при этом самый младший разряд соответствует десятым градуса.

Т.е. значение 1000 получаемое от модуля расширения с температурным входом соответствует физическому значению 100.0 °С.

5.4 Преобразование аналогового сигнала в пользовательский вид

При работе с аналоговыми сигналами зачастую возникает необходимость преобразования сигнала в пользовательский вид. Классический пример: показания от датчика давления с диапазоном 0...10 бар представлены токовым сигналом 4...20 мА.

Для преобразования сигнала 4...20 мА в диапазон 0...10 бар необходимо произвести несколько математических операций. Однако проще воспользоваться специальной инструкцией SCALE, которая выполняет все необходимые вычисления. Ниже представлен пример использования данной инструкции.



Преобразование сигнала от датчика с выходом 4...20мА в диапазон 0...1000

На вход S1 указывается источник сигнала.

Диапазон исходного сигнала определяют входы S2 и S3.

Диапазон выходного сигнала определяют входы S4 и S5.

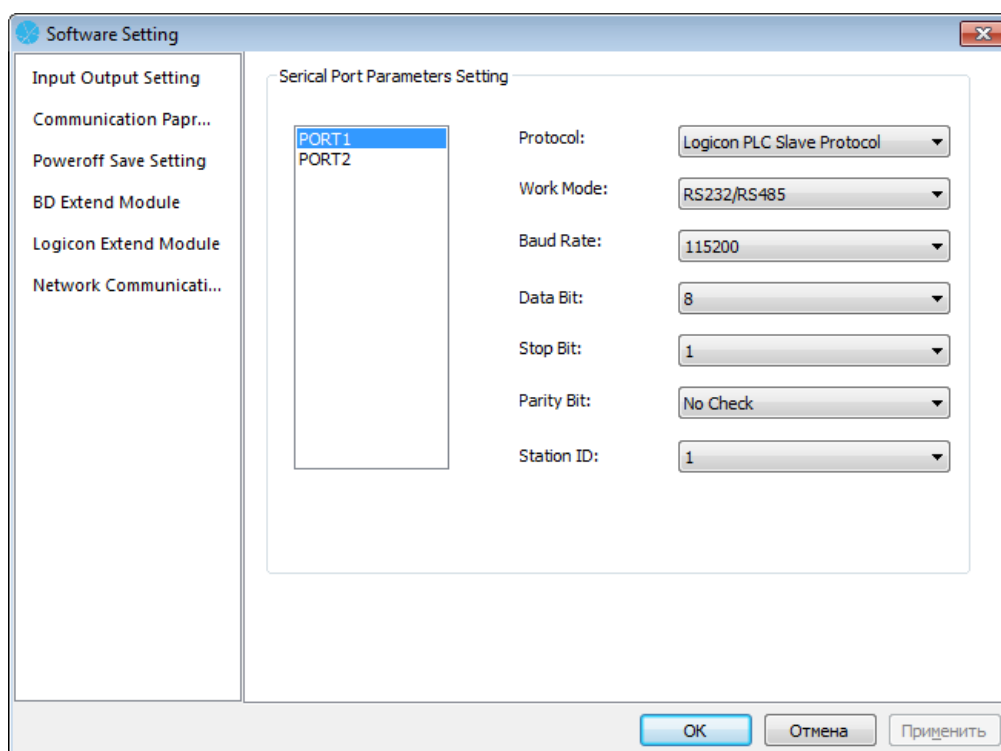
Выход D1 содержит значение после преобразования.

Более подробное описание данной инструкции приведено в разделе «Прикладные инструкции».

5.5 ПЛК в режиме Slave-устройства Modbus

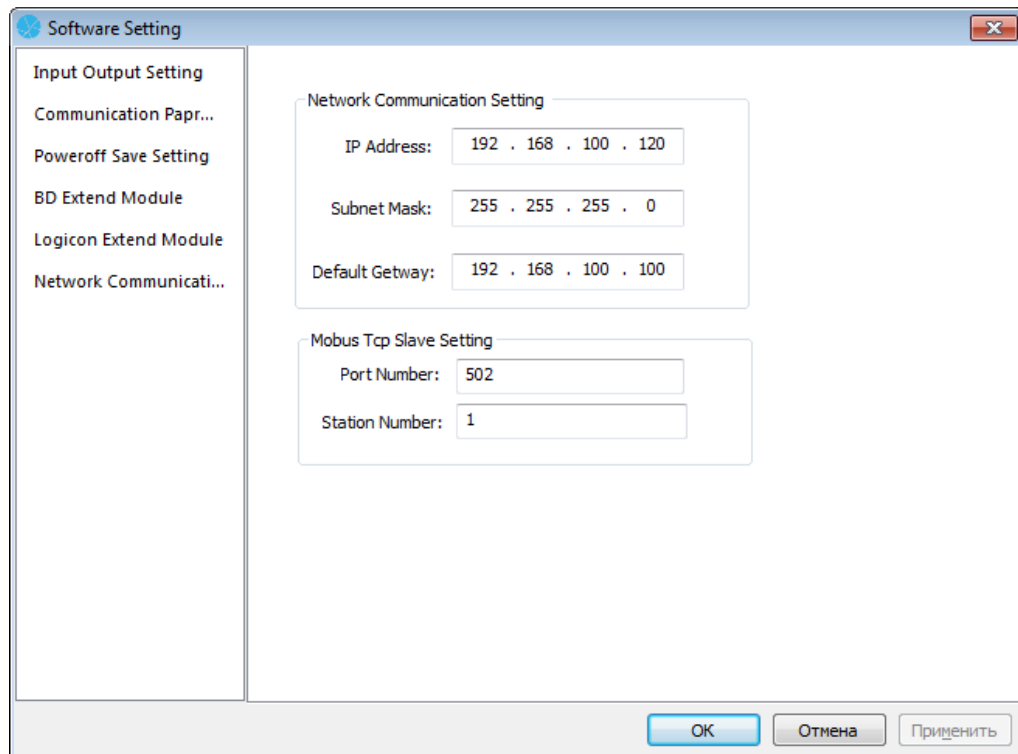
ПЛК может работать как Slave-устройство по всем доступным портам: COM1, COM2, Ethernet. Настройки Slave-соединения для последовательных портов доступны во вкладке «Serial Port Settings». В данной вкладке возможно выбрать параметры сети для каждого порта: скорость работы, контроль четности, стоп-биты, адрес устройства и т. п.

Для работы в режиме Slave обязательно должен быть выбран протокол «Logicon PLC Slave Protocol».



Окно конфигурации последовательных портов

Настройки slave-соединения для Ethernet доступны во вкладке «Ethernet Communication Parameters Setting».



Окно конфигурации последовательных портов

Slave-регистры являются общими для всех интерфейсов. Таблица доступных регистров и функций приведена ниже.

Программные элементы	Тип	Диапазон	Номер регистра DEC	Код функции
Y (выходы)	1-бит	Y0 - Y377	0001-0256	1, 5, 15
X (входы)		X0 - X377	1201-1456	
M (внутренние реле)		M0 - M2047	2001-4048	
SM (системные реле)		SM0 - SM511	4401-4912	
S (шаговое управление)		S0 - S999	6001-7000	
T (таймеры: состояние)		T0 - T255	8001-8256	
C (счётчики: состояние)		C0 - C255	9201-9456	
D (пользовательские регистры)	16-бит	D0 - D4095	0001-4096	3, 6, 16
SD (системные регистры)		SD0 - SD511	8001-8512	
T (таймеры: значение)		T0 - T255	9001-9256	
C (счётчики: значение)		C0 - C199	9501-9700	
C (счётчики: значение)	32-бит	C200 - C255	9701-9756	

5.6 ПЛК в режиме Master-устройства Modbus

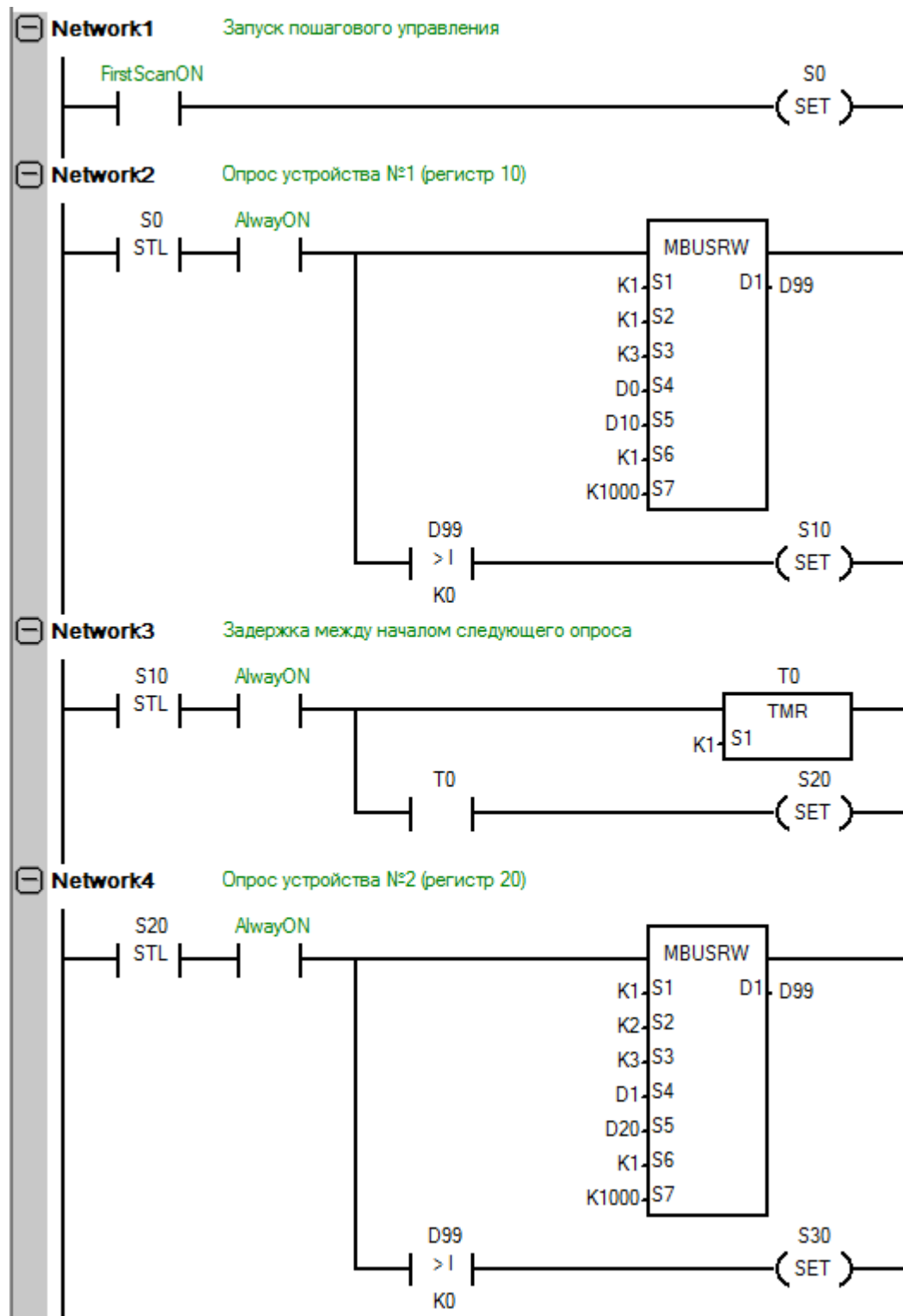
Для работы ПЛК в качестве Master-устройства необходимо установить соответствующую конфигурацию во вкладке «Serial Port Settings». В качестве протокола необходимо выбрать «Logicon PLC Master Protocol».

Для чтения регистров сторонних slave-устройств используются инструкции MBUSRB и MBUSRW. Для записи параметров используются инструкции MBUSWB и MBUSWW.

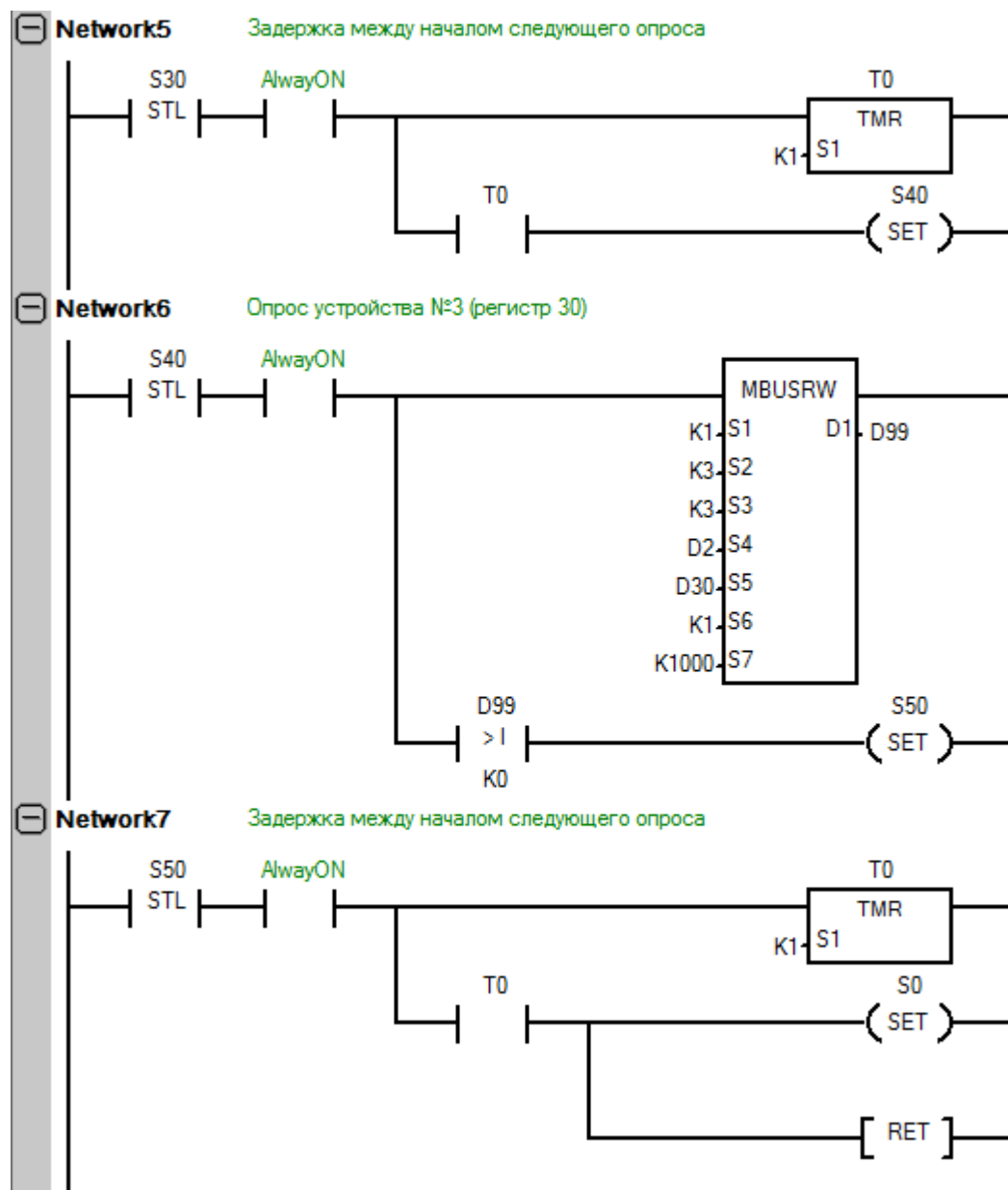
В один момент времени может выполняться только одна инструкция чтения или записи по одному порту. По этой причине, для организации опроса нескольких устройств требуется использование пошаговых инструкций STL.

Ниже представлен пример опроса трех Slave-устройств. Между шагами с инструкциями чтения дополнительно предусмотрены шаги задержек, чтобы гарантировать переключение ведомых устройств в режим «прослушки» линии. Дополнительно таймеры позволяют избежать вызова двух инструкций Modbus в одном цикле, т. к. последовательные шаги STL могут выполняться в пределах одного цикла.

Данный шаблон доступен для скачивания на сайте: <https://elhart.ru/>



Пример программы опроса нескольких устройств Modbus (начало)



Пример программы опроса нескольких устройств Modbus (окончание)

Это один из возможных способов организации программы. Также возможен вариант с использованием одной инструкции и динамической заменой значений адреса устройства и требуемых регистров.

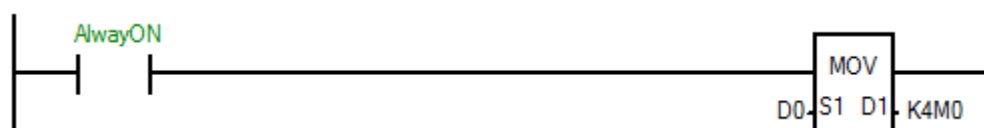
Завершением одного запроса в текущей реализации (и условием перехода на следующий шаг) является значение регистра D99, отличное от нуля. Это означает, что программа будет переходить на следующий шаг независимо от того, завершился предыдущий опрос успешно или с ошибкой. Возможно предусмотреть дополнительный обработчик ошибок связи, в случае если это требуется параметрами технологического процесса.

5.7 Получение и установка битовой маски

Зачастую, при опросе сторонних slave-устройств, возникает задача выделения определённых битов посылки. В качестве примера можно рассмотреть один из регистров частотного преобразователя ELHART EMD-MINI.

Адрес регистра (HEX)	Описание	
2000h	bit1 ~ bit0	00В: нет действия 01В: Стоп 10В: Пуск 11В: работа на частоте JOG
	bit3 ~ bit2	00В: нет действия 01В: вращение в обратном направлении 10В: вращение в прямом направлении 11В: сменить направление вращения
	bit4	0В: нет действия 1В: сброс аварии
	bit15 ~ bit5	резерв

Для преобразования 16-битного D-регистра в 16 последовательных M-регистров достаточно воспользоваться стандартной инструкцией MOV.



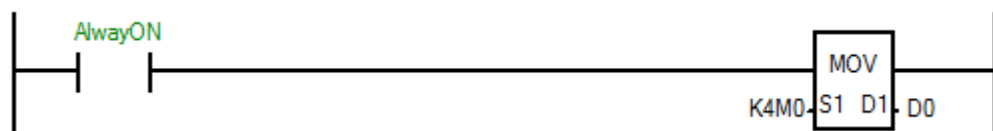
Преобразование 16-битного регистра в 16 последовательных одноканальных регистров

После преобразования младший бит будет находиться в регистре M0, в то время как самый старший – M15. Таким образом, определённому управляющему воздействию на частотный преобразователь будет соответствовать следующее состояние битов.

Управление работой				
	Нет действия	Стоп	Пуск	Работа на частоте JOG
M0	OFF	ON	OFF	ON
M1	OFF	OFF	ON	ON
Управление направлением движения				
	Нет действия	Вращение в обратном направлении	Вращение в прямом направлении	Сменить направление вращения
M2				
M3				
Сброс аварий				
	Нет действия		Сброс аварии	
M4	OFF		ON	
Резерв				
M5 ... M15				

Чтобы не расходовать память на резервные биты, выходу инструкции MOV требуется указать K2M0. В этом случае будут преобразованы только первые 8 битов регистра D0 (будут использованы внутренние регистры ПЛК с M0 по M7).

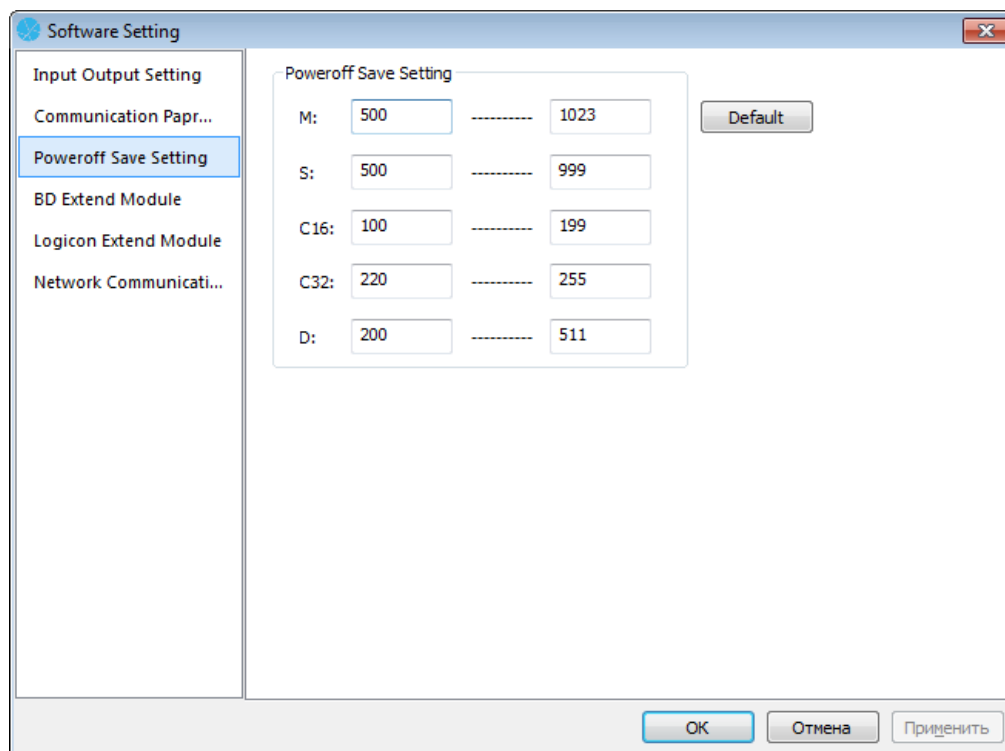
Обратное преобразование из 16 однобитных M-регистров в один 16-битный D-регистр происходит аналогичным образом.



Преобразование 16 последовательных однобитных регистров в один 16-битный регистр

5.8 Энергонезависимая память

Работа со встроенной энергонезависимой памятью не требует использование специальных инструкций. Все регистры, выбранные в окне «Poweroff Save Setting» автоматически становятся энергонезависимыми.



Окно конфигурации энергонезависимых регистров

Разрешается увеличивать или уменьшать количество регистров определённого типа, однако общий объём энергонезависимой памяти не может превышать 2 кБ.

При создании программы, в которой используется энергонезависимая память, рекомендуется предварительно обнулять значения использованных энергонезависимых регистров, т. к. эти регистры могут содержать значение отличное от 0.

5.9 Часы реального времени (RTC)

Встроенные часы реального времени доступны только для модификации ПЛК CPU 24/24.

Работа с часами осуществляется посредством взаимодействия с системными регистрами SD13-SD19.

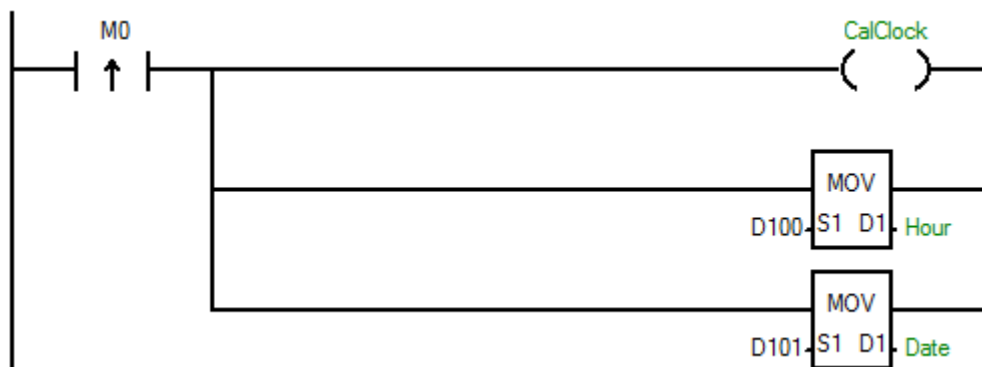
Номер регистра	Описание
SD13	Часы реального времени, секунды (0...59)
SD14	Часы реального времени, минуты (0...59)
SD15	Часы реального времени, часы (0...23)
SD16	Часы реального времени, день (1...31)
SD17	Часы реального времени, месяц (1...12)
SD18	Часы реального времени, год (0...99)
SD19	Часы реального времени, день недели (0...6)

Данные в указанных регистрах непрерывно обновляются.

Если активировать специальный битовый регистр SM16, то обновление значений SD13...SD19 прекратится, при этом фактическое значение часов реального времени продолжит изменяться.

Для настройки часов реального времени необходимо активировать системный регистр SM15, после этого можно перемещать требуемые значения в регистры SD13...SD19. Активация регистра SM15 полностью останавливает работу часов реального времени, поэтому рекомендуется проводить настройку импульсно.

Ниже представлен пример настройки часов реального времени: обновление значения часов из регистра D100 и значения дней из регистра D101.

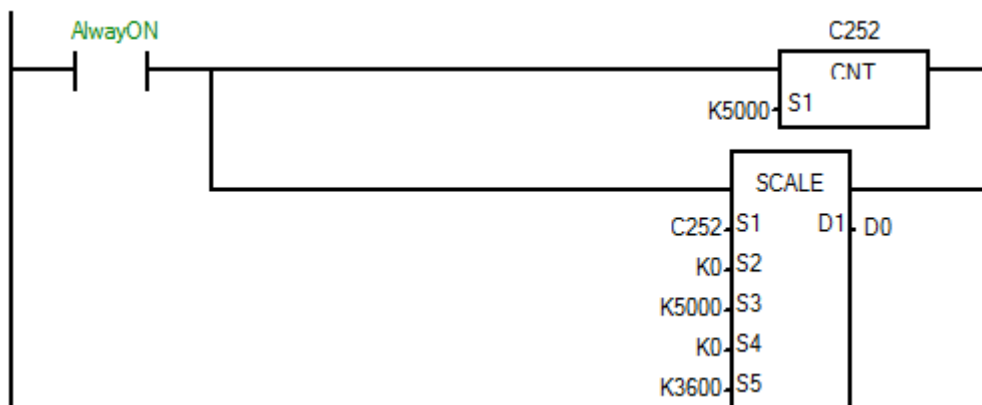


Обновление значений часов и дней в часах реального времени

5.10 Работа с энкодером

Для работы с энкодерами в ПЛК предусмотрены специальные счётчики. Более подробное описание различных типов счётчиков приведено в разделе «Программные элементы».

Ниже представлен пример программы, в которой импульсы от энкодера с разрешением 5000 имп/об преобразуются в угол поворота. В программе использован счетчик C252, поэтому выход Z энкодера должен быть подключен к входу X2 ПЛК (для сброса)

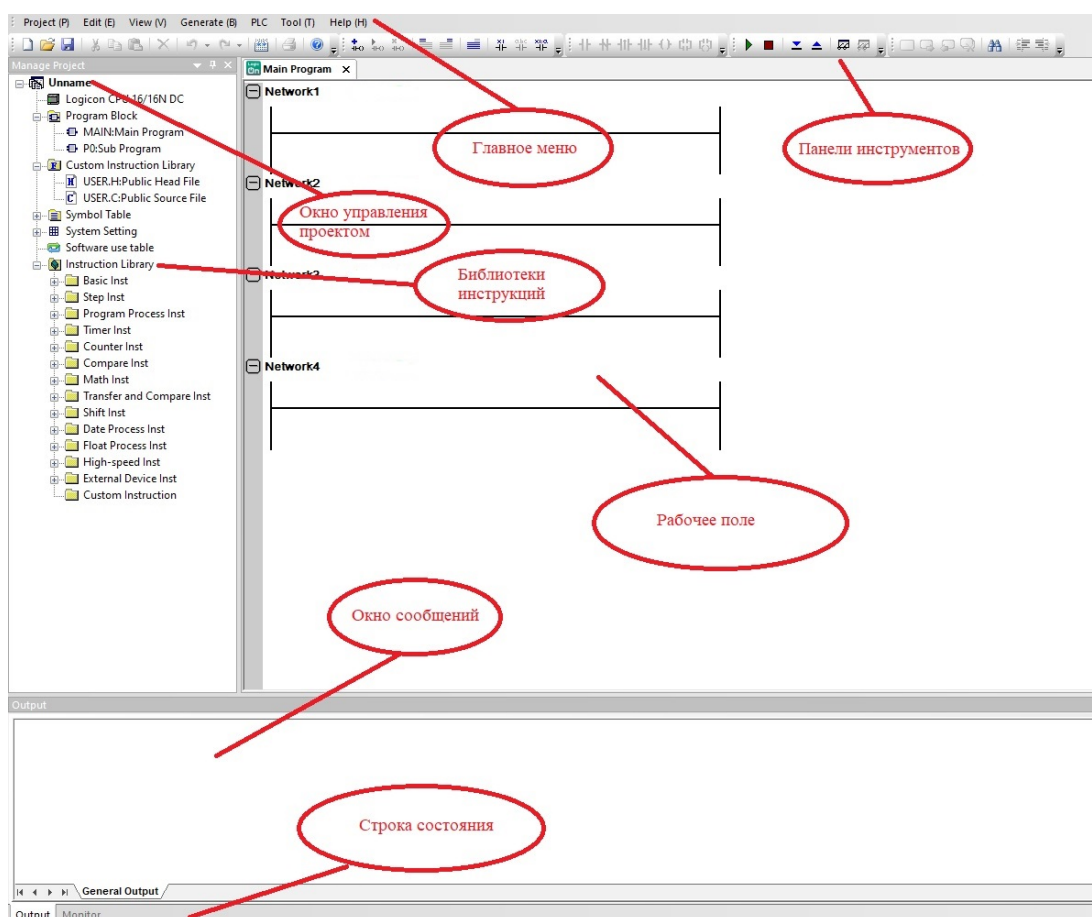


Подсчет импульсов от энкодера с помощью счётчика C252 и дальнейшее преобразование импульсов в угол 0 до 3600

ПРИЛОЖЕНИЕ 1 - Программное обеспечение

Описание интерфейса программы

Основной интерфейс ELHART Logicon Soft состоит из семи частей: главное меню, панели инструментов, окна управления проектом, библиотеки инструкций, окна сообщений, строки состояния и рабочего поля.



Основной интерфейс ELHART LogicOn Soft

Главное меню (Menu Bar)

Главное меню содержит следующие подпункты:

«Project» - содержит опции управления проектом.

«Edit» - содержит опции редактирования пользовательской программы.

«View» - включает в себя настройки интерфейса программы.

«Generate» - компиляция программы.

«PLC» - содержит инструменты работы с ПЛК -загрузка/выгрузка программы пользователя, запуск мониторинга и получения данных с контроллера.

«Tool» - содержит настройки программы и настройки параметров связи с ПЛК.

«Help» - справка по функциям и информация о версии среды разработки

Панели инструментов (Tool Bar)

Программное обеспечение предоставляет несколько панелей инструментов, включая быстрый доступ к рабочим командам. Основные операции можно осуществить с помощью меню или predetermined сочетаний клавиш.

Панели инструментов находятся под главным меню. По умолчанию отображаются все имеющиеся панели инструментов. Скрытие/добавления панелей инструментов производится нажатием правой кнопки мышки. При наведении курсора на элемент панели будет показана текстовая подсказка с названием данного элемента.

Главная панель инструментов (Main Tool Bar)



Главная панель инструментов

Главная панель инструментов включает в себя основные функции работы с проектом и редактирования программы ПЛК:



«New Project» - Создать новый проект



«Open Project» - Открыть проект



«Save Project» - Сохранить проект




«Cut» - Вырезать





«Copy» - Копировать



«Past» - Вставить


 «Delete» - Удалить

 «Undo» - Отменить

 «Redo» - Восстановить

 «Build» - Компиляция программы

 «Print» - Вывести на печать

 «About» - О программе


Панель инструментов редактирования программ (Program Editing Tools)


Панели инструментов редактирования программ содержит основные инструкции и операции по созданию программы, добавлению сети, вставке сети, удалению сети, настройке отображения комментариев.




Панель инструментов редактирования программ


 «Add» - Добавить сеть


 «Insert» - Вставить сеть


 «Delete» - Удалить сеть

 «Program Title» - показать/скрыть название программы

 «Network Title» - показать/скрыть название сети

 «Network Comment» - показать/ скрыть комментарий к сети

 «Operand» - отображать адрес элемента

 «Operand» - отображать символ элемента.



«Operand» - отображать адрес и символ элемента.

Панель инструментов редактирования программ на языке LD (Ladder node Tools)



Панель инструментов редактирования программ на языке LD

Данная панель инструментов позволяет добавлять основные элементы языка LD на рабочее поле.



«Insert Open Contact» - Вставить нормально открытый контакт



«Insert Close Contact» - Вставить нормально закрытый контакт



«Insert Rising Edge Contact» - Вставить детектор переднего фронта импульса.



«Insert Falling Edge Contact» - Вставить детектор заднего фронта импульса



«Insert output Contact» - Вставить выходную катушку



«Insert a parallel node» - Вставить контакт параллельно имеющимся



«Insert a parallel node» - Вставить выходную катушку параллельно имеющимся

Панель инструментов управления ПЛК (PLC Tools)



Панель инструментов управления ПЛК

Данная панель предоставляет пользователю возможность запускать и останавливать выполнение программы на ПЛК, производить загрузку программы в

ПЛК, выгружать программу из ПЛК, запускать и останавливать мониторинг исполнения программы ПЛК.



«Run» - Запуск выполнения программы



«Stop» - Остановка выполнения программы



«Download Program» - Загрузка программы в ПЛК



«Upload Program» - Выгрузка программы из ПЛК



«Start Monitor» - Запуск мониторинга исполнения программы



«Stop Monitor» - Остановка мониторинга исполнения программы

Панель инструментов редактирования кода(Code Editing Tools)



Панель инструментов редактирования кода

Данная панель инструментов позволяет произвести установку и удалению метки, переход к следующей/предыдущей метке, удаление всех меток, проверку, вставку/удаление разделителя TAB. Операции с метками доступны только в редакторе кода на языке С.



«Note» - Установить метку



«Next» - Переход к следующей метке



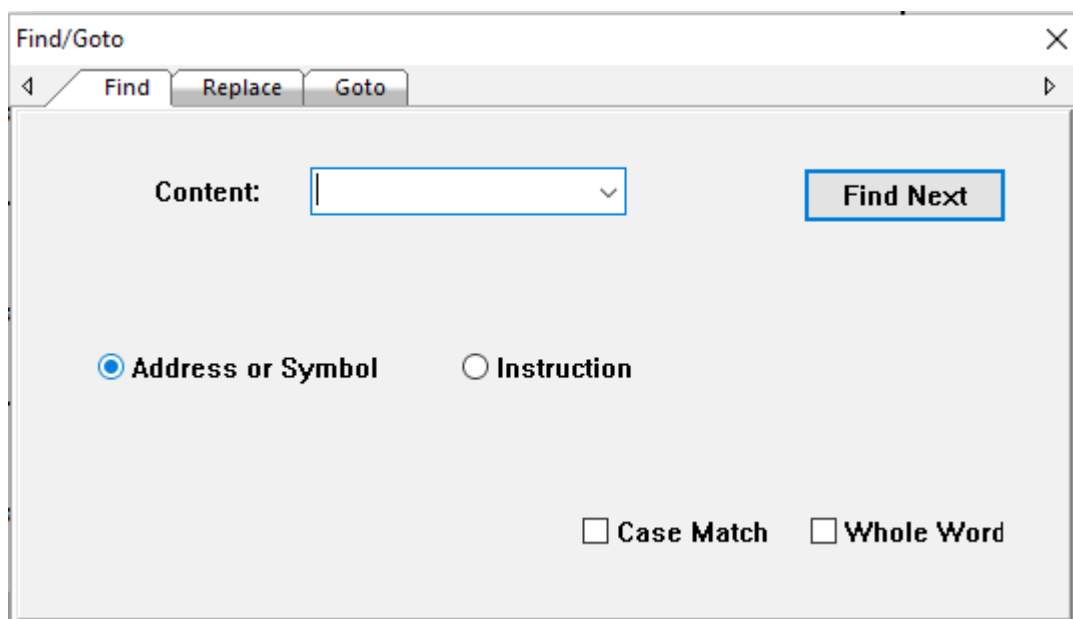
«Last» - Переход к предыдущей метке



«Delete All notes» - удалить все установленные метки



«Find» - Поиск. Позволяет найти, заменить или перейти к нужному адресу, символу или инструкции



Окно поиска



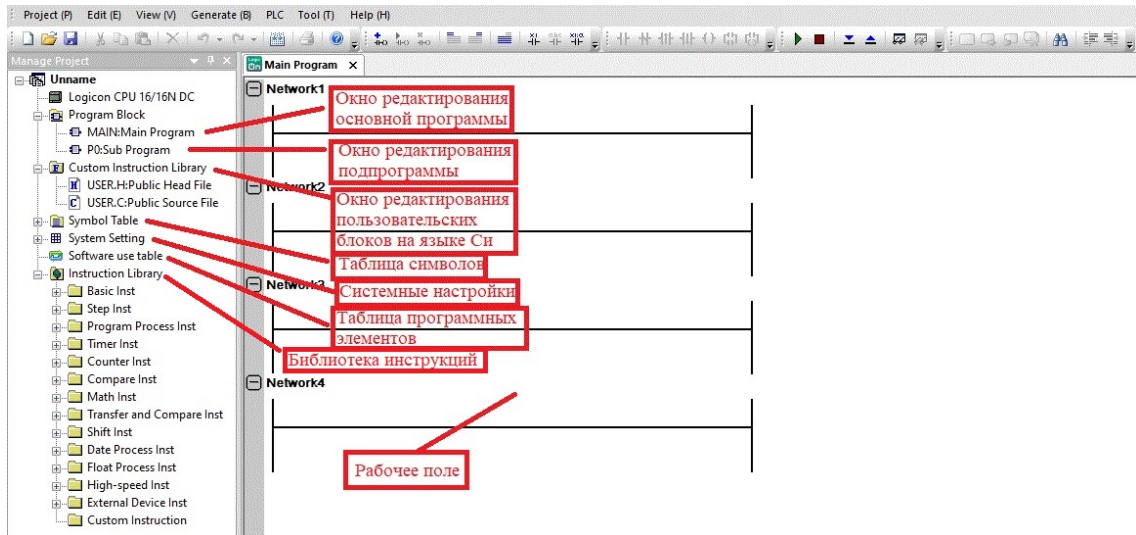
«Insert TAB delimiter» - вставить символ табуляции



«Delete TAB delimiter» - удалить символ табуляции

Рабочее поле и окно управления проектом

Рабочее поле и окно управления проектом включает в себя : рабочее поле, окно редактирования основной программы, окно редактирования подпрограммы, окно редактирования пользовательских блоков на языке Си, таблицы символов, системные настройки, таблицу программных элементов и библиотеку инструкций.



Рабочее поле и окно управления проектом

Окно редактирования основной программы(Main program)

«Main program» - переход к окну редактирования основной программы.

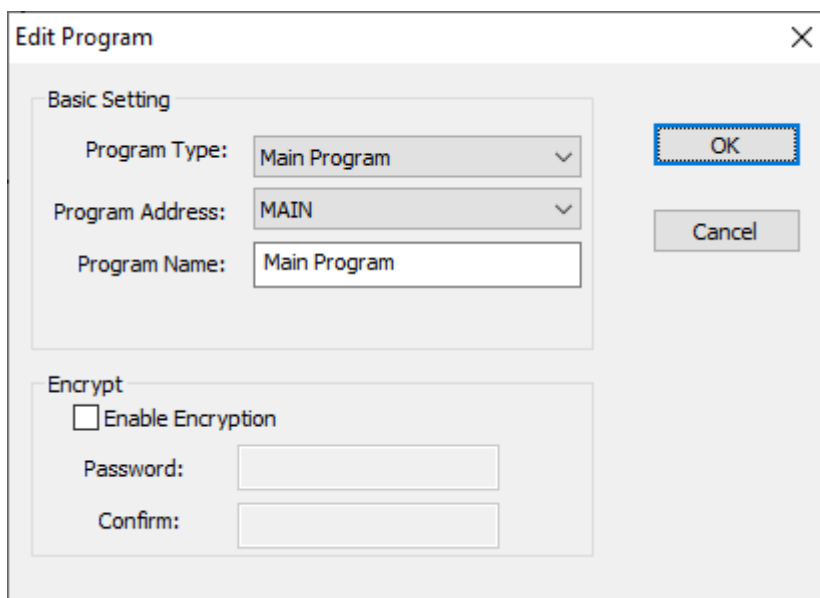
Щелчок правой кнопки мышки на пункте вызывает дополнительное меню в котором доступны следующие пункты:

«Insert SUB program» - добавить подпрограмму.

«Insert INT program» - добавить подпрограмму обработки прерывания.

«Delete» - Удалить подпрограмму.

«Propertys» - Характеристики основной программы. В данном пункте можно задать имя основной программы и включить/выключить защиту паролем.



Характеристики основной программы

Окно редактирования подпрограммы(Sub Programm)

«Sub Programm» - переход к окну редактирования подпрограммы.

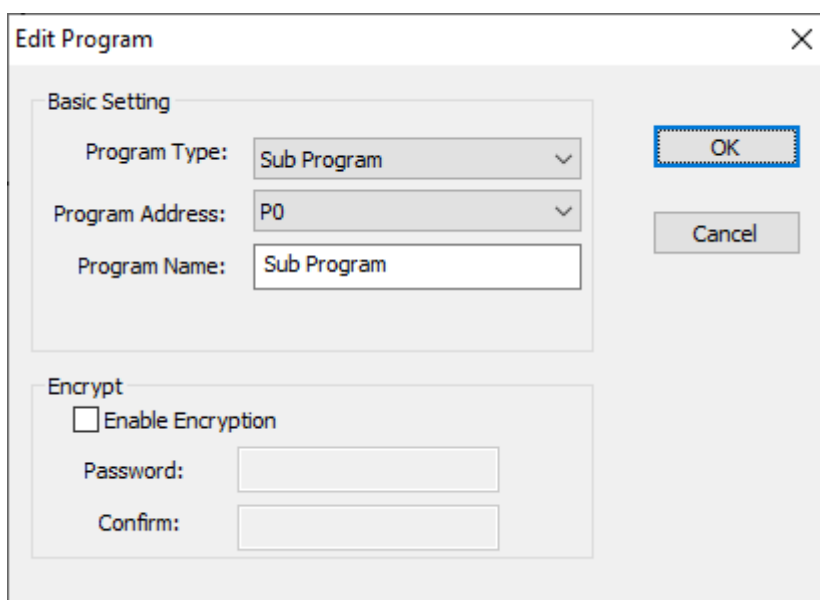
Щелчок правой кнопки мышки на пункте вызывает дополнительное меню в котором доступны следующие пункты:

«Insert SUB program» - Добавить подпрограмму.

«Insert INT program» - Добавить подпрограмму обработки прерывания.

«Delete» - Удалить подпрограмму .

«Propertys» - Характеристики подпрограммы. В этом пункте можно задать тип подпрограммы, адрес, по которому будет вызываться подпрограмма пользователя, или номер прерывания по которому будет вызвана подпрограмма обработки прерывания, имя подпрограммы и включить/выключить защиту паролем.



Характеристики подпрограммы

Окно редактирования пользовательских блоков (Custom instruction Library)

«Custom instruction Library» - переход к окну редактирования пользовательских блоков на языке Си. Щелчок правой кнопки мышки на данном пункте меню вызывает дополнительное меню, в котором доступны следующие пункты:

«Add Custom inst» - Добавить пользовательский блок.

«Delete Custom inst» - Удалить пользовательский блок.

«Export Custom inst» - Экспортировать пользовательский блок в виде отдельного файла.

«Export Custom inst» - Импортировать в программу пользовательский блок.

Таблицы символов (Symbol Table)

«Symbol Table» - Таблица доступных специальных битовых и словных регистров, а также присвоенные пользователем имена переменных.

«System Special Bit Address» - Таблица специальных битовых регистров.

«System Special Word Address» - Таблица специальных регистров данных.

«Custom_Symbol_Table» - Пользовательская таблица. Щелчок правой кнопки мышки на данном пункте меню вызывает дополнительное меню, в котором доступны следующие пункты:

«New User Symbol Table» - создать новую пользовательскую таблицу.

«Delete User Symbol Table» - удалить пользовательскую таблицу.

«Rename Table Name» - переименовать пользовательскую таблицу.

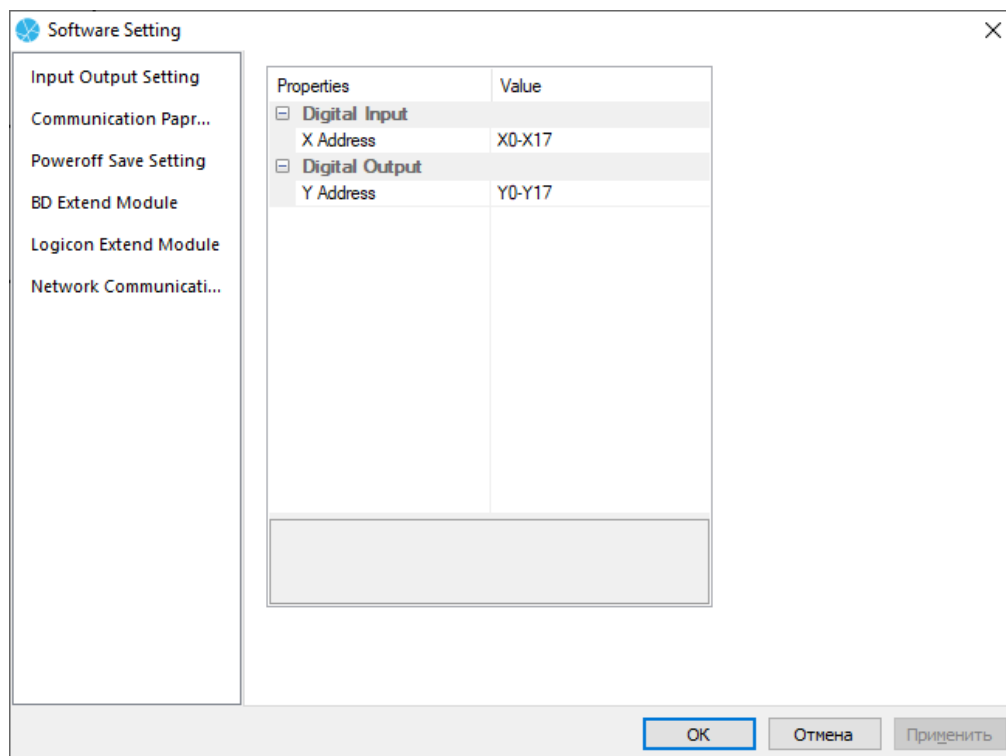
Удаление и переименование доступно только для пользовательских таблиц и недоступно для системных таблиц.

Пользовательская таблица символов содержит поля для имени символа, адреса и комментария. Таблица символов используется для изменения адресов (символов программных компонентов) на пользовательские имена для того чтобы сделать программу более понятной и наглядной. Имя программного компонента не может состоять полностью из цифр.

Системные настройки (System Setting)

«Input output setting» - Просмотр доступного количества входов и выходов.

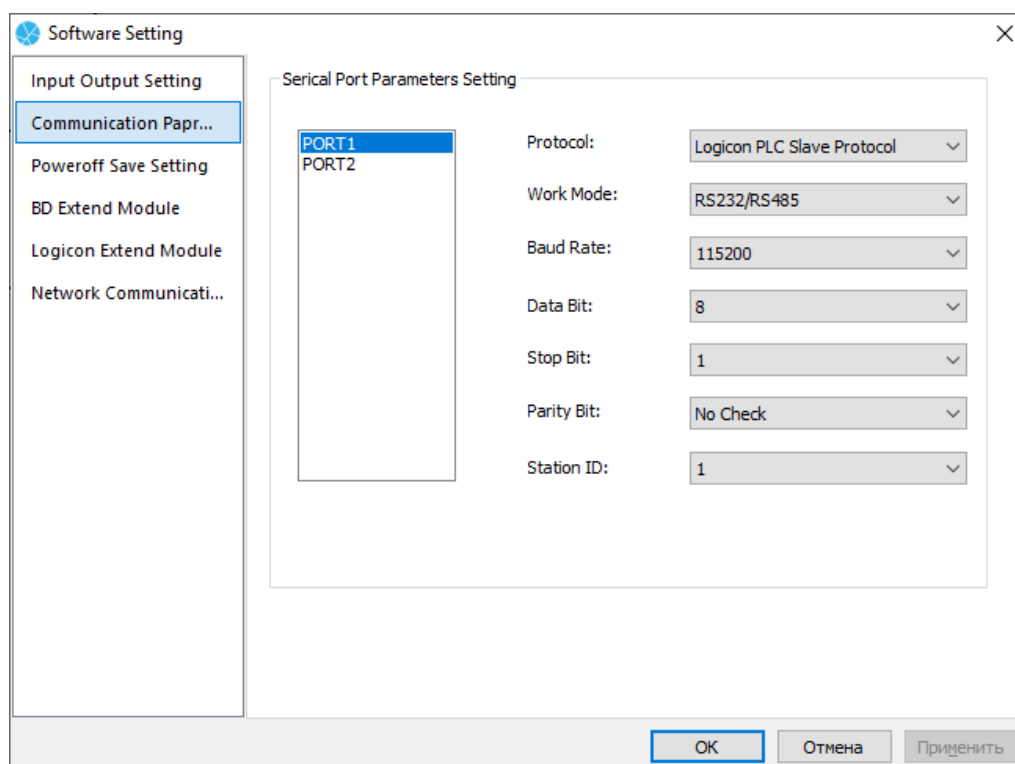
В данном пункте меню отображаются нумерация и количество доступных входов и выходов



Окно просмотра доступных входов и выходов

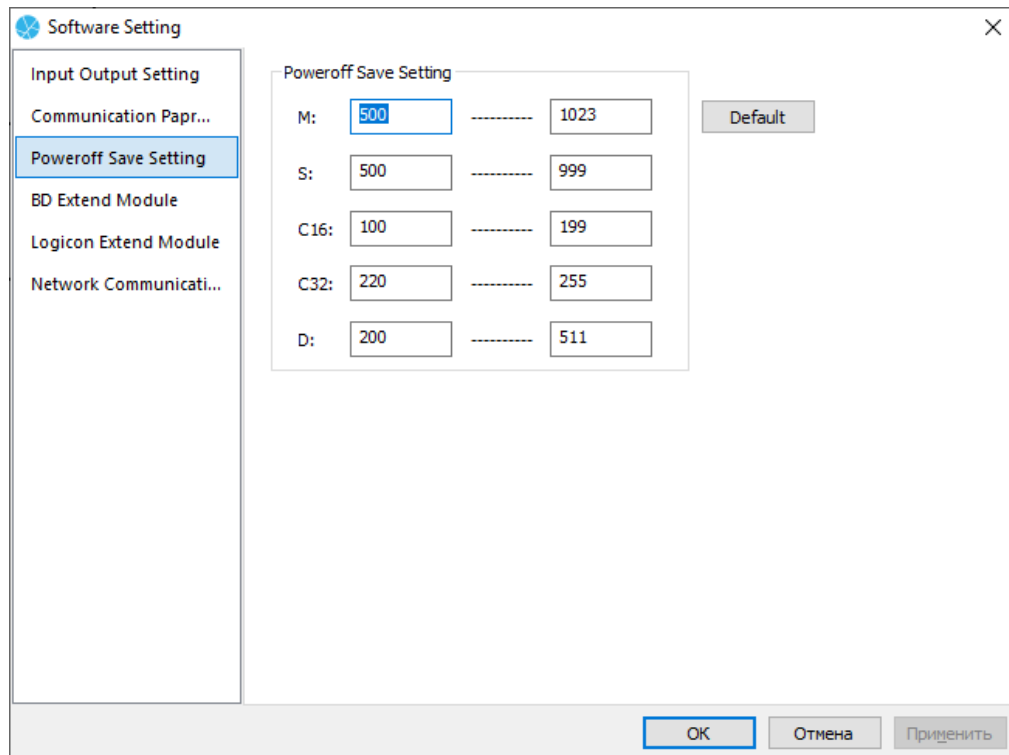
«Serial port setting» - Настройка коммуникационных портов. В данном меню доступна настройка интерфейсов связи PORT1 и PORT2. Для каждого порта доступны следующие настройки:

- «Baud Rate» - Скорость обмена.
- «Data Bit» - Количество бит данных.
- «Stop Bit» - Количество стоповых бит.
- «Station ID» - Сетевой адрес ПЛК.



Окно настройки коммуникационных портов

«Poweroff Save Setting» - Настройка диапазона адресов энергонезависимых регистров. Данные и состояния этих регистров будут сохранены после отключения питания.



Окно настройки диапазонов адресов энергонезависимых регистров

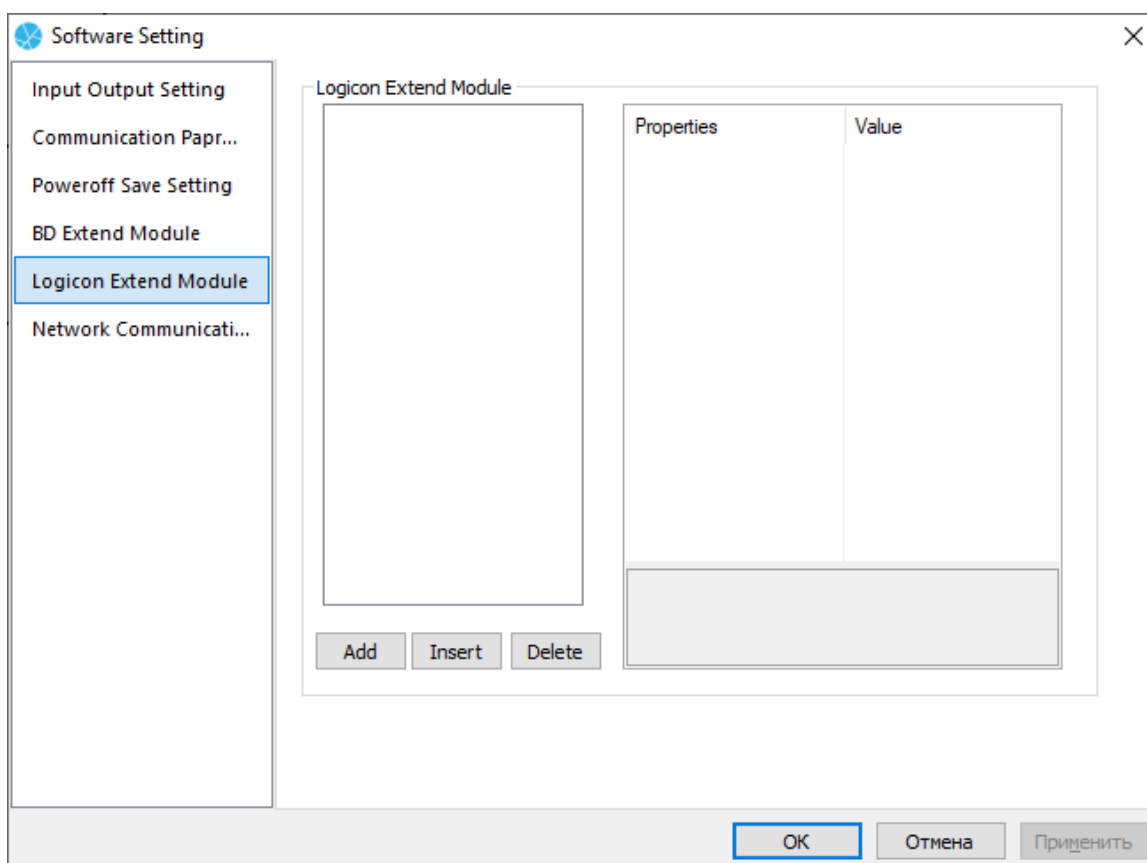
«Logicon Extend Module Setting» - Настройка модулей расширения подключенных по правосторонней шине расширения.

«Add» - Добавить модуль на позицию ниже.

«Insert» - Добавить модуль на позицию выше

«Delete» - Удалить модуль.

Модули в списке должны располагаться в таком же порядке, в каком они подключены к ПЛК. Например, первый модуль подключенный к ПЛК должен располагаться в первой строчке списка. Всего можно подключить не более восьми модулей к одному ПЛК.



Окно настройки модулей расширения

«Ethernet Communication Parameters Setting» - Настройка параметров связи по сети Ethernet.

«IP Address» - IP адрес контроллера.

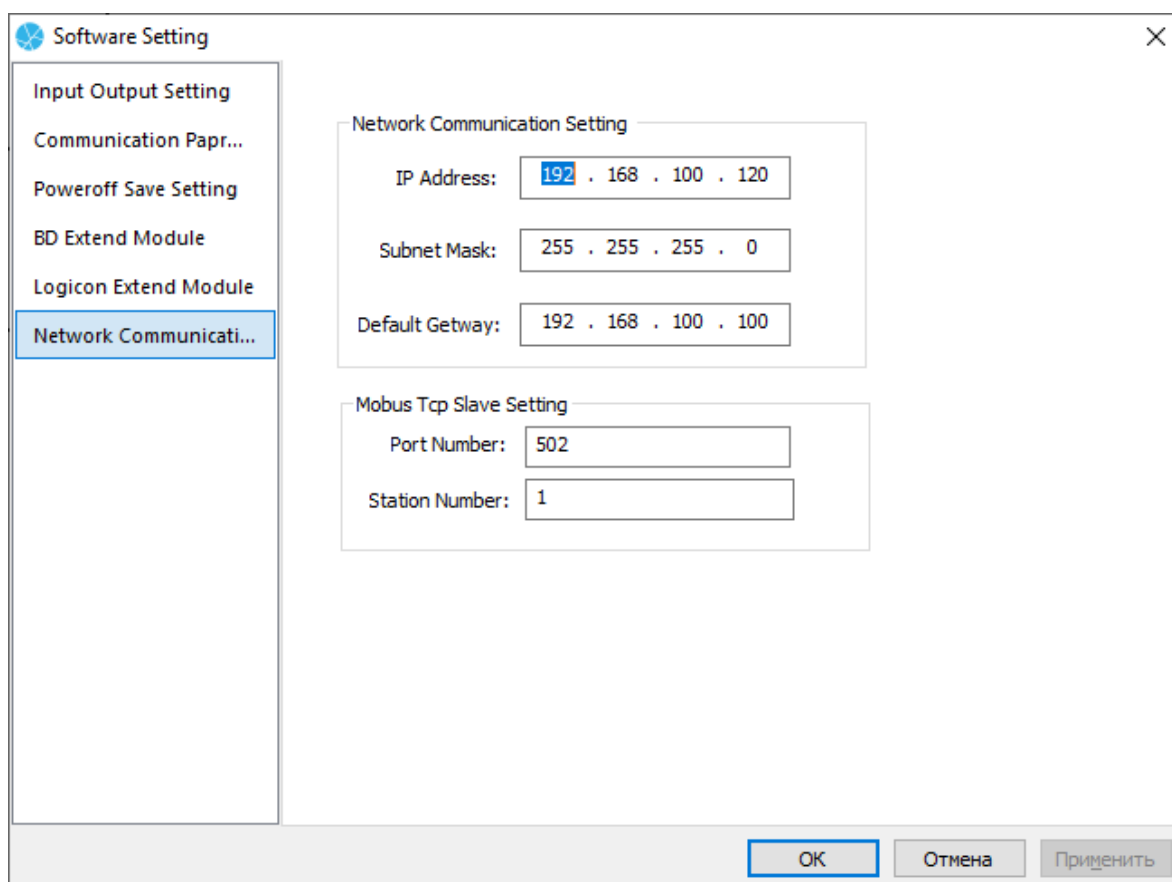
«Subnet Mask» - Маска подсети.

«Default Getway» - IP адрес шлюза.

«Port Number» - Номер порта при работе по протоколу Modbus TCP.

«Station Number» - Адрес ПЛК при работе по протоколу Modbus TCP.

Работа ПЛК по протоколу Modbus TCP возможна только в режиме Slave.



The screenshot shows a software configuration window titled "Software Setting". On the left is a sidebar menu with the following items: "Input Output Setting", "Communication Papr...", "Poweroff Save Setting", "BD Extend Module", "Logicon Extend Module", and "Network Communicati...". The "Network Communicati..." item is selected and highlighted in blue. The main area of the window is divided into two sections:

- Network Communication Setting:** Contains three input fields:
 - IP Address: 192 . 168 . 100 . 120
 - Subnet Mask: 255 . 255 . 255 . 0
 - Default Getway: 192 . 168 . 100 . 100
- Modbus Tcp Slave Setting:** Contains two input fields:
 - Port Number: 502
 - Station Number: 1

At the bottom right of the window, there are three buttons: "ОК", "Отмена", and "Применить".

Окно настройки связи по Ethernet

Таблица программных элементов(Software use table)

«Software use table» - Таблица программных элементов. В данном меню отображается количество использованных, доступных для использования и специальных программных элементов.



Элемент свободен для использования.



Системный элемент.



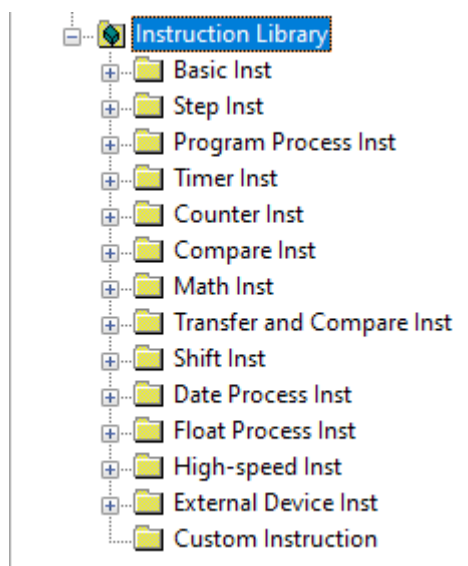
Элемент использован в программе пользователя.

X	Y	M	SM	S	T	C	D	SD	V	Z	AI	AQ						
				+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	ID	Component name	Program Bl...	Network	Instruction
				SM00C														
				SM01C														
				SM02C														
				SM03C														
				SM04C														
				SM05C														
				SM06C														
				SM07C														
				SM08C														
				SM09C														
				SM10C														
				SM11C														
				SM12C														
				SM13C														
				SM14C														
				SM15C														
				SM16C														
				SM17C														
				SM18C														
				SM19C														
				SM20C														
				SM21C														
				SM22C														
				SM23C														
				SM24C														

Таблица программных элементов

Библиотека инструкций (Instruction Library)

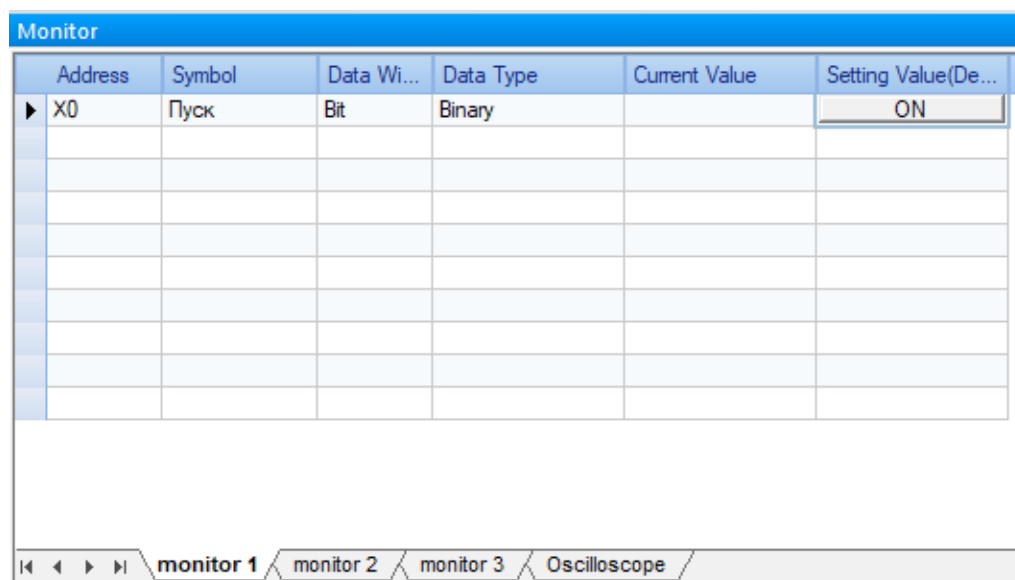
«Библиотека инструкций» (Instruction Library) содержит инструкции для написания программ на языке LD. Инструкции разбиты на группы по своему назначению. Для использования нужной инструкции необходимо выбрать инструкцию и с зажатой левой кнопкой мыши перенести элемент на рабочее поле.



Библиотека инструкций

Окно таблицы мониторинга (Monitoring Table Window)

Окно таблицы мониторинга используется для отслеживания значения каждого программного элемента в режиме реального времени и полезно для отладки программы. В таблице мониторинга отображается адрес элемента, обозначение, длина и тип данных, и их текущее значение. Значение программного элемента может быть изменено вводом заданного значения. Заданное значение каждого программного элемента вводится в десятичном виде.



Address	Symbol	Data Wi...	Data Type	Current Value	Setting Value(De...
X0	Пуск	Bit	Binary		ON

Окно таблицы мониторинга

Состояние элемента, длину данных, тип и текущее значение можно увидеть, введя адрес программного элемента в поле «Address», а значение устанавливаемого элемента можно ввести в поле «Setting Value»

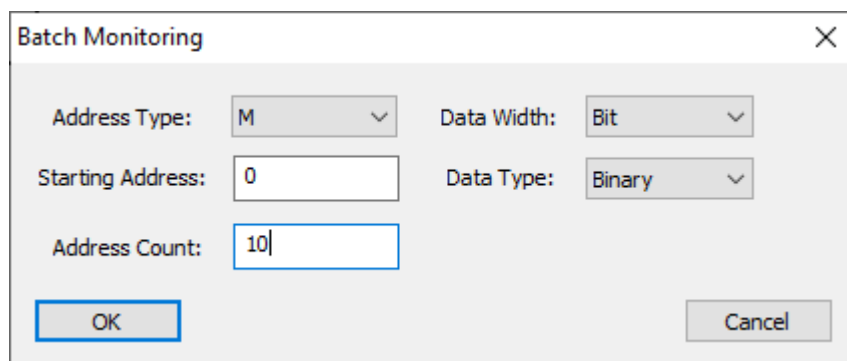
Для добавления или удаления строки в таблицу необходимо щелкнуть правой кнопкой мыши любую позицию в таблице мониторинга чтобы открыть диалоговое окно и выбрать необходимое действие:

«Insert line» - Вставить строку в таблицу выше.

«Add Line» - Добавить строку в таблицу ниже.

«Delete Line» - Удалить строку.

«Batch Monitoring» - Добавление сразу нескольких программных элементов одного типа. С помощью данной функции можно добавлять и контролировать группу элементов одного типа а также выбирать диапазон адресов.



Добавление нескольких элементов одного типа

Окно сообщений (General Output)

В окне сообщений выводится информация по этапам работы LogicOn Soft, компиляции программы, информация о возникших ошибках

Строка состояния (Status Bar)

В строке состояния отображается текущий режим работы ПЛК. Правым щелчком мыши на строке состояния вызывается меню где можно отключить или включить отображаемые элементы интерфейса

ПРИЛОЖЕНИЕ 2 - Программа «Светофор»

Пример реализует работу вызывного светофора с кнопкой пешехода. Для простоты отладки, время каждого шага уменьшено.

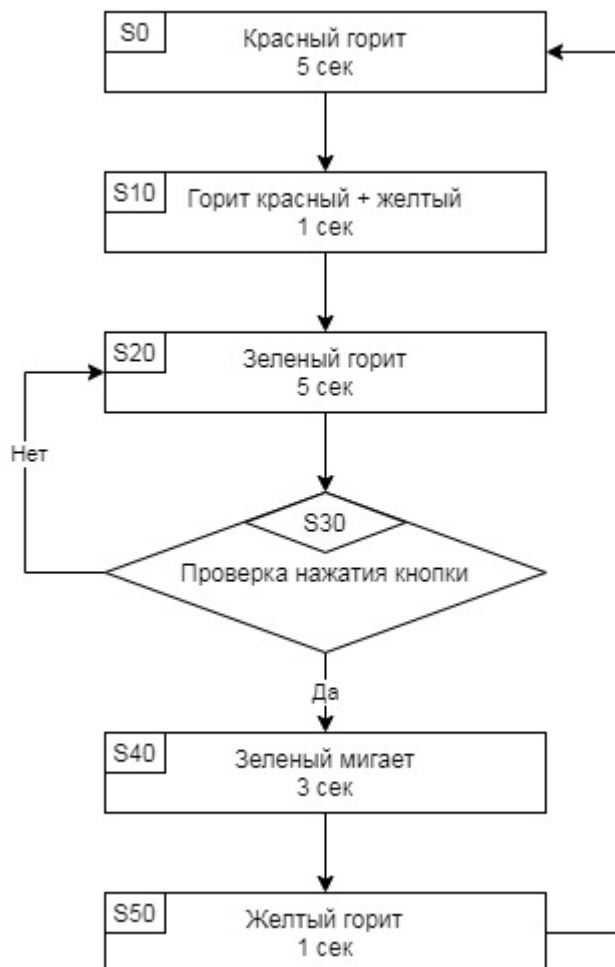
Данный пример демонстрирует использование таймеров и инструкций пошагового управления в ПЛК LogicOn.

Блок-схема алгоритма представлена справа.

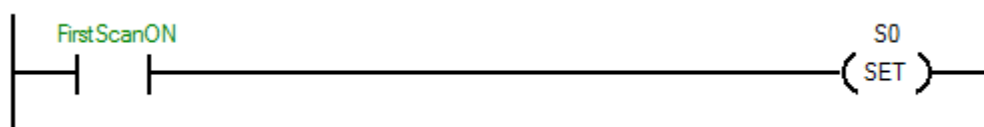
Из блок-схемы видно, что на различных шагах программы требуется управление одними и теми же выходами. К примеру, красный сигнал используется на шагах S0 и S10, желтый — на S10 и S50, а зеленый на S20 и S40.

Для того, чтобы условия разных шагов не накладывались друг на друга, следует использовать инструкции пошагового управления.

Диапазон шагов не случайно выбран кратным 10, это стандартная практика. Можно использовать и последовательные числа (1, 2, 3...). Но если в дальнейшем появится необходимость добавления шага между инструкциями 1 и 2, то придется использовать какое-то свободное число за пределами этого промежутка, что не логично.

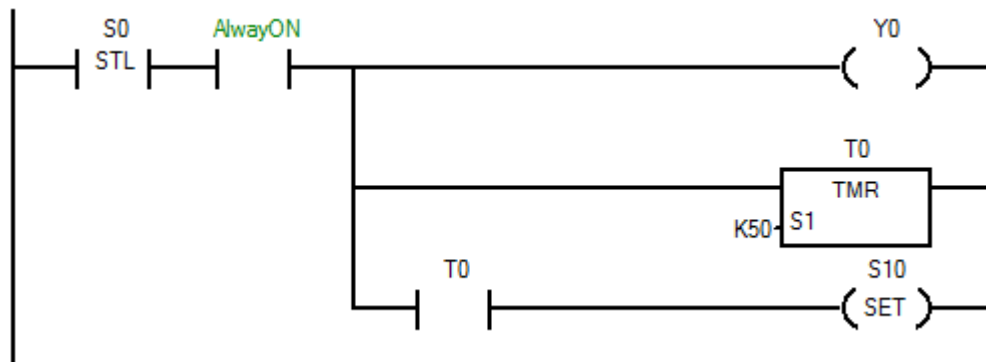


Запуск пошагового управления



Для запуска пошагового управления используется специальный регистр ПЛК FirstScanOn (SM2)*. Данный контакт срабатывает один раз при запуске ПЛК в работу. Таким образом в первом цикле активируется шаг №1 алгоритма.

Начало пошагового управления (горит красный свет)

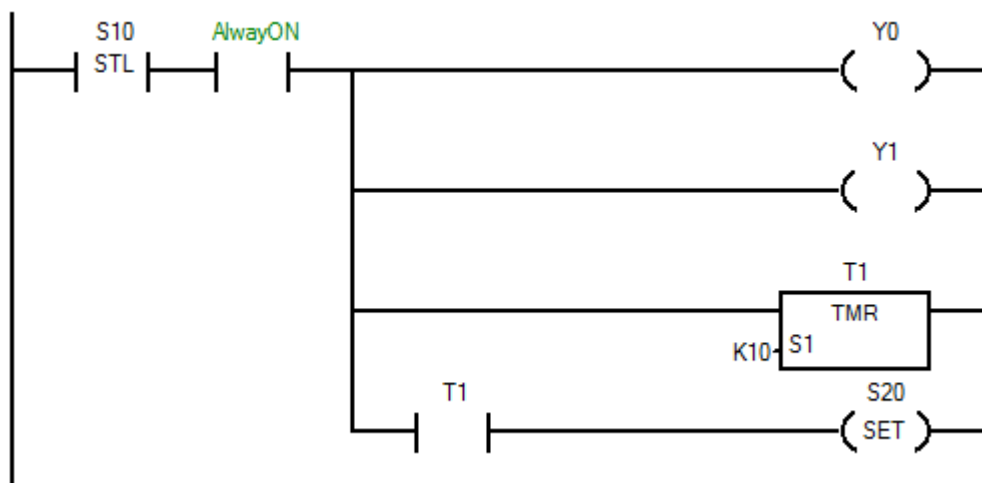


На этом шаге на 5 секунд включается красный свет (выход Y0).

По достижению уставки таймера T0 происходит активация выхода S10.

Обратите внимание, системный регистр AlwaysON (SM0) используется, так как подключение инструкций напрямую к контакту STL не рекомендуется.

Приготовиться к движению (одновременно горят красный и желтый)



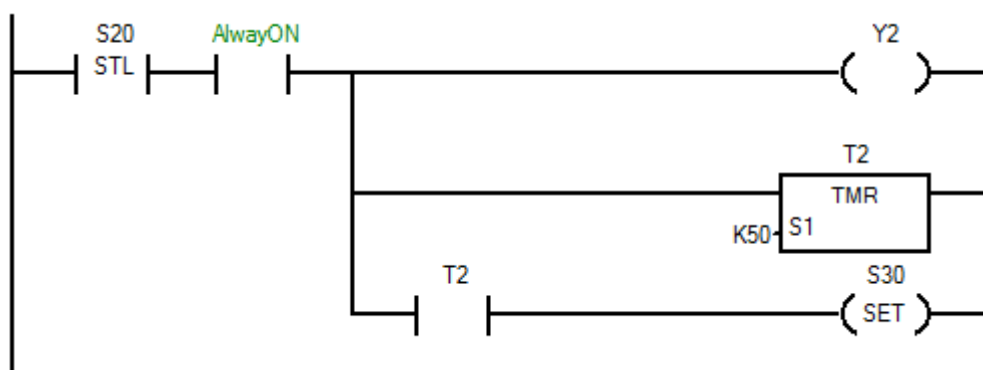
Работа этого шага аналогична предыдущему, только здесь включаются уже два сигнала светофора на время 1 сек.

Обратите внимание, сбрасывать предыдущий шаг S0 не нужно. Когда программа перейдет к исполнению S10, шаг S0 сбросится автоматически. То же

касается и таймера T0. Использование инструкций STL позволяет не думать о сбросе таймеров.

В этом шаге обязательно нужно снова указать выход Y0, в противном случае красный свет перестанет гореть сразу же после переключения шага S0.

Движение разрешено (горит зеленый свет)

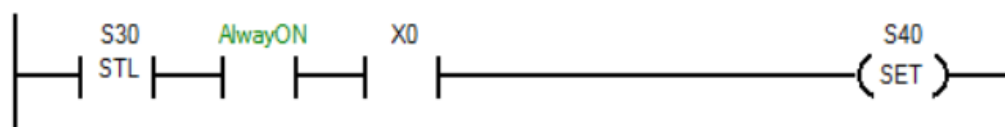


Данный шаг аналогичен предыдущим, только в этот раз включается зеленый свет.

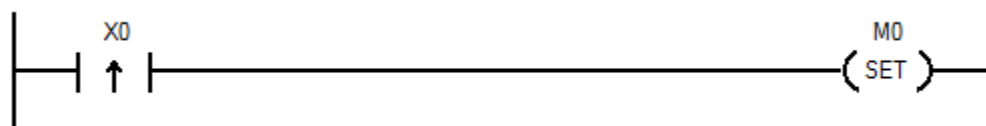
Проверка нажатия кнопки (выбор следующего шага)

На этом шаге требуется проверить состояние кнопки. Так как пешеход может нажать кнопку на любом шаге алгоритма, следует предусмотреть обработчик нажатия кнопки за пределом инструкций STL.

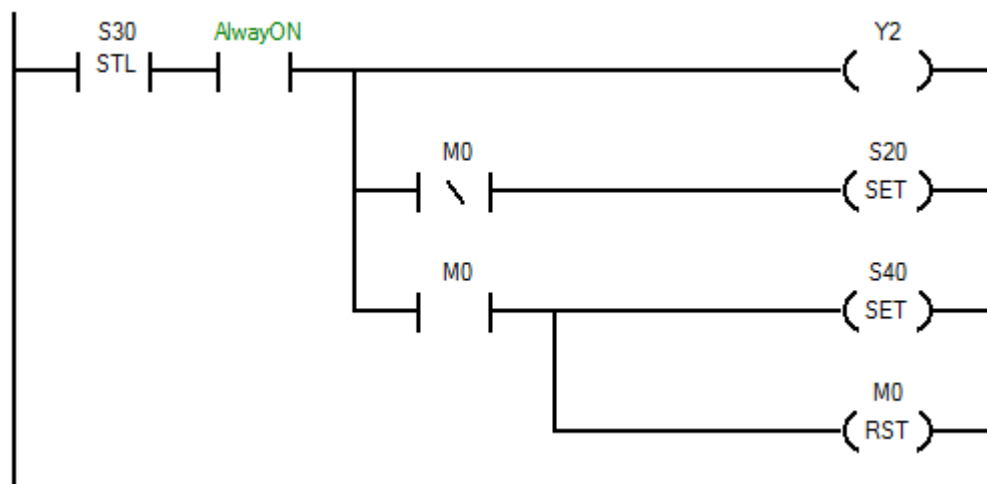
Иными словами, программу нельзя писать таким образом:



Т.к. если пользователь нажмет кнопку на любом другом шаге, например когда горит красный свет, программа этого не заметит. Поэтому обработчик нажатий кнопки выносится за пределы шагов STL. Чтобы избежать путаницы, сделать это лучше в самом начале или в самом конце программы (т.е. за пределами пошагового управления).



Данный код активирует вспомогательный регистр M0 при нажатии кнопки. Теперь, в зависимости от состояния этого регистра, будет происходить переход на следующий шаг (мигает зеленый), либо возврат на предыдущий (горит зеленый).

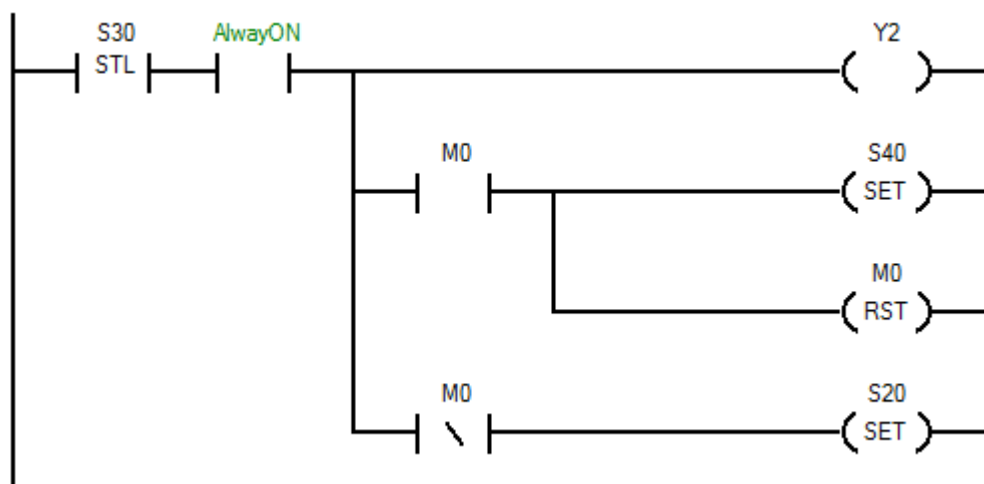


На шаге S30 нужно также включать выход Y2. Если этого не сделать, то зеленый свет будет выключаться на время одного цикла ПЛК (несколько миллисекунд).

Если кнопка пешехода не была нажата, то регистр M0 не активен, а значит происходит возврат на шаг S20. Если же кнопка нажата была, то произойдет переход на шаг S40. Параллельно будет сброшен регистр M0.

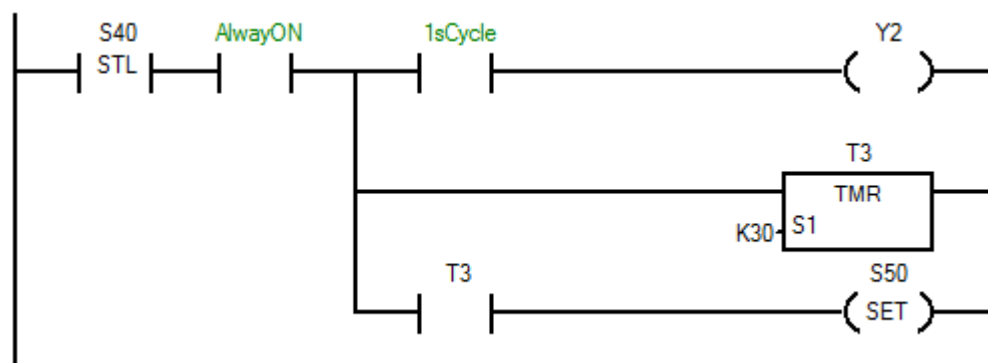
Важно: инструкции обязательно должны идти в порядке, указанном выше. Если изменить порядок выполнения, то алгоритм будет нарушен.

Ниже показан код, который вызовет ошибочное выполнение.



При написании программы таким образом, произойдет одновременный выбор шага S40 и S20. Это произойдет, т.к. после сброса M0 следующее условие тут же будет выполнено.

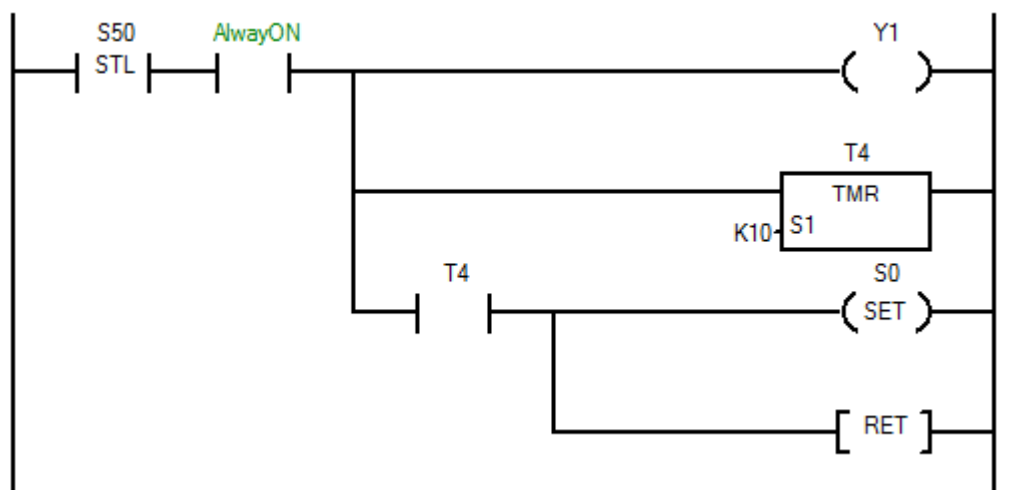
Предупреждение о переключении сигнала «Движение разрешено» (зелёный мигает)



Данный шаг аналогичен шагу S20. Дополнительно тут задействован системный регистр 1sCycle (SM13), который переключает свое состояние раз в половину секунды.

Окончание сигнала «Движение разрешено» (горит желтый)

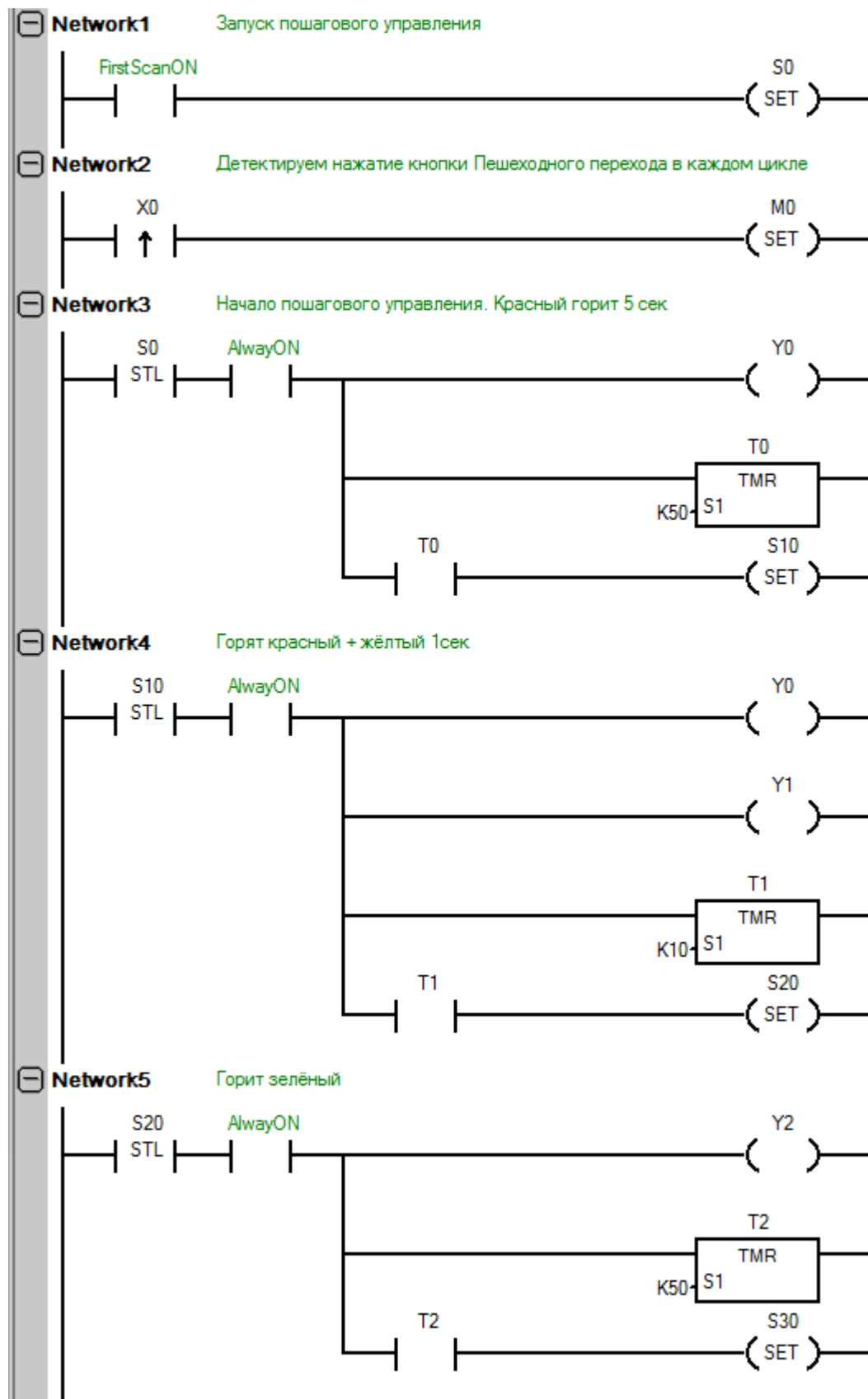
На последнем шаге происходит возврат управления первому шагу.

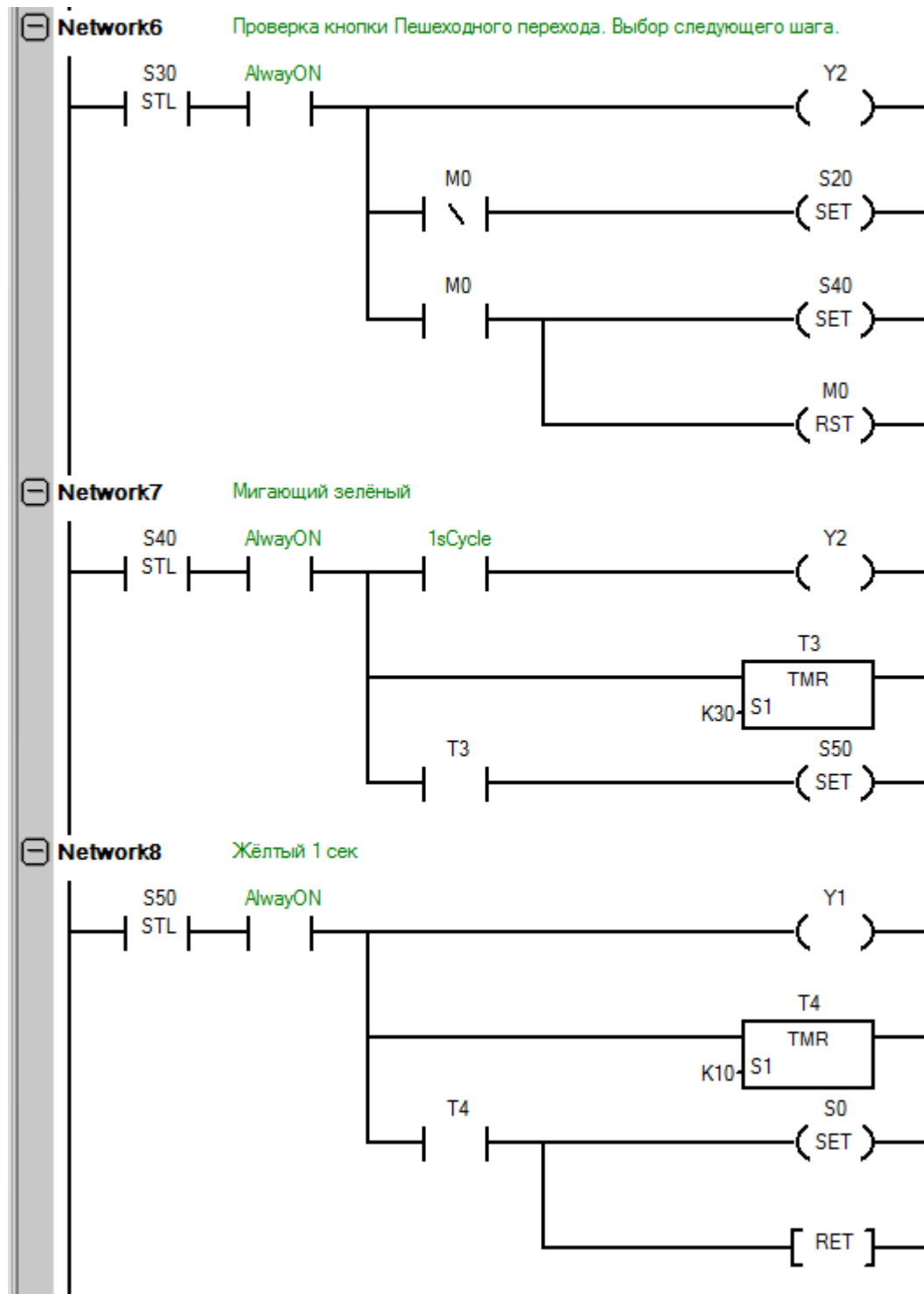


Логика данного шага аналогична предыдущему.

Т.к. это последняя инструкция STL, то обязательно использование инструкции RET.

Итоговая программа





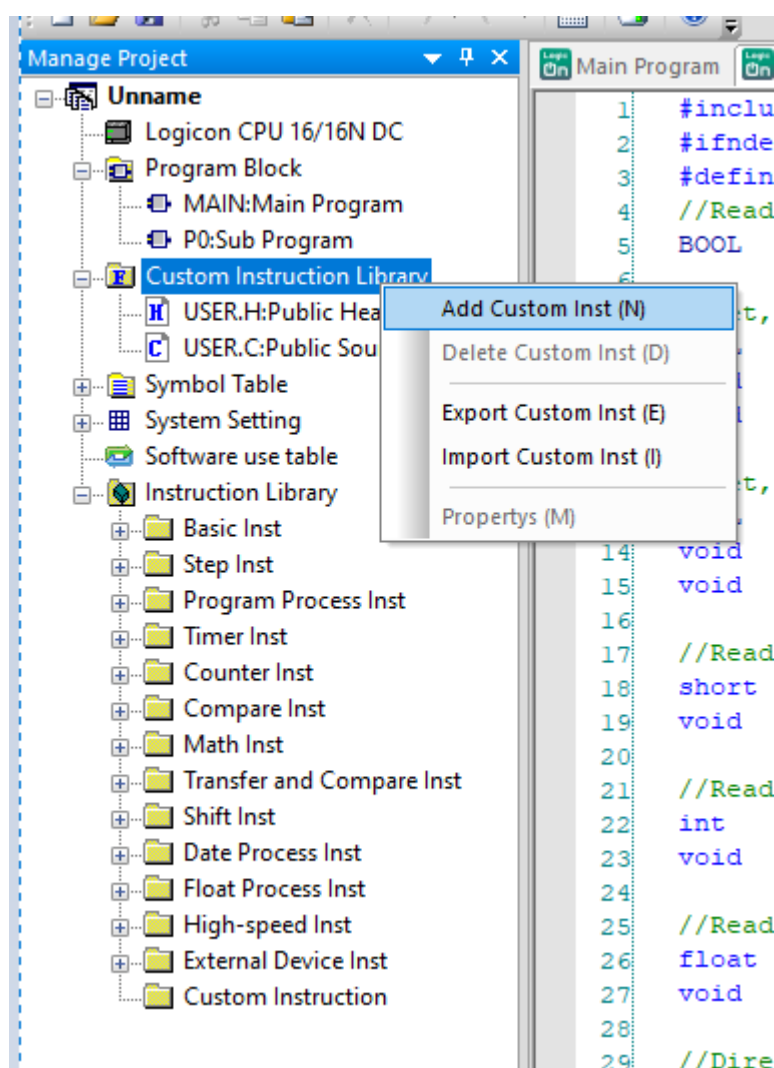
Данная программа демонстрирует основные преимущества инструкций пошагового управления. Условия в разных шагах не влияют друг на друга. Не нужно заботиться о сбросе таймеров. Представление программы становится более простым и понятным.

ПРИЛОЖЕНИЕ 3 - Создание макросов на языке C

Описание переменных, пример макроса для математического вычисления

Макросы на языке C добавляются в программу как «пользовательские инструкции», для этого необходимо:

1. Нажать правой кнопкой мыши на поле «Custom Instruction Library» и выбрать пункт «Add Custom Instruction»



2. В появившемся окне задать имя макроса, тип вызова (непрерывное или в импульсном режиме), общее количество параметров, выбрать входные и выходные параметры (доступно после задания общего количества), задать пароль на открытие макроса. Также возможно добавить общее описание макроса и задать имена параметров (не обязательно).

Создаваемые в теле макроса переменные, ссылающиеся на входные и выходные переменные макроса, получают автоматические имена вида "parax", где x - порядковый номер переменной. Далее, при редактировании макроса автоматические имена этих переменных можно заменять на собственные.

Edit the Custom Inst [X]

Name: Execute Mode: [v]

Describe:

Parameter Count: [v]

Parameter

Paramet...	Data Type	Input Ou...	Para Name
1	WORD	Input	para1
2	WORD	Input	para2
3	WORD	Output	

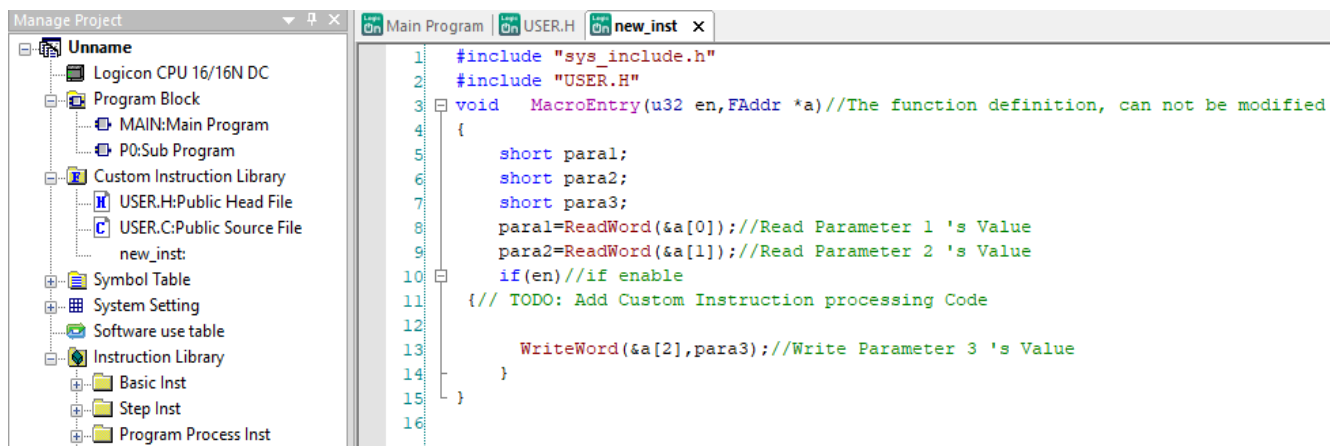
Encrypt

Enable Encryption

Password:

Confirm:

3. В результате в проект будет добавлен макрос следующего вида:



```
1 #include "sys_include.h"
2 #include "USER.H"
3 void MacroEntry(u32 en,FAddr *a)//The function definition, can not be modified
4 {
5     short para1;
6     short para2;
7     short para3;
8     para1=ReadWord(&a[0]);//Read Parameter 1 's Value
9     para2=ReadWord(&a[1]);//Read Parameter 2 's Value
10    if(en)//if enable
11    {
12        // TODO: Add Custom Instruction processing Code
13
14        WriteWord(&a[2],para3);//Write Parameter 3 's Value
15    }
16 }
```

Строка 3 - вход в тело макроса. В данном случае en - битовый признак выполнения инструкции макроса, а - указатель на массив входных переменных.

Строки 5 - 7 являются областью объявления переменных; поля short para1, short para2, short para3 являются переменными макроса типа short с диапазоном значений -32768..32767. При необходимости здесь же можно создавать локальные переменные. Доступными типами локальных переменных являются стандартные для языка C char, signed/unsigned char, int, unsigned int, short unsigned short, long, unsigned long, float, double, long double.

Получение значений входных переменных организовано в строках 8 и 9 при помощи системной инструкции ReadWord со ссылкой на адрес нужной переменной во входном массиве. Запись выходной переменной - в строке 13, при помощи инструкции WriteWord.

То есть, para1=ReadWord(&a[0]), para2=ReadWord(&a[1]) - входные переменные, WriteWord(&a[2],para3) - выходная переменная.

Телом макроса является область внутри фигурных скобок после инструкции if(en). Эта область предназначена для добавления собственного кода. Также, необходимо отметить, что при каждом вызове макроса из основной программы ПЛК происходит инициализация всех локальных переменных (заданными значениями или, если они не указаны, нулями), то есть значения локальных переменных не сохраняются в памяти ПЛК между вызовами макроса. Для сохранения значений переменных между вызовами макроса необходимо добавлять спецификатор static при объявлении переменной.

Добавим простую формулу для расчёта суммы двух переменных с записью полученного значения в третью:

```
para3 = para1 + para2;
```

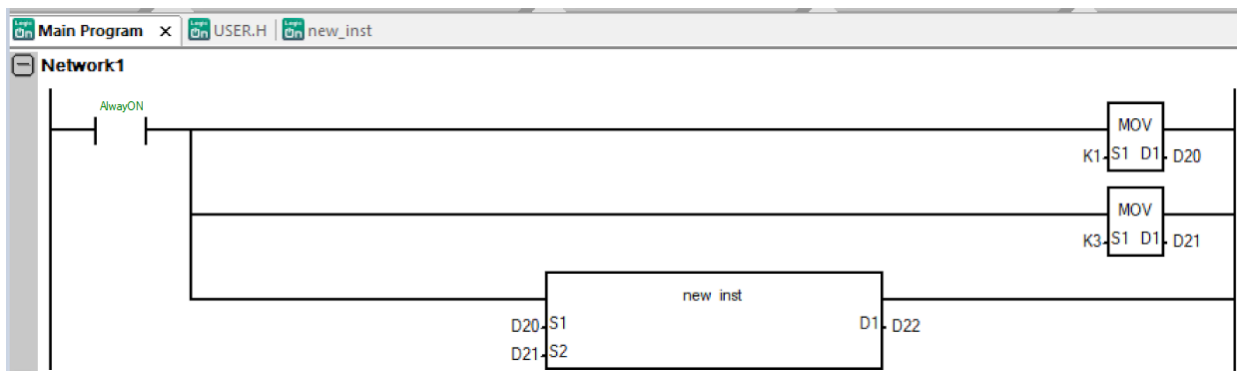
Проект необходимо скомпилировать (Ctrl + B) и убедиться в отсутствии ошибок на данном шаге.

```
----Start Compile Project ----
>[Main Program]Start Conversion...!
>[Main Program]Convert Success!
>[Sub Program]Start Conversion...!
>[Sub Program]Convert Success!
User program version: 1 ,MD5: 8d82f79db9e314dcb8ee340b4d0db1cc
User program space: 9152 bytes
Archiving Date: 2023 / 5 / 2 13:22:23
Build Executable Package: C:\Users\pnp\Documents\check_1\check_1.fld
----Project Compiled Over!----
```

4. Созданный макрос появится в Instruction Library – Custom Instruction. В качестве примера данный макрос можно добавить в проект в Main Program. Предварительно в первую ветку Main Program необходимо добавить системный бит SM0 (включен при работающем ПЛК) и значения двух регистров (адреса задаются произвольно), которые будут подаваться на вход экземпляра макроса:



Макрос из библиотеки можно добавить в эту же ветку путём перетаскивания мышкой. Входам и выходу макроса необходимо присвоить адреса регистров в памяти ПЛК:



Далее следует скомпилировать проект и убедиться в отсутствии ошибок.

При загрузке текущего проекта на ПЛК в регистре D22 появится результат операции макроса, равный 4.

Переменные со спецификатором `static`, работа с системным временем и внутренними регистрами ПЛК

Для сохранения значений переменных между системными вызовами макроса используется спецификатор класса хранения `static`. Данный спецификатор указывает компилятору на хранение локальной переменной вместо её инициализации при каждом вызове макроса. Подобная опция позволяет эффективно работать в макросе с системным временем ПЛК, например, для создания таймеров и задержек внутри макроса. В примере ниже будет рассмотрено подобное применение спецификатора `static` для организации циклического включения/выключения внутреннего реле ПЛК M1 (“меандр”).

Код макроса:

```
#include "sys_include.h"
#include "USER.H"

void MacroEntry(u32 en, FAddr *a) //The function definition, can not
be modified
{
    unsigned int cycle_time, system_time; //время цикла (в данном
примере будет задаваться в 100 мс), системное время ПЛК
    static unsigned int start_time; //сохранённое значение интервала
времени от начала цикла, спецификатор static
    static BOOL start_timer, output; //признак начала интервала,
вспомогательный бит для управления битом в памяти ПЛК
```

```

cycle_time = ReadWord(&a[0]); //запись значения, подаваемого на
вход макроса, в переменную времени цикла

if(en)//if enable
{// TODO: Add Custom Instruction processing Code
//каждый цикл ПЛК получаем значение системного времени ПЛК
system_time = GetSysTick();
//если признак начала интервала не установлен
if (!start_timer) {
//установить признак начала интервала
start_timer = TRUE;
//приравнять текущее системное время в сохранённое значение
start_time = system_time;
//поменять состояние вспомогательного бита
output = !output;
//если вспомогательный бит установлен в 1, то установить M1 в 1,
иначе установить M1 в 0; состояние будет изменяться через интервал
"время цикла, умноженное на 100"
if (output) SET_M(1);
else RST_M(1);
}

//если текущее системное время ПЛК больше сохранённого на момент
начала отсчёта плюс время цикла, умноженное на 100 (минимальный
интервал - 1 сек)
if (system_time >= (start_time + cycle_time * 100)) {
//сбросить признак начала интервала
start_timer = FALSE;
//записать в выходную переменную макроса текущее значение
системного времени; данная переменная будет изменяться через
интервал "время цикла, умноженное на 100"
WriteDWord(&a[1],system_time);//Write Parameter 1 's Value
}
}
}

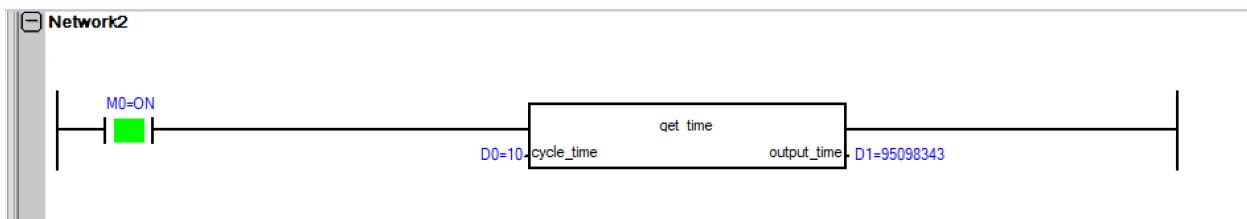
```

Общее описание: макрос запрашивает системное время ПЛК каждый цикл выполнения; если не был установлен признак начала отсчёта временного интервала, он устанавливается, одновременно с этим текущее системное время записывается в переменную со спецификатором *static* (при следующем вызове такая переменная сохранит своё значение), а также инвертируется состояние вспомогательного бита и бита в памяти ПЛК. Так как системное время в переменной макроса обновляется постоянно, а сохранённое значение на момент начала интервала остаётся неизменным, то по истечении интервала времени, равного «сохранённое время начала плюс время цикла», признак начала отсчёта сбрасывается, выходной переменной макроса передаётся текущее значение

системного времени (в данном случае для визуального контроля исполнения кода макроса).

В качестве примера работы с регистрами памяти ПЛК в данном примере используется установка и сброс бита M1.

В основную программу ПЛК данный макрос добавляется также, как в предыдущем примере («Создание макросов»).



Время цикла в данном случае равно 1 сек. (задаётся в сотнях мсек). Состояние бита M1 можно проконтролировать в мониторе (будет изменяться на противоположное 1 раз в секунду):

Address	Symbol	Data Wi...	Data Type	Current Value	Setting Value(De...
D0		WORD	Signed	10	
D1		DWORD	Signed	95215437	
M1		Bit	Binary	ON	OFF