МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)

Д. А. Тарасов, А. В. Липов, Г. С. Большаков

МОДЕЛИРОВАНИЕ РАБОТЫ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ Delta Electronics В РЕДАКТОРЕ WPLsoft

Учебное пособие

Пенза
Издательство ПГУ
2019

УДК 681.5(075)
Т19

Р е ц е н з е н т ы :

технический директор
ООО «Специальное конструкторское бюро турбонагнетателей» (г. Пенза)
Г. В. Тарабрин;

кандидат технических наук, доцент,
заведующий кафедрой «Автоматизированные системы безопасности»
Пензенского государственного университета
А. Е. Шаралапов

Тарасов, Д. А.

Т19 Моделирование работы программируемых логических контроллеров Delta Electronics в редакторе WPLSoft : учеб. пособие / Д. А.Тарасов, А. В. Липов, Г. С. Большаков. – Пенза : Изд-во ПГУ, 2019. – 100 с.

ISBN 978-5-907185-22-7

Содержится теоретический и практический материал по применению программируемых логических контроллеров Delta Electronics в редакторе WPLSoft в системах управления технологическим оборудованием.

Издание подготовлено на кафедре «Компьютерное и технологическое оборудование» Пензенского государственного университета и предназначено для обучающихся по направлениям подготовки 15.03.02 «Технологические машины и оборудование» и 15.04.05 «Конструкторско-технологическое обеспечение машиностроительных производств».

УДК 681.5(075)

ISBN 978-5-907185-22-7

© Пензенский государственный университет, 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 5

1. ОСНОВНЫЕ ПОНЯТИЯ И ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ ПРОМЫШЛЕННЫХ КОНТРОЛЛЕРОВ 7
1.1. Принципы построения ступенчатых диаграмм 7
1.2. Различия в работе традиционных релейно-контактных схем и ступенчатых диаграмм 11
1.3. Внутренние объекты программируемых логических контроллеров (операнды) 14
1.4. Условные обозначения элементов ступенчатых диаграмм 21
1.5. Основные правила написания ступенчатых диаграмм 25
1.6. Прикладные инструкции программируемых логических контроллеров 27

2. РЕДАКТОР WPLsoft 29
2.1. Окно редактора 29
2.2. Создание нового проекта. Окна редактирования логики проекта 30
2.3. Сохранение проекта 31
2.4. Закрытие проекта. Выход из редактора 32

3. ДЕМОНСТРАЦИОННЫЙ ПРОЕКТ 33
3.1. Открытие созданного ранее проекта 33
3.2. Создание простой логики 33
3.3. Работа с таймерами и счетчиками 37
3.4. Добавление RS-триггера 40
3.5. Импульсные выходы. Добавление пустых строк 41
3.6. Отключаемые блоки на ступенчатой диаграмме 42
3.7. Инверсия. Логический элемент ИЛИ-НЕ 44
3.8. Кнопки редактирования вертикальных соединений 44
3.9. Отображение комментариев 45
3.10. Компиляция проекта 47
3.11. Типичные ошибки при написании ступенчатых диаграмм 48

4. МОНИТОРИНГ ВЫПОЛНЕНИЯ ПРОГРАММ 51

5. ОПИСАНИЕ ПРИКЛАДНЫХ ИНСТРУКЦИЙ 54
5.1. Сравнение числовых данных (CMP) 54
5.2. Зонное сравнение числовых данных (ZCP) 55
5.3. Пересылка данных (MOV) 57
5.4. Пересылка блока данных (BMOV) 58
5.5. Сложение двух чисел (ADD) 60
5.6. Вычитание двух чисел (SUB) 61
5.7. Умножение двух чисел (MUL) 63
5.8. Деление двух чисел (DIV) 64
5.9. Инкрементирование (увеличение на 1 INC) 65
5.10. Декрементирование (уменьшение на 1 DEC) 66
5.11. Кольцевой сдвиг влево/вправо (ROL/ROR) 67

5.12. Высокоскоростная обработка (высокоскоростной счет DHSCS).....69
5.13. Управление параметрами ПИД-регулятора преобразователя частоты (FPID).....71
5.14. Управление выходной частотой преобразователя частоты (FREQ) 72
5.15. Чтение/запись параметров преобразователя частоты (RPR/WPR) 73
6. ПРИМЕРЫ НАПИСАНИЯ ПРОГРАММ 76
6.1. Самоблокировка выхода с приоритетом кнопки СТОП.....76
6.2. Самоблокировка выхода с приоритетом кнопки СТАРТ.....77
6.3. Самоблокировка выхода с использованием команд SET и RST 77
6.4. Схема с условным управлением 77
6.5. Схема с взаимоблокировкой 78
6.6. Схема с последовательным управлением 78
6.7. Мерцающая схема 79
6.8. Триггерная схема.....79
6.9. Схема с задержкой таймера на выключение 80
6.10. Схема с задержкой таймера на включение и выключение 80
7. ПРИМЕРЫ ПРОГРАММИРОВАНИЯ.....81
7.1. Включение освещения с автоматическим отключением 81
7.2. Управление уровнем смазочно-охлаждающей жидкости в баке станка 82
7.3. Управление освещением.....84
7.4. Ступенчатый выключатель вентилятора 85
7.5. Контроль длительности использования инструмента 86
7.6. Управление системой подачи защитного газа автомата для наплавки 89
7.7. Управление подъемным столом-перегружателем 91
7.8. Последовательный запуск трех двигателей.....93
7.9. Педальный переключатель для выбора скоростей станка 94
7.10. Цикловое управление приводом продольного перемещения суппорта токарного станка 96
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 98

ВВЕДЕНИЕ

До середины прошлого века автоматизация различных процессов и технологического оборудования в основном проводилась на базе релейно-контактных схем управления электроприводом, собранных на дискретных компонентах – реле, таймерах, счетчиках, элементах жесткой логики. При этом с увеличением количества исполнительных органов и вариантов их включения требуется большое число элементов релейно-контактной системы управления, что иногда вызывает необходимость наличия значительных объемов пространства для их размещения. Кроме того, для реализации различных циклов управления оборудованием необходима отдельная схема управления с переключением между циклами.

Для отладки и даже небольшого изменения логического управления в релейно-контактной схеме необходим пересмотр всех схем буквально с самого начала, что при ее громоздкости может быть связано с колоссальными затратами времени.

В связи с этим в конце шестидесятых годов прошлого столетия для решения подобных задач стали внедряться программируемые логические контроллеры (ПЛК).

В задачи как релейно-контактных систем управления, так и ПЛК входит автоматизация следующих операций: включение и отключение электродвигателя; выбор направления и скорости вращения; пуск и торможение двигателя; создание временных пауз в движении и защитное отключение двигателя, остановка механизма и т.д.

Таким образом, ПЛК стали разрабатываться и внедряться для исключения части недостатков релейно-контактных систем управления, собранных на дискретных компонентах: реле, таймерах, счетчиках и т.д.

Принципиальное отличие ПЛК от релейно-контактных схем заключается в том, что в нем все алгоритмы управления реализованы программно. При этом надежность работы схемы не зависит от ее сложности. Использование ПЛК позволяет заменить одним логическим устройством любое необходимое количество отдельных элементов релейной автоматики, что увеличивает надежность системы, минимизирует затраты на ее тиражирование, ввод в эксплуатацию и обслуживание. ПЛК может обрабатывать дискретные и аналоговые сигналы, управлять клапанами, сервоприводами, преобразователями частоты и осуществлять регулирование.

Благодаря таким свойствам ПЛК получили широкое распространение в технологических машинах с числовым программным и цикловым управлением.

В России ПЛК производят порядка 41 предприятия, а среди известных иностранных производителей, такие как Simens, Honeywell, Rockwell, Octogon system и др.

В учебном пособии описана работа с ПЛК компании Delta Electronics, для которых написание программ управления и отладки производится с использованием пакета программирования WPLSoft, не требующего больших ресурсов компьютера и являющегося простым инструментом для всех категорий специалистов.

1. ОСНОВНЫЕ ПОНЯТИЯ И ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ ПРОМЫШЛЕННЫХ КОНТРОЛЛЕРОВ

Промышленный контроллер, именуемый также программируемым логическим контроллером (ПЛК), относится к разряду электронных устройств.

Любой ПЛК работает по следующей обобщенной схеме:

Шаг 1. Чтение состояния внешних входных устройств, таких как датчики, клавиатура, переключатели и кнопки.

Шаг 2. Исполнение микропроцессором программы, заложенной в ПЛК пользователем и состоящей из логических связей, различных последовательностей, циклов, таймеров, счетчиков, формул и т.п.

Шаг 3. Формирование выходных сигналов в соответствии с результатами, полученными в ходе обработки заложенной в ПЛК программы. В качестве выходных сигналов выступают открытие или закрытие реле (транзистора), запуск процедуры по управлению технологической установкой или процессом, формирование аналоговой величины или цифрового значения.

ПЛК могут использоваться совместно с различным периферийным оборудованием, например с панелями оператора, управляемым приводом, персональным компьютером и т.п.

В связи с этим контроллеры отвечают строгим требованиям, предъявляемым к промышленным электронным устройствам и касающимся, например, электромагнитной совместимости, способности работать в коммуникационной сети, выполнять специфические задачи.

В системе автоматического управления ПЛК является важнейшим звеном.

1.1. Принципы построения ступенчатых диаграмм

Большинство производителей промышленных контроллеров используют стандартный инструмент программирования – язык ступенчатых диаграмм (англ. Ladder Diagram – LD), который в отечественной практике автоматизации называется еще как релейно-

контактные схемы. Данный язык является универсальным средством отображения схемы автоматического управления.

Ступенчатые диаграммы были изобретены во время Второй мировой войны и первоначально включали лишь базовые компоненты, такие как контакт А (нормально открытый), контакт В (нормально закрытый), выходная катушка, таймер, счетчик и ряд других элементов. Обычный пост управления состоит в основном из данных устройств.

Однако традиционный набор элементов не в состоянии выполнять математические и логические операции, сравнивать и преобразовывать величины, выполнять прикладные инструкции. Необходимость решения данных задач в ходе управления технологическими объектами и привела к созданию нового класса устройств – «промышленные контроллеры».

В связи с тем, что контроллеры исторически создавались в развитие постов управления, к ним по наследству перешел и язык ступенчатых диаграмм. Данный термин теперь применяется именно по отношению к ПЛК.

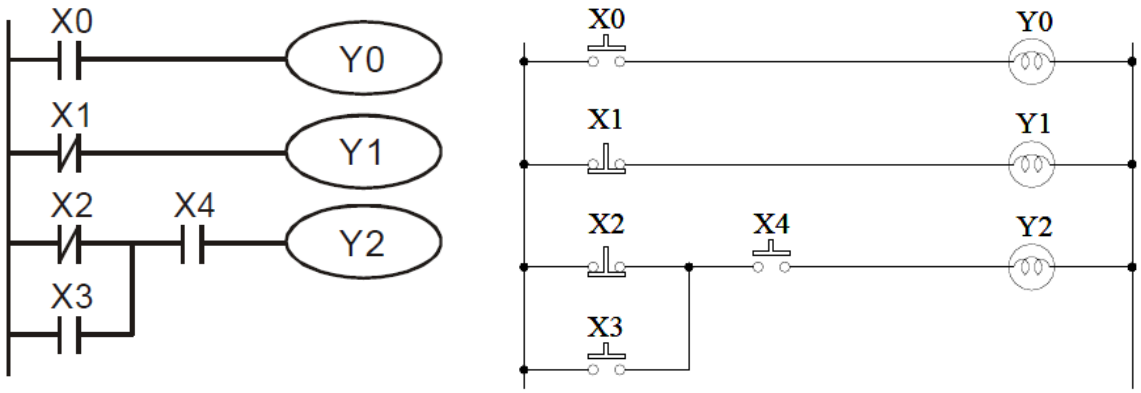
Традиционные ступенчатые диаграммы и разработанные для контроллеров внешне очень похожи, а также построены по тем же принципам. Отличие заключается в том, что при создании традиционной диаграммы вид условных значков на схеме максимально приближен к виду исходного объекта, а значки, используемые для контроллеров, адаптированы для исполнения диаграмм на компьютере или отображения в современном описании на изделие.

Логику формирования ступенчатой диаграммы можно подразделить на два подхода: комбинационная логика и последовательная логика. Рассмотрим отличия между ними.

1. Комбинационная логика.

При данной логике схема состоит из независимых друг от друга фрагментов, положение которых в схеме роли не играет. Входы и выходы различных фрагментов не оказывают друг на друга никакого влияния.

На следующем примере, представленном на рис. 1.1, изображена комбинационная ступенчатая диаграмма в традиционном виде и в форме для контроллеров.



а) *б*)

Рис. 1.1. Ступенчатые диаграммы при комбинационной логике:
а – в форме для ПЛК;
б – традиционная (релейно-контактная) электросхема

Фрагмент № 1 состоит из одного нормально открытого контакта X0, называемого еще как контакт или кнопка типа A, и выхода Y0. Исходным состоянием контакта X0, т.е. когда кнопка не нажата, является ВЫКЛ. Соответственно выход Y0 в исходном состоянии также будет ВЫКЛ. При нажатии кнопки контакт X0 замкнется, выходная катушка перейдет в состояние ВКЛ.

Фрагмент № 2 состоит из одного нормально закрытого контакта X1, называемого еще как контакт или кнопка типа B, и выхода Y1. В отличие от контакта X0 исходным состоянием контакта X1, т.е. когда кнопка не нажата, является ВКЛ. Соответственно выход Y1 в исходном состоянии также будет ВКЛ. Но при нажатии кнопки контакт X1 разомкнется и выходная катушка Y1 перейдет в состояние ВЫКЛ.

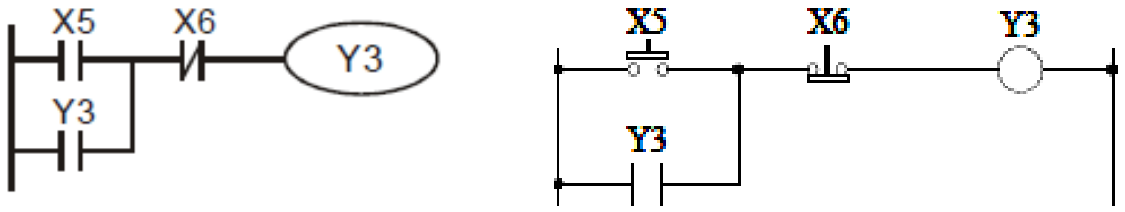
Фрагмент № 3 представляет собой комбинацию входных контактов X2 (нормально закрытый), X3 и X4 (нормально открытые). Выход Y2 перейдет в состояние ВКЛ при условии, что будут замкнуты или X2 (кнопка не нажата) и X4 (кнопка нажата), или X3 и X4 (кнопки нажаты).

2. Последовательная логика.

При данной логике выход одного шага схемы является входным условием для другого. Таким образом, входы и выходы различных фрагментов схемы являются логически связанными и положение каждого фрагмента в схеме строго определено. Такие схемы нельзя «распараллелить».

Для примера рассмотрим частный случай последовательной логики – схему с самоблокировкой выхода. В данной схеме условием включения выхода могут быть несколько параллельных входных контактов, один из которых логически связан с выходом. Когда выход замыкается, связанный с ним входной контакт также замыкается и выход самоблокируется, т.е. его состояние больше не зависит от состояния входных контактов. Таким образом, при одних и тех же входных условиях выход может принимать разные состояния.

Данная схема продемонстрирована на рис. 1.2.



а) *б*)

Рис. 1.2. Ступенчатые диаграммы при последовательной логике:
а – в форме для ПЛК;
б – традиционная (релейно-контактная) электросхема

Когда на схему подается питание первый раз, выход Y3 будет в состоянии ВЫКЛ. Хотя контакт X6 и замкнут, но контакт X5 разомкнут и цепь соответственно тоже.

В данной схеме знаком Y3 обозначается как выход (катушка), так и входной контакт, состояние которого зависит от состояния выхода (катушки). Таким образом, в первоначальный момент связанный контакт Y3 будет также разомкнут, так как катушка Y3 не замкнута.

При замыкании контакта X5 сигнал пройдет через нормально замкнутый контакт X6 и переведет выход (катушку) Y3 в замкнутое состояние, что в свою очередь замкнет связанный входной контакт Y3. Схема перейдет в состояние самоблокировки. Теперь положение контакта X5 больше не оказывает влияния на состояние выхода (катушки) Y3.

Чтобы разомкнуть выход Y3, необходимо нажать кнопку X6, т.е. разомкнуть цепь. Это приведет к размыканию выхода (катушки) Y3 и взаимосвязанного с ней входного контакта Y3. Схема разблокируется. Логические связи и состояния по шагам в данной схеме приведены в табл. 1.1.

Таблица 1.1

Состояние входов и выхода по шагам

Шаг	Состояние		
	X5	X6	Y3
1	неактивен	неактивен	ВЫКЛ.
2	активен	неактивен	ВКЛ.
3	неактивен	неактивен	ВКЛ.
4	неактивен	активен	ВЫКЛ.
5	неактивен	неактивен	ВЫКЛ.

Важным моментом является то, что для нормально открытого и закрытого контактов состояние «активен» является диаметрально противоположным.

У нормально открытого контакта состояние «активен» соответствует замыканию цепи при нажатии кнопки (срабатывании датчика).

Для нормально закрытого контакта состояние «активен» соответствует разрыву цепи при нажатии кнопки (срабатывании датчика).

1.2. Различия в работе традиционных релейно-контактных схем и ступенчатых диаграмм

Несмотря на общность подходов в логике работы релейно-контактных электросхем и ступенчатых диаграмм ПЛК, между ними существует два кардинальных отличия.

Первое и основное отличие заключается в том, что релейно-контактные электросхемы отражают работу реально существующих приборов (счетчиков, таймеров, контактов), а контроллер имитирует их в своем процессоре.

Так как в ПЛК всего один процессор, то контроллер обрабатывает программу последовательно шаг за шагом, что порождает определенную временную задержку. Чем длиннее программа, тем больше это время задержки. Контроллер сначала опрашивает все входы, затем обрабатывает всю программу, и только после этого обновляет все свои физические выходы.

Таким образом, контроллеры работают циклично: опрос входов – отработка программы – установка выходов. Один полный цикл еще называется сканом, а сам процесс – циклическим сканированием.

В традиционных схемах, где все элементы существуют физически, временной задержки не возникает, так как каждый прибор

самостоятельно воспринимает свой входной сигнал, сам его отрабатывает, а затем обновляет свой выход. Таким образом, любые изменения на входах сразу же отрабатываются путем установки выходов в соответствии с логикой релейно-контактной схемы.

Помимо задержки на отработку программы, в контроллерах существует задержка на реакцию входов (цифровой фильтр), а также на время изменения физических выходов, что в совокупности дает еще большую задержку реакции на изменение во внешней среде. Единственным способом уменьшить время реакции является уменьшение времени одного скана, что достигается применением современных быстродействующих процессоров.

Вследствие наличия временной задержки в процессе работы ПЛК может возникнуть ситуация, когда какой-нибудь входной сигнал окажется слишком коротким и попадет в фазу цикла отработки программы, а в фазе опроса входов уже не будет активен. Следовательно, ПЛК просто «не заметит» данный сигнал и это может привести к серьезной аварии. Поэтому при проектировании систем автоматики необходимо выбирать модель контроллера с достаточным быстродействием.

Схематично работа контроллера представлена на рис. 1.3.

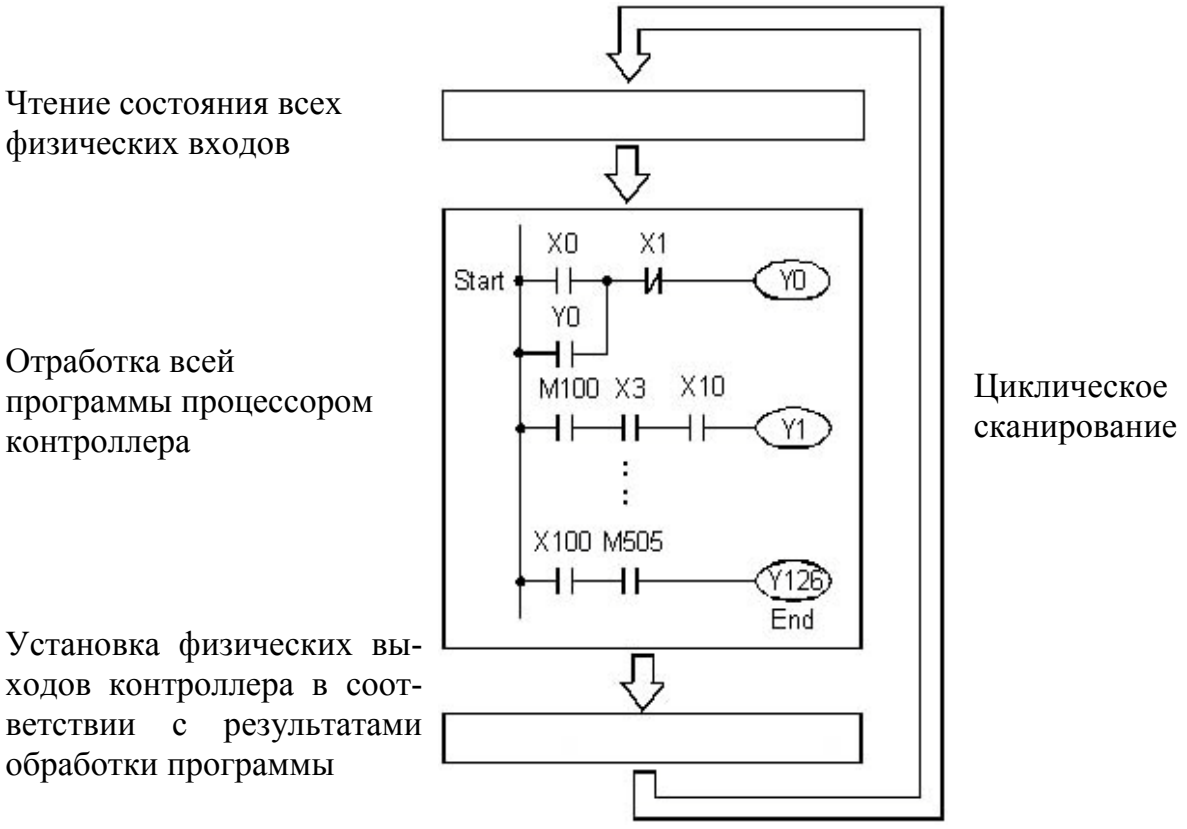
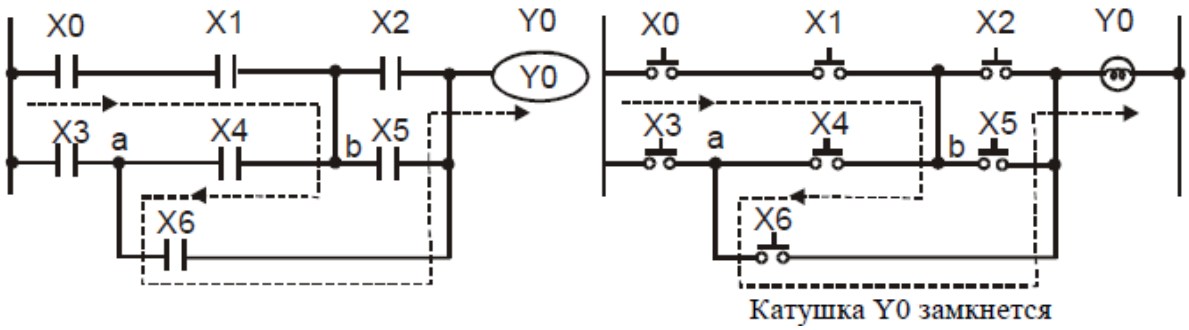


Рис. 1.3. Схема работы контроллера

Второе важное отличие в работе релейно-контактных электросхем от ступенчатых диаграмм ПЛК заключается в том, что контроллер выполняет программу исключительно слева направо и сверху вниз. Следовательно, не допускаются никакие реверсивные направления при выполнении программы (обратное протекание тока по цепи). В обычных схемах допускаются любые направления исполнения схемы (протекание тока по элементам цепи). Данный момент иллюстрируется на рис. 1.4.



а) *б*)

Рис. 1.4. Протекание тока в ступенчатых диаграммах:
а – в форме для ПЛК;
б – традиционная (релейно-контактная) электросхема

Если замкнуть контакты X0, X1, X4 и X6 при остальных разомкнутых, то возникнет «обратное» протекание тока по цепи: контакт X0 – контакт X1 – точка b – контакт X4 – контакт X6 и далее к катушке Y0.

Традиционная схема состоит из физически существующих приборов, поэтому протекание тока обуславливается только законами электротехники и возможно в любых направлениях.

В контроллерах все элементы ступенчатой диаграммы имитируются процессором, который работает циклично только вперед и не может вернуться на шаг назад в рамках одного скана. Поэтому в случае возникновения ситуации, при которой процессору как бы придется «вернуться» назад в программе в представленном примере на рис. 1.4, от точки b к контакту X6, произойдет сбой и будет выдана ошибка при компиляции ступенчатой диаграммы.

1.3. Внутренние объекты программируемых логических контроллеров (операнды)

Как было показано выше, процессор контроллера имитирует функции различных приборов и их работу. Все имитируемые объекты, существующие только внутри процессора и памяти контроллера, называются операндами. Контроллер можно представить как определенный набор не существующих реально контактов, выходных катушек, реле, таймеров, счетчиков и т.д. Однако при программировании их можно использовать как реальные приборы, или точнее сказать, их функции. Каждая модель контроллера имеет свой ограниченный набор операндов, доступных для написания программы.

Любой операнд занимает определенный объем памяти контроллера – бит, байт (8 бит), слово (16 бит) или двойное слово (32 бит). Например, для включения катушки необходимо записать «1» в соответствующую ячейку памяти, а чтобы выключить – записать «0».

Для удобства создания и чтения ступенчатых диаграмм каждый операнд имеет свое стандартное название и обозначение. Данная мера позволяет использовать одну и ту же схему разработчиком системы автоматизации, эксплуатационным инженером, электриком, монтажником, оператором технологической установки. В табл. 1.2 приводится описание всех типов операндов встроенного контроллера преобразователя частоты (ПЧ) Delta VFD-E.

Таблица 1.2

Типы операндов контроллера Delta VFD-E

Тип и обозначение операнда		Описание
1	2	3
Вход	X	Входные реле. Определяют состояние внешних битовых устройств, подключенных к входным клеммам ПЛК. Могут принимать одно из двух состояний: 0 или 1. Нумерация ведется с X0 до X10
Выход	Y	Выходные реле. Определяют состояние выходных клемм ПЛК, к которым подключается нагрузка. В программе могут быть как контактами, так и катушками, и принимать одно из двух состояний: 0 или 1. Нумерация ведется с Y0 до Y4

Окончание табл. 1.2

1	2	3
Внутреннее реле	M	Внутреннее (вспомогательное) реле. Память для двоичных промежуточных результатов. В программе могут быть как контактами, так и катушками, и принимать одно из двух состояний: 0 или 1. Нумерация ведется с M0 до M159 для общих и с M1000 до M1031 – для специальных реле
Таймер	T	Реле времени. В программе могут использоваться для хранения текущего значения таймера, быть контактами и принимать одно из двух состояний: 0 или 1. Нумерация ведется с T0 до T15
Счетчик	C	Используются для реализации счета. В программе могут использоваться для хранения текущего значения счетчика и иметь 16- или 32-битный формат, а также могут быть контактами, и принимать одно из двух состояний: 0 или 1. Нумерация ведется с C0 до C7, с обозначением высокоскоростного счетчика C235
Регистры данных	D	Память данных. 16- или 32-битный формат. Нумерация ведется с D0 до D29 для общих и с D1000 до D1044 – для специальных регистров. В 32-битном формате один регистр занимает две ячейки, например при обращении к D10 данные будут прочитаны из ячеек D10 и D11
Десятичная константа	K	Определение числа в десятичной системе отсчета
Шестнадцатеричная константа	H	Определение числа в шестнадцатеричной системе отсчета

1.3.1. Назначение входных реле (X)

Входные реле X считывают состояния внешних физических устройств (кнопок, переключателей, контактов реле, датчиков и др.), непосредственно подключенных к входным клеммам ПЛК. Каждый входной контакт X может использоваться в программе неограниченное число раз. В ступенчатой диаграмме обозначается как нормально открытый или закрытый контакт.

1.3.2. Назначение выходных реле (Y)

Выходные реле Y управляют состоянием физических выходных контактов ПЛК (релейных или транзисторных), а следователь-

но, и устройствами нагрузки (лампами, катушками реле и др.) непосредственно подключенными к выходным клеммам ПЛК. В качестве нормально открытого или закрытого контакта выход Y может использоваться в программе неограниченное число раз, но выходную катушку Y рекомендуется использовать в программе не более одного раза, так как при повторении катушки Y состояние выхода будет определяться последним Y в скане. Данный пример, в котором состояние выхода Y0 будет определяться только контактом X10, приведен на рис. 1.5.

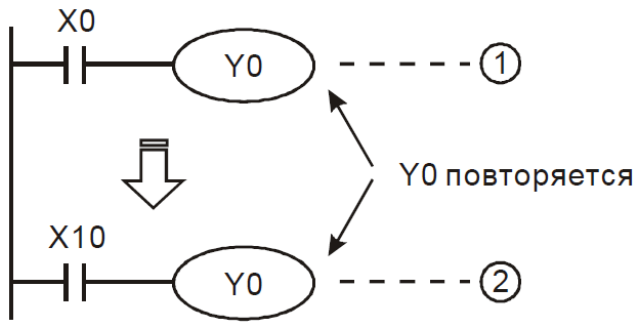


Рис. 1.5. Ступенчатая диаграмма

1.3.3. Типы и назначение внутренних реле (M)

Для запоминания двоичных результатов логических связей (состояний сигналов «0» или «1») внутри программы применяется промежуточная память – внутренние реле, которые соответствуют промежуточным реле в системах управления на релейной логике.

В контроллерах Delta VFD-E используется несколько типов внутренних реле:

1. Общие: не сохраняют свое состояние при отключении питания.

2. Специальные: предоставляют в распоряжение пользователя специальные функции.

В программе внутренние реле могут использоваться как контакты и как выходы. Однако они не могут воспринимать сигналы от внешних устройств (для этого нужно использовать операнды входных реле X) и не могут воздействовать на внешние выходы (для этого нужно использовать операнды выходных реле Y). В программе каждое внутреннее реле может использоваться неограниченное количество раз.

Специальные реле представлены в табл. 1.3.

Таблица 1.3

Специальные реле контроллера Delta VFD-E

Номер	Функция
M1000	Нормально открытый контакт. Контакт замкнут, когда на ПЛК подано напряжение питания, и он находится в состоянии RUN
M1001	Нормально закрытый контакт. Контакт разомкнут, когда на ПЛК подано напряжение питания, и он находится в состоянии RUN
M1002	Контакт замыкается при включении ПЛК во время первого цикла выполнения программы на период, равный периоду сканирования. Все остальное время контакт разомкнут
M1003	Контакт размыкается при включении ПЛК во время первого цикла выполнения программы на период, равный периоду сканирования. Все остальное время контакт замкнут
M1004	Зарезервирован
M1005	Замыкается при возникновении ошибки, сбоя ПЧ
M1006	Выходная частота = 0 Гц
M1007	Направление вращения привода
M1008	Зарезервирован
M1009	Зарезервирован
M1010	Зарезервирован
M1011	Датчик тактов с периодом 10 мс (ON = 5 мс, OFF = 5 мс)
M1012	Датчик тактов с периодом 100 мс (ON = 50 мс, OFF = 50 мс)
M1013	Датчик тактов с периодом 1 с (ON = 0,5 с, OFF = 0,5 с)
M1014	Датчик тактов с периодом 1 мин (ON = 30 с, OFF = 30 с)
M1015	Частота достигнута
M1016	Ошибка чтения/записи параметра
M1017	Запись параметра выполнена успешно
M1018	Разрешение функции высокоскоростного счетчика (когда M1028 = On)
M1019	Зарезервирован
M1020	Флаг нуля (Zero)
M1021	Флаг заимствования (Borrow)
M1022	Флаг переноса (Carry)
M1023	Делитель = 0
M1024	Зарезервирован
M1025	Пуск (ON)/Стоп (OFF) привода
M1026	Изменение направления вращения привода (FWD: OFF, REV: ON)
M1027	Сброс
M1028	Разрешение (ON)/запрещение (OFF) функции высокоскоростного счетчика
M1029	Сброс высокоскоростного счетчика
M1030	Направление счета: вверх (OFF)/вниз (ON)
M1031	Принудительное использование интегрального значения ПИД регулирования из регистра D1019 (ВКЛ.: 1)

1.3.4. Назначение таймеров (T)

Для многих процессов управления необходимы реле времени. В релейной технике для этого применяются реле времени с задержкой на включение или выключение. В ПЛК для этих целей используются внутренние элементы памяти, называемые таймерами, характеристики которых могут определяться программой.

Требуемая установка времени определяется с помощью десятичной константы K, которая указывает количество отсчитываемых шагов времени (дискрет).

Например, для таймера с дискретностью 100 мс, у которого установка времени задана как K100, действительное значение установки будет равно 100 x 100 = 10 000 мс или 10 с.

Таймер работает с задержкой на включение. Он активизируется состоянием входного контакта, равным «1». После отсчета установленного значения времени таймер устанавливает в состояние «1» соответствующий контакт T. Таймер возвращается в отключенное состояние и обнуляет свое текущее значение при установке своего входного контакта в «0».

Схема работы таймера приведена на рис. 1.6.

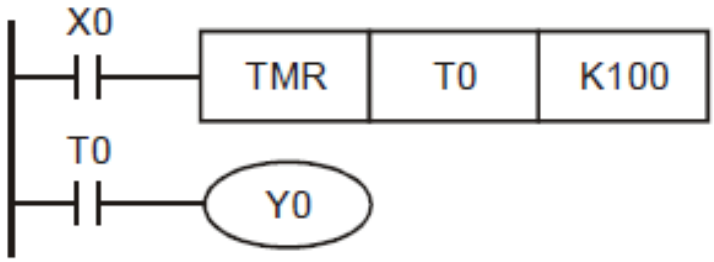


Рис. 1.6. Схема работы таймера

При замыкании контакта X0 начнется отсчет установки таймера T0 (10 с). При достижении заданного значения времени замкнется контакт таймера T0, который в свою очередь замкнет катушку Y0. Как только перестанет выполняться входное условие, контакт T0 разомкнется, размыкая при этом выходной контакт Y0.

Если при отсчете установки таймера контакт X0 хотя бы на мгновение разомкнется, таймер T0 обнулится и при восстановлении входного условия отсчет начнется заново с нуля.

1.3.5. Назначение счетчиков (C)

Когда входной сигнал счетчика изменяет свое состояние с «0» на «1», текущее значение счетчика увеличивается на единицу и, когда оно станет равным задаваемому значению (установке), рабочий контакт счетчика включится.

Схема работы счетчика показана на рис. 1.7.

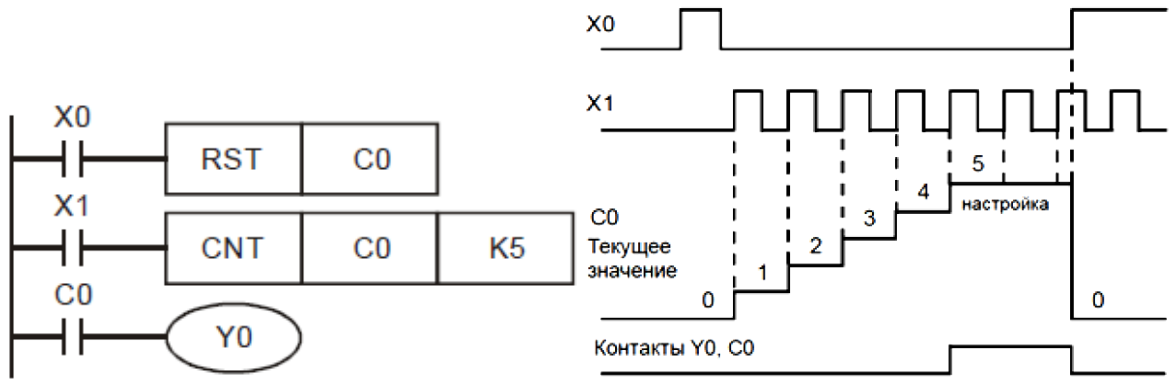


Рис. 1.7. Схема работы счетчика

Когда X0 = 1, происходит сброс счетчика командой RST: текущее значение регистра C0 = 0, контакт C0 разомкнут.

При изменении X1 с 0 на 1 текущее значение регистра C0 будет увеличиваться на 1, происходит прямой счет.

Когда C0 достигнет заданного значения, равного K5, контакт C0 и выходная катушка Y0 замкнутся и последующие импульсы контакта X1 перестанут восприниматься.

1.3.6. Типы и назначение регистров (D)

Регистры представляют собой память для хранения данных внутри ПЛК. В регистры можно записывать числовые значения и следующую друг за другом двоичную информацию.

В контроллере Delta VFD-E имеется 3 типа регистров:

1. Энергонезависимые регистры данных общего назначения (D0–D9).

2. Регистры данных общего назначения: без сохранения данных при отключении напряжения питания (D10–D29).

3. Специальные регистры: для определенных контрольных и проверочных функций (D1000–D1044).

Специальные регистры представлены в табл. 1.4.

Таблица 1.4

Специальные регистры контроллера Delta VFD-E

Номер	Функция
1	2
D1000	Зарезервирован
D1001	Версия программного обеспечения ПЛК
D1002	Объем памяти программы
D1003	Контрольная сумма
D1004–D1009	Зарезервированы
D1010	Текущее время сканирования (ед. = 0,1 мс)
D1011	Минимальное время сканирования (ед. = 0,1 мс)
D1012	Максимальное время сканирования (ед. = 0,1 мс)
D1013	Зарезервированы
D1014	Зарезервированы
D1015	Состояние кнопок пульта управления KPE-LE02: бит 0: MODE; бит 1: STOP; бит 2: RUN; бит 5: UP;бит 6: DOWN; бит 7: ENTER
D1016	Зарезервированы
D1017	Зарезервированы
D1018	Интегральное значение тока
D1019	Принудительно заданное интегральное значение ПИД регулирования
D1020	Выходная частота
D1021	Выходной ток
D1022	Идентификационный номер (ID) платы расширения
D1023	Зарезервированы
D1024	Зарезервированы
D1025	Текущее значение высокоскоростного счетчика C235 (младший байт)
D1026	Текущее значение высокоскоростного счетчика C235 (старший байт)
D1027	Заданная частота ПИД-регулятора
D1028	Значение на аналоговом входе AVI: 0–10 V соответствует 0–1023
D1029	Значение на аналоговом входе ACI: 4–20 mA соответствует 0–1023 или значение на аналоговом входе AVI2: 0–10 V соответствует 0–1023
D1030	Значение потенциометра (V.R) пульта: 0–10 V соответствует 0–1023
D1031	Значение на аналоговом входе AI1 платы расширения: 0-10V или 0–20 mA соответствует 0–4095

Окончание табл. 1.4

1	2
D1032	Значение на аналоговом входе AI2 платы расширения: 0–10 V или 0–20 mA соответствует 0–4095
D1033–D1035	Зарезервированы
D1036	Код ошибки преобразователя частоты
D1037–D1039	Зарезервированы
D1040	Значение на аналоговом выходе AFM
D1041	Значение на аналоговом выходе AO1 платы расширения: при M1010 = 0: 0–10 V или 0–20 mA соответствует (0–65535), при M1010 = 1: 0–10 V или 0–20 mA соответствует (0–10000)
D1042	Значение на аналоговом выходе AO2 платы расширения: при M1010 = 0: 0–10 V или 0–20 mA соответствует (0–65535), при M1010 = 1: 0–10 V или 0–20 mA соответствует (0–10000)
D1043	Определяется пользователем (когда Pr.00.04 = 2, регистр данных будет отображаться на дисплее, как C xxx)
D1044	Режим высокоскоростного счетчика

1.3.7. Константы (K) и (H)

С помощью десятичных (K) и шестнадцатеричных (H) констант можно задавать числовые значения внутри программы ПЛК (например, задаваемые значения установок времени или счета). Числовые значения кодируются внутри ПЛК в двоичном исчислении.

Символ «K» обычно ставится перед числом и обозначает, что число представлено в десятичном формате. Например, K100 означает 100 в десятичном формате.

Символ «H» обычно ставится перед числом и обозначает, что число представлено в шестнадцатеричном формате. Например, H100 означает 100 в шестнадцатеричном формате.

1.4. Условные обозначения элементов ступенчатых диаграмм

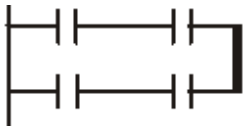
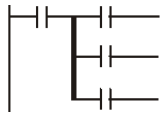
Условные обозначения элементов ступенчатых диаграмм ПЛК представлены в табл. 1.5.

Таблица 1.5

Условные обозначения элементов ступенчатых диаграмм ПЛК

Графический значок	Пояснение	Команда (инструкция)	Операнд
1	2	3	4
	Нормально открытый контакт, тип «А»	LD	X, Y, M, T, C
	Нормально закрытый контакт, тип «В»	LDI	X, Y, M, T, C
	Последовательный нормально открытый контакт (логическое «И»)	AND	X, Y, M, T, C
	Параллельный нормально открытый контакт (логическое «ИЛИ»)	OR	X, Y, M, T, C
	Параллельный нормально закрытый контакт	ORI	X, Y, M, T, C
	Контакт, формирующий импульс по переднему фронту входного сигнала	LDP	X, Y, M, T, C
	Контакт, формирующий импульс по заднему фронту входного сигнала	LDF	X, Y, M, T, C
	Последовательный контакт, формирующий импульс по переднему фронту входного сигнала	ANDP	X, Y, M, T, C
	Последовательный контакт, формирующий импульс по заднему фронту входного сигнала	ANDF	X, Y, M, T, C
	Параллельный контакт, формирующий импульс по переднему фронту входного сигнала	ORP	X, Y, M, T, C
	Параллельный контакт, формирующий импульс по заднему фронту входного сигнала	ORF	X, Y, M, T, C

Окончание табл. 1.5

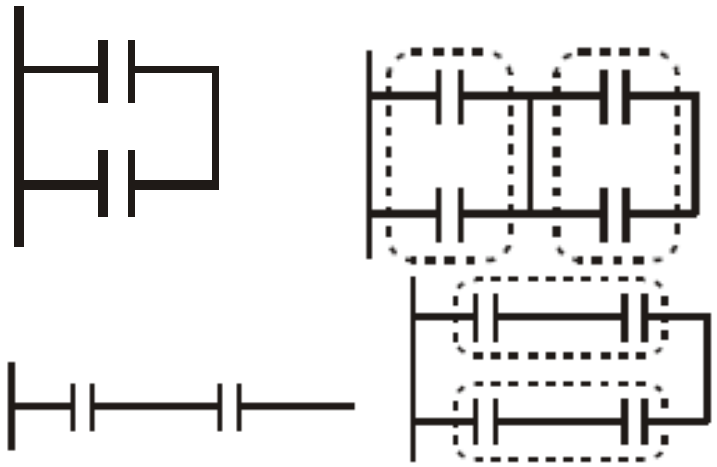
1	2	3	4
	Последовательный блок контактов	ANB	–
	Параллельный блок контактов	ORB	–
	Срабатывание нескольких выходов от одного входа	MPS MRD MPP	–
	Команда для активации выхода (катушки)	OUT	Y, M
	Выбор команды или прикладной инструкции и установка ее параметров	Прикладная инструкция	–
	Логическая инверсия (логическое «НЕ»)	INV	–

Графические значки элементов используются в языке «ступенчатых диаграмм» (англ. LD), а буквенное обозначение команд и инструкций (например, LD, OR, LDF и т.д.) используется в языке «список инструкций» (англ. IL).

Блоком называется такая ступенчатая диаграмма, которая состоит из последовательно или параллельно объединенных групп взаимосвязанных контактов. Каждая группа является как бы единым объектом и внутри блока происходят логические операции именно с этими объектами, а не с отдельными контактами.

В целом блок обрабатывается в следующем порядке: сначала обрабатывается состояние контактов внутри каждой группы, затем формируется выходной сигнал каждой группы, далее происходит логическая операция уже над данными группами, которые выступают в качестве объектов логической операции.

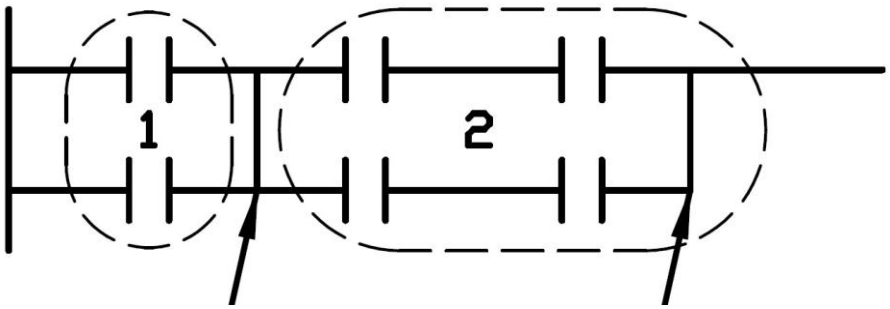
Наиболее распространенными операциями над блоками, представленными на рис. 1.8, являются логическое умножение и сложение.



а) *б*)

Рис. 1.8. Операции над блоками:
а – последовательный блок. Команда ANB (логическое умножение);
б – параллельный блок. Команда ORB (логическое сложение)

Тип блока, параллельный или последовательный, определяется положением соединительной вертикальной линии. Слева от линии – один блок, справа – другой. В зависимости от схемы одна и та же линия может объединять одну группу контактов последовательно, а другую параллельно. Данный пример иллюстрируется рис. 1.9, на котором линия 1 для контактов блока 1 выступает как параллельное объединение, а для блоков 1 и 2 выступает как последовательное объединение. Линия 2 контактов для блока 2 выступает как параллельное объединение.

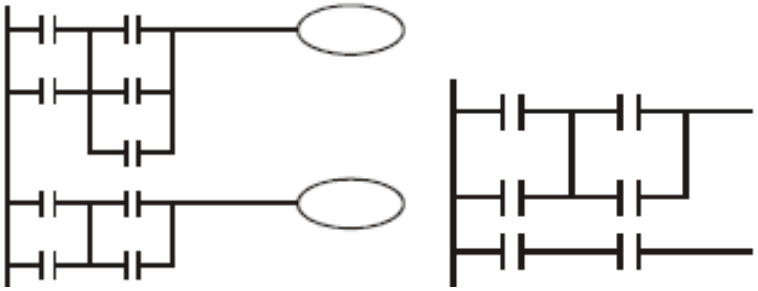


Линия 1 Линия 2

Рис. 1.9. Типы блоков

Контакты и блоки, которые объединены вертикальными и горизонтальными линиями в единую систему формируют одну независимую схему. Как показано на рис. 1.10, программа может содер-

жать много независимых схем. Каждая схема должна заканчиваться каким-либо выходным операндом. В противном случае схема будет незаконченной и при компиляции будет выдана ошибка.



а) *б*)

Рис. 1.10. Виды схем:
а – две независимые схемы; *б* – незаконченные схемы

1.5. Основные правила написания ступенчатых диаграмм

Написание программы необходимо осуществлять по строкам. Закончив одну строку, переходят к следующей строке. Левая вертикальная линия представляет собой общую силовую шину. В традиционных релейно-контактных схемах изображалась и правая силовая шина, т.е. представлялся путь тока через контакты и приборы. В ступенчатых диаграммах контроллеров правая шина не изображается.

Максимально возможное количество контактов в одном ряду – 11. Если количество превышает 11, то контакты с порядковым номером больше 11 автоматически переносятся на новую строку, вначале которой будет стоять номер исходной строки, продолжением которой является новая строка. Рассмотренная ситуация представлена в качестве примера на рис. 1.11. В конечном счете, каждый ряд должен заканчиваться одним из выходных операндов (контактом, катушкой или прикладной инструкцией).

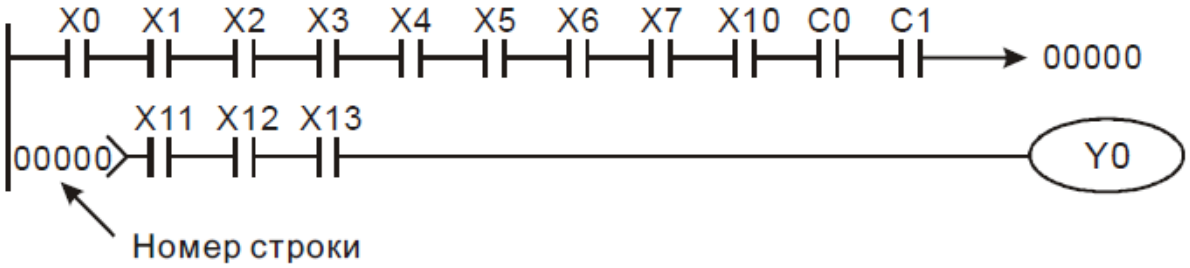


Рис. 1.11. Схема к переносу контактов по строкам

При написании программы необходимо учитывать, что контроллер сканирует программу слева направо сверху вниз и только вперед, т.е. из верхнего левого угла к правому нижнему. Программа обрабатывается по взаимосвязанным контактам одного фрагмента. Когда обработана одна связка контактов, контроллер делает шаг вперед и захватывает следующие контакты и т.д., пока не дойдет в данном фрагменте до выходного операнда, которым должен заканчиваться каждый фрагмент ступенчатой диаграммы. Потом контроллер переходит к следующему фрагменту.

Процесс сканирования программы проиллюстрирован на рис. 1.12.

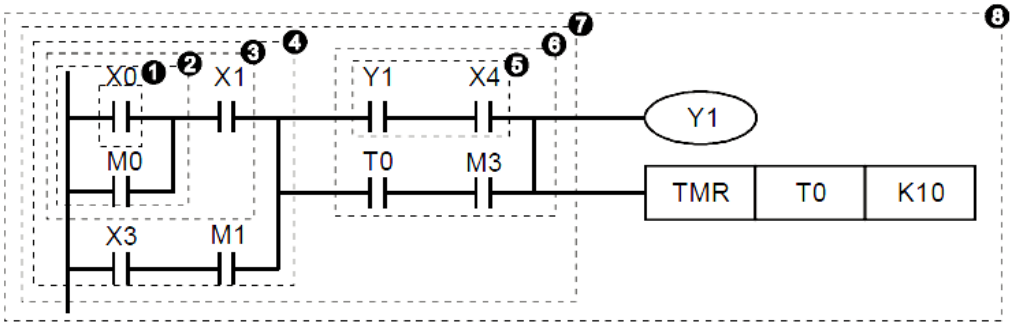


Рис. 1.12. Схема к последовательности сканирования программы

Последовательность отработки контроллером фрагмента программы, представленной на рис. 1.12, отображена в виде табл. 1.6.

Таблица 1.6

Последовательность отработки программы

Номер шага сканирования	Отрабатываемые команды	Контакты на ступенчатой диаграмме
1	2	3
1	LD	X0
2	OR	M0
3	AND	X1
4	LD	X3
–	AND	M1
–	ORB	–
5	LD	Y1
–	AND	X4
6	LD	T0
–	AND	M3
–	ORB	–
7	AND	–
8	OUT	Y1
–	TMR	T0 K10

Как видно из примера, один шаг при сканировании соответствует участку до ближайшего контакта или вертикальной линии. Обработка осуществляется по блокам, начиная с левого. Состояние выходных операндов изменяется в последнем шаге.

1.6. Прикладные инструкции программируемых логических контроллеров

Перечень прикладных инструкций, реализованных в контролере Delta VFD-E, представлен в табл. 1.7.

Таблица 1.7

Перечень прикладных инструкций контролера Delta VFD-E

Классификация	Обозначение	Функция
Передача и сравнение данных	CMP	Сравнение числовых данных
	ZCP	Зонное сравнение числовых данных
	MOV	Пересылка данных
	BMOV	Пересылка блока данных
Операции с данными	ADD	Сложение двух чисел
	SUB	Вычитание двух чисел
	MUL	Умножение двух чисел
	DIV	Деление двух чисел
	INC	Инкрементирование (увеличение на 1)
	DEC	Декрементирование (уменьшение на 1)
Поворот и сдвиг	ROR	Кольцевой сдвиг вправо
	ROL	Кольцевой сдвиг влево
Высокоскоростная обработка	DHSCS	Разрешение высокоскоростного счетчика
Управление Delta VFD	FPID	Управление параметрами ПИД-регулятора ПЧ
	FREQ	Управление выходной частотой ПЧ
	RPR	Чтение параметров
	WPR	Запись параметров

Контрольные вопросы

1. Для чего используются ПЛК?
2. Какова область применения ПЛК?
3. Какова обобщенная схема работы ПЛК?
4. Язык программирования ПЛК – ступенчатые диаграммы. Общие сведения.
5. Поясните принципы построения комбинационной и последовательной логики ступенчатых диаграмм.
6. Назовите основные отличия в работе релейно-контактных схем и ступенчатых диаграмм.
7. Перечислите основные внутренние объекты ПЛК.
8. Поясните назначение дискретных входов и выходов ПЛК.
9. Назовите типы внутренних реле ПЛК и их назначение.
10. Поясните назначение и особенности работы таймеров ПЛК.
11. Поясните назначение и особенности работы счетчиков ПЛК.
12. Назовите типы внутренней памяти ПЛК и ее назначение.
13. Какая ступенчатая диаграмма называется блоком и в какой последовательности он обрабатывается ПЛК?
14. Какова последовательность сканирования программы ПЛК?
15. Назовите основные прикладные инструкции ПЛК.

2. РЕДАКТОР WPLsoft

2.1. Окно редактора

Программа установки создает на рабочем столе изображение (иконку) быстрого запуска редактора WPLSoft (рис. 2.1).

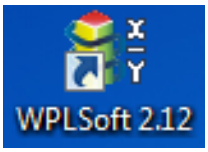


Рис. 2.1. Иконка быстрого запуска WPLSoft

Двойным щелчком мыши по иконке WPLSoft запускается редактор WPLSoft, после чего появится окно редактора, представленное на рис. 2.2.

В верхней части окна расположено главное меню приложения, чуть ниже – панель кнопок быстрого запуска. Серым отмечены неактивные кнопки и пункты меню. Они будут активированы только после создания или открытия проекта.

Рабочая область окна разделена на две части: слева расположено окно коммуникации, справа – пустая серая область, в которой после создания проекта появятся окна редактирования логики проекта.

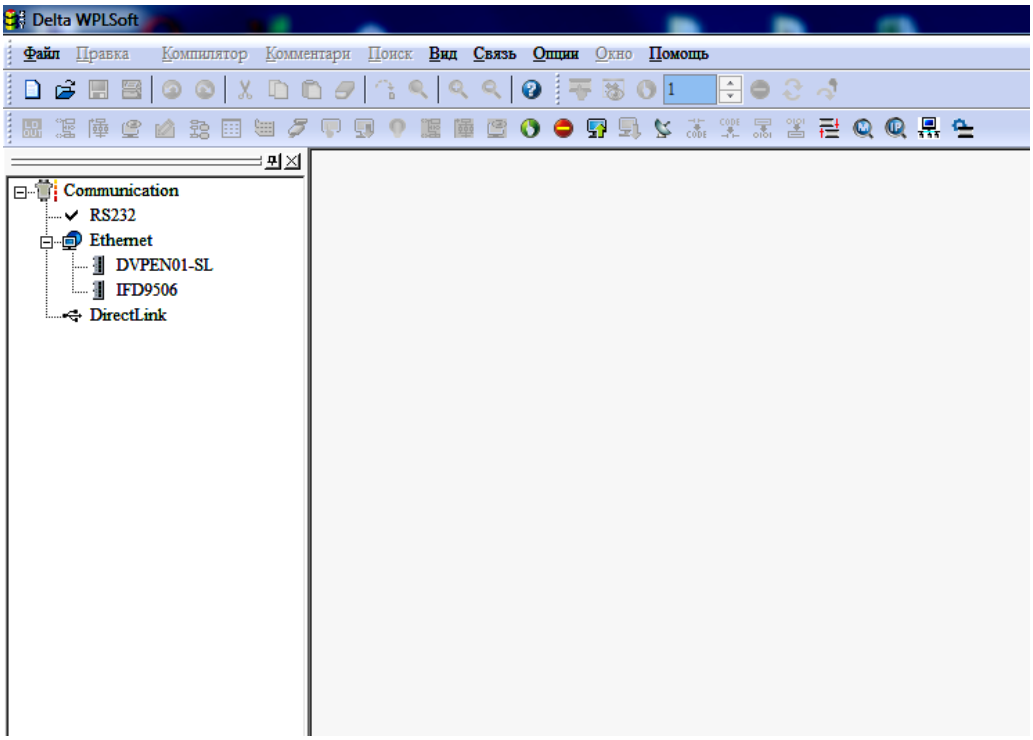


Рис. 2.2. Окно редактора WPLSoft

2.2. Создание нового проекта. Окна редактирования логики проекта

Чтобы создать новый проект, выбирается пункт меню Файл → Новый. Появится окно конфигурирования проекта (рис. 2.3).

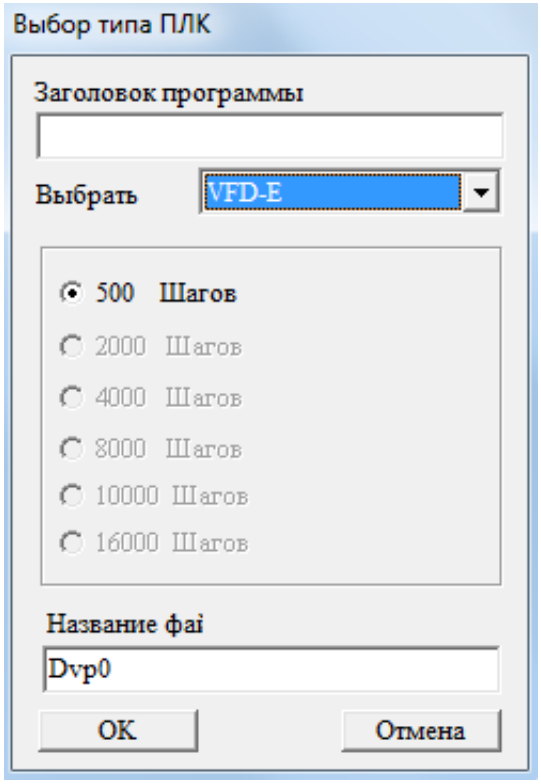


Рис. 2.3. Окно конфигурирования проекта

В верхней части окна можно задать название проекта (Заголовок программы). Чуть ниже расположена конфигурация ПЛК (выбирается VFD-E). В самом низу – область ввода имени файла проекта.

При нажатии кнопки ОК появятся два окна, каждое из которых по-своему отражает логику созданного проекта:

- режим списка инструкций (рис. 2.4) – отображает логику проекта в виде последовательности инструкций контроллера;
- режим ступенчатых диаграмм – отображает логику проекта в виде ступенчатых диаграмм. Примеры, рассмотренные в данном учебном пособии, будут использовать именно это окно.

Нижняя часть панели кнопок быстрого запуска изменяется в зависимости от выбранного окна отображения. Эта часть панели содержит кнопки редактирования логики проекта, которые будут рассмотрены на соответствующих этапах создания логики демонстрационного проекта.

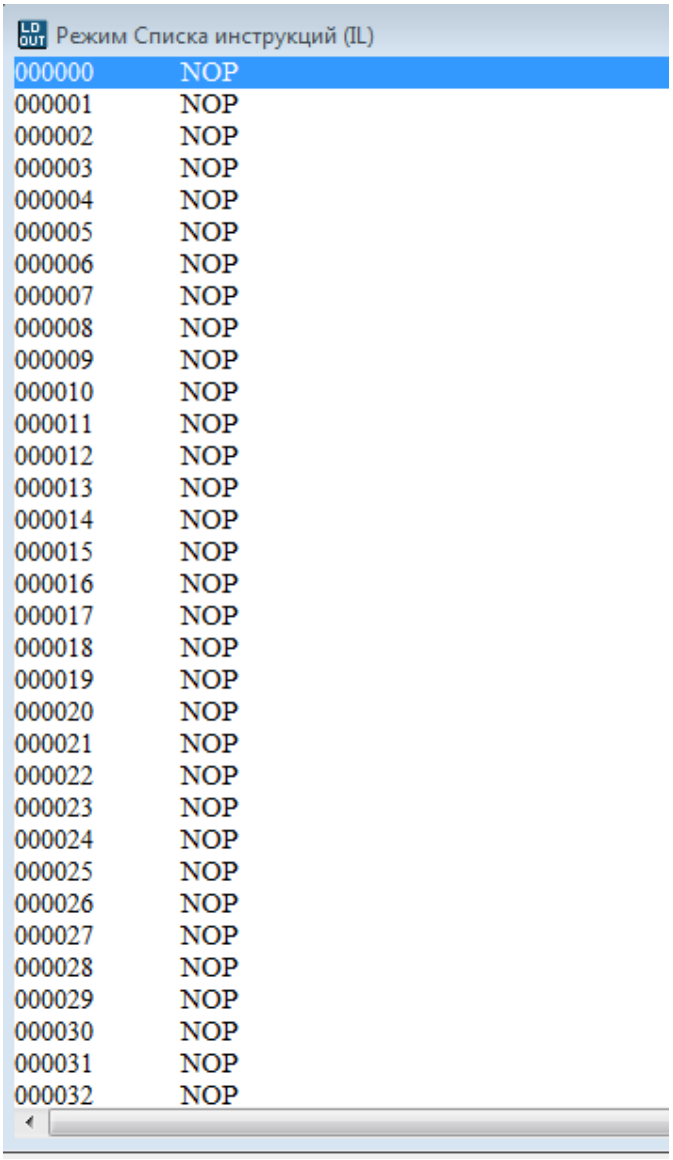


Рис. 2.4. Окно режима списка инструкций

Любое выбранное отображение логики может быть преобразовано в набор инструкций. В свою очередь, набор инструкций может быть преобразован в любой другой тип отображения. Все эти операции осуществляются вызовом команд меню Компилятор.

2.3. Сохранение проекта

Чтобы сохранить созданный проект, выбирается пункт меню Файл → Сохранить, после чего появится стандартное диалоговое окно сохранения файла системы Windows. Задается имя файла, которое будет содержать фамилию и номер группы. Нажатием кнопки Сохранить проект будет сохранен в указанном месте.

2.4. Закрытие проекта. Выход из редактора

Чтобы закрыть проект, выбирается пункт меню Файл → Закрыть проект.

Чтобы выйти из приложения, выбирается пункт меню Файл → Выход или нажимаются кнопки «Alt+X» с клавиатуры.

Контрольные вопросы

1. Каково назначение редактора WPLsoft?
2. Назовите части рабочей области окна редактора WPLsoft и их назначение.
3. Назовите режимы отображения логики проекта редактора WPLsoft.

3. ДЕМОНСТРАЦИОННЫЙ ПРОЕКТ

Для знакомства с основными инструментами редактирования логики, предоставляемыми редактором WPLSoft, рассмотрено создание простого демонстрационного проекта.

Этапы создания демонстрационного проекта можно в дальнейшем использовать в качестве краткого руководства по реализации того или иного элемента разрабатываемой логики.

3.1. Открытие созданного ранее проекта

Для открытия созданного ранее проекта выбирается пункт меню Файл → Открыть. Появится стандартное диалоговое окно открытия файла системы Windows. В этом окне выбирается имя файла проекта, сохраненного на одном из предыдущих этапов и нажимается кнопка Открыть.

Как и в случае создания нового проекта, после открытия сохраненного проекта неактивные кнопки быстрого запуска и пункты меню станут активными, а в правой части рабочей области появятся окна редактирования логики проекта.

В том случае, когда в проекте не создавалось никакой логики, все окна, кроме окна редактирования набора инструкций, будут пустыми. По умолчанию вся память ПЛК заполнена командами NOP (no operation) – нет операции.

3.2. Создание простой логики

Рассмотрим на примере. Пусть входы X0 и X1 должны зажигать выход Y0, а вход X2 управлять работой схемы: если вход X2 включен, выход Y0 всегда выключен. Такая логика описывается следующей простейшей формулой:

$$Y0 = (X0 + X1) \cdot \overline{X2}, \qquad (3.1)$$

где Y0 – выход; X0...X2 – входы.

Для реализации этой логики на языке ступенчатых диаграмм активируется окно редактирования ступенчатых диаграмм и выбирается пункт меню Вид → Ступенчатая диаграмма.

Все ступенчатые диаграммы «растут» слева направо. Левая граница ступенчатой диаграммы отмечена вертикальной чертой.

Окно редактирования ступенчатых диаграмм разбито на клетки, в каждой из таких клеток может быть размещено не более одного элемента ступенчатой диаграммы. Синий квадрат представляет собой область ввода. Перемещать область ввода можно с клавиатуры клавишами «вверх», «вниз», «влево», «вправо» или мышкой, щелкая по интересующей части ступенчатой диаграммы.

Область ввода устанавливается в левом верхнем углу ступенчатой диаграммы, как показано на рис. 3.1.

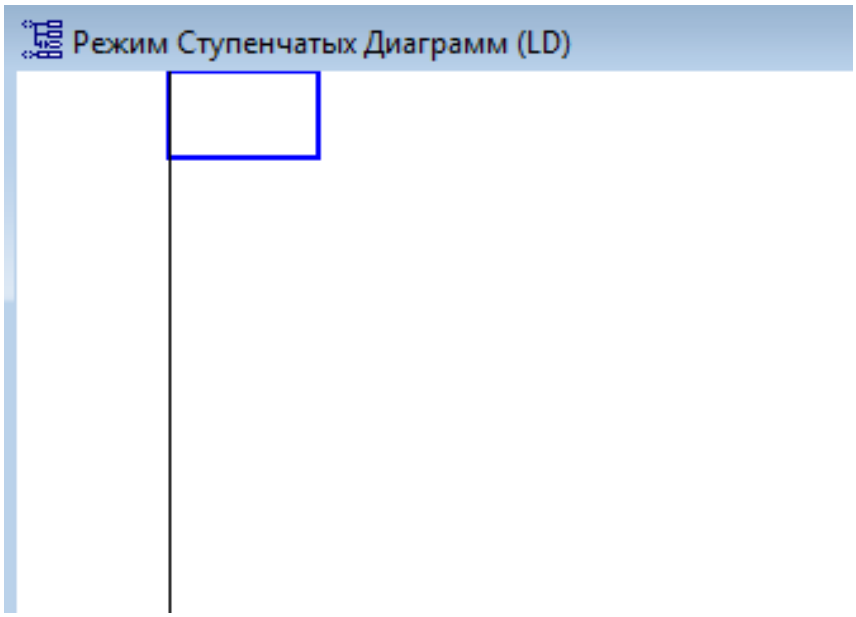


Рис. 3.1. Исходное положение области ввода

Выбор команд редактирования осуществляется нажатием на соответствующие кнопки панели быстрого запуска. Кроме того, все команды редактирования имеют горячие клавиши (они изображены на нижней части кнопок редактирования).

Входы и внутренние реле на ступенчатой диаграмме представлены в виде контактов различного типа. Всего существует четыре типа контактов, каждому из которых соответствует кнопка панели быстрого запуска:

- Нормально открытый контакт, тип A.
- Нормально закрытый контакт, тип B.
- Контакт с включением по переднему фронту сигнала.
- Контакт с включением по заднему фронту сигнала.

На панели быстрого запуска нажимается кнопка «F1» клавиатуры. Появится окно ввода параметров операнда (рис. 3.2).

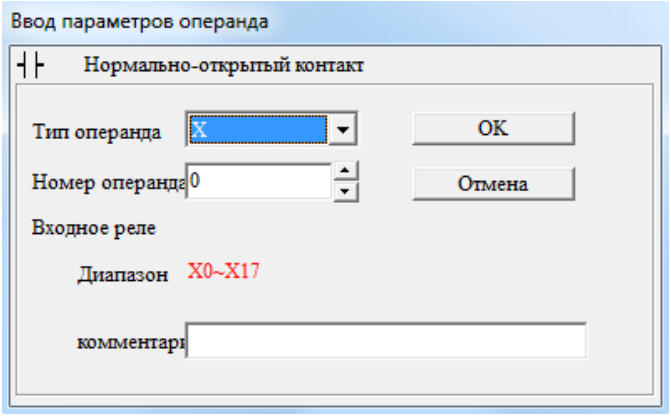


Рис. 3.2. Окно ввода параметров операнда

В левом верхнем углу окна указан тип редактируемого операнда – нормально открытый контакт.

В окне редактирования операнда указываются тип и номер операнда.

Диапазон доступных номеров операнда обозначается красными буквами рядом с надписью «Диапазон».

Поле «Комментарии» позволяет задать комментарий к операнду. Комментарии сделают ступенчатую диаграмму более понятной и простой для восприятия.

Задается имя операнда X, номер – 0, в поле комментария вводится «Вход 0». После этого нажимается кнопка OK.

Смещается поле ввода вниз и все повторяется для операнда X1. В поле комментария вводится «Вход 1».

Результат действий приведен на рис. 3.3.

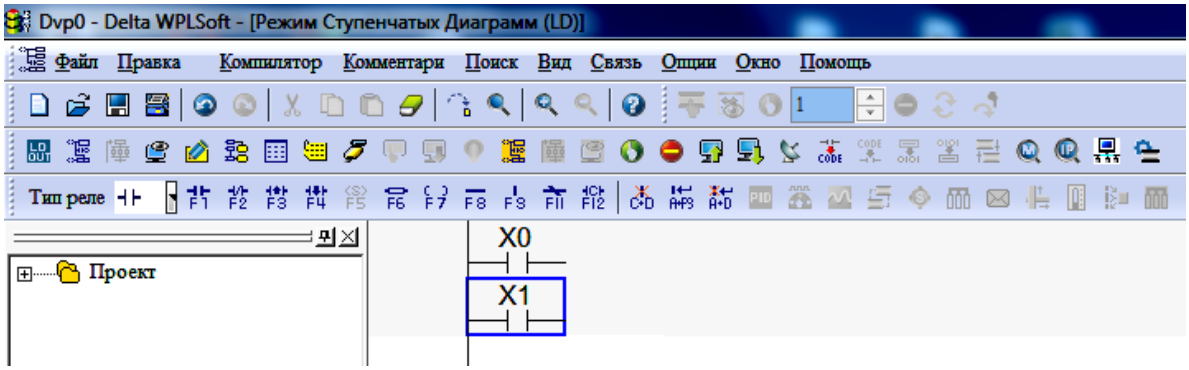


Рис. 3.3. Входы 0 и 1 на ступенчатой диаграмме

Так как оба входа 0 и 1 должны зажигать выход 0, то объединяется X0 с X1. Помещается поле ввода справа от X0 и нажимается кнопка «F9» с клавиатуры, после чего появляется вертикальная черта, соединяющая X0 с X1.

Выходы на ступенчатых диаграммах отображаются в виде катушек. В отличие от контактов катушка на ступенчатой диаграмме всегда находится в крайнем правом положении. Катушка устанавливает состояние соответствующего ей устройства как результат работы участка диаграммы (последовательности контактов) слева от нее. Помимо выходов (Y) катушками могут быть внутренние реле (M), что позволяет сохранять результаты работы участков диаграммы и использовать их на следующих циклах работы ПЛК.

На диаграмме добавляется выход 0. Для этого нажимается кнопка «F7» с клавиатуры, после чего появляется окно ввода параметров операнда (см. рис. 3.2). В левом верхнем углу, как обычно, указан тип редактируемого операнда – выходная катушка. Задается тип операнда – Y, номер операнда – 0, комментарий – «Выход 0» и нажимается ОК.

Катушка появится не в выбранной области ввода, а в крайнем правом положении на ступенчатой диаграмме. Результат действий приведен на рис. 3.4.

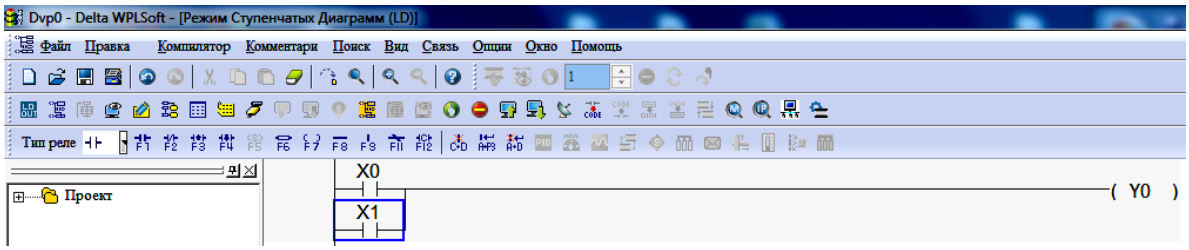


Рис. 3.4. Входы 0, 1 и выход 0 на ступенчатой диаграмме

Добавляется вход 2. В соответствии с формулой (3.1) вход 2 должен быть нормально закрытым реле. Помещается область ввода справа от X0 и нажимается кнопка «F2» с клавиатуры. Появится окно ввода параметров операнда (см. рис. 3.2), типом устройства будет нормально закрытый контакт. Вводится имя и номер операнда (X2) и задается комментарий («Вход 2»). Результат действий приведен на рис. 3.5.

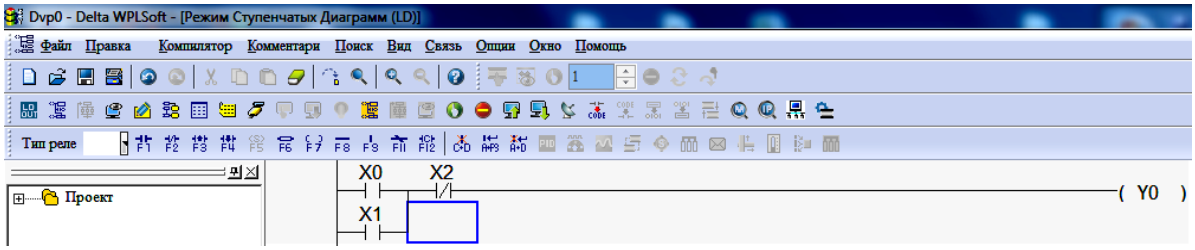


Рис. 3.5. Входы 0...2, выход 0 на ступенчатой диаграмме

3.3. Работа с таймерами и счетчиками

Таймеры являются более сложными операндами, чем катушки и контакты. Таймеры служат для введения задержки распространения сигналов. Длительность задержки устанавливается в периодах работы ПЛК. У каждого таймера есть вход и выход. Вход таймера управляет работой таймера, а выход отражает его текущее состояние.

При наличии положительного сигнала на входе таймера таймер начинает работать. Как только сигнал пропадает, таймер сбрасывается. По достижении таймером установленного значения задержки включается выход таймера.

В ступенчатых диаграммах работой таймеров управляют специальные инструкции. Прикладные инструкции могут управлять не только таймерами, во многом они похожи на катушки: они так же располагаются в крайнем правом положении на диаграмме и используют в качестве условия активации результат работы последовательности контактов слева. При этом операции, выполняемые прикладными инструкциями, значительно более сложные, чем простое включение или выключение некоторого устройства, к тому же инструкции могут принимать конфигурирующие их работу параметры.

Добавляется задержка на включение выхода 0. Для этого помещается область ввода справа от контакта X2 и нажимается кнопка «F6» с клавиатуры. После этого появляется окно редактирования прикладных инструкций (рис. 3.6).

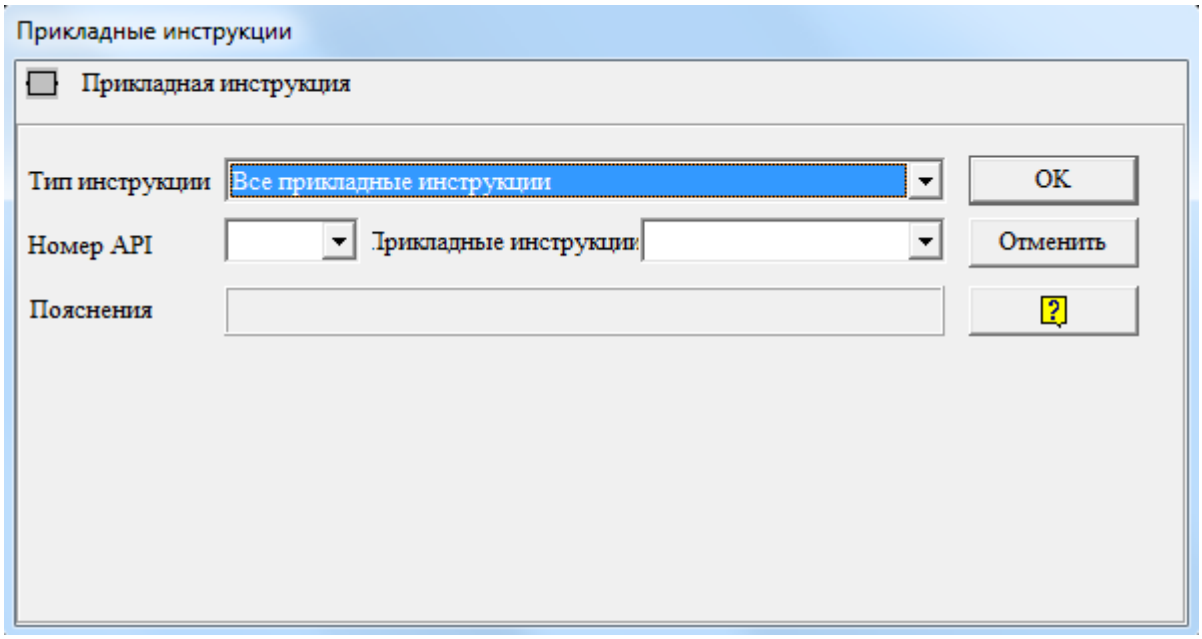


Рис. 3.6. Окно редактирования прикладных инструкций

В списке «Тип инструкции» выбирается «Таймеры & Счетчики». В списке «Прикладные инструкции» выбирается «TMR». Окно приобретет вид, представленный на рис. 3.7, в котором появятся параметры S и D, а также справка внизу окна.

Справка состоит из двух окон. В верхнем окне звездочками отмечены допустимые для каждого из параметров типы операндов. В нижнем окне размещена справочная информация по каждому из параметров: S – номер таймера, D – значение задержки таймера.

Выбирается из списка параметра S значение T, затем в появившемся числовом поле ввода задается номер таймера – 0. Затем аналогично для параметра D – тип операнда K (десятичная константа), значение 100 (задержка на 10 с) и нажимается OK.

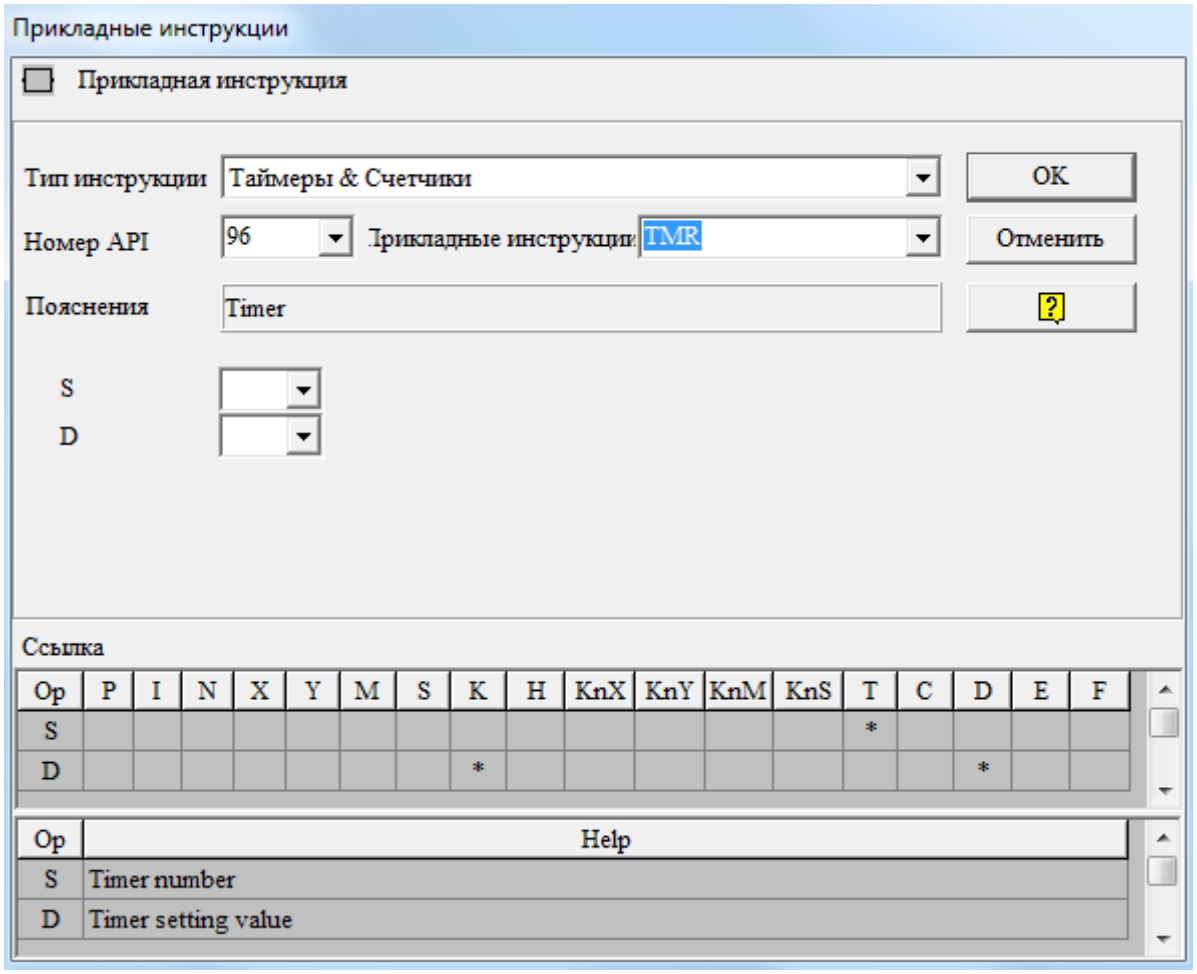


Рис. 3.7. Окно редактирования инструкции после выбора ее типа

Катушка Y0 заменяется символом таймера 0. В этом символе указано слева направо: имя прикладной инструкции, конфигурирующей таймер (TMR), имя таймера (T0) и величина задержки таймера (K100 – константа 100).

Теперь, чтобы таймер 0 включал выход 0, необходимо добавить на диаграмму выход таймера и выход 0. Выход таймера добавляется в виде нормально открытого контакта, а выход 0, как и прежде, в виде катушки. Редактор WPLSoft запомнил комментарий к устройству Y0 («Выход 0»). Результат выполненной работы представлен на рис. 3.8.

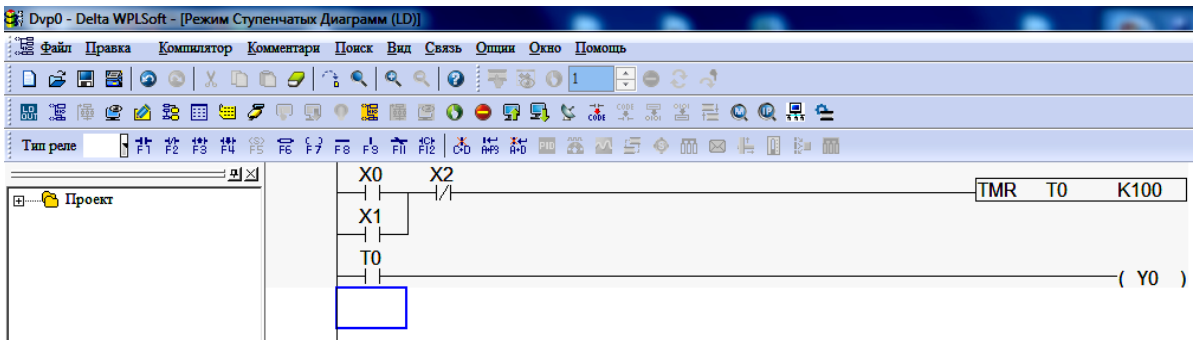


Рис. 3.8. Результат добавления таймера 0

Счетчики предназначены для подсчета импульсов. Так же как и таймеры, счетчики имеют собственный вход и выход. Счет импульсов осуществляется по входу счетчика. Как только значение счетчика достигает заданного значения, счетчик устанавливает свой выход в активное состояние.

Счетчик импульсов добавляется по входам X0 и X1. Для этого устанавливается область ввода справа от X1 и нажимается кнопка «F6» с клавиатуры. В появившемся окне редактирования прикладных инструкций (см. рис. 3.6) выбирается в списке «Тип инструкции» – «Таймеры & Счетчики». В списке «Прикладные инструкции» выбирается «CNT». Окно изменится подобно тому, как это было при добавлении таймера. Задаются следующие параметры: S – C1, D – K5.

Выход счетчика 1 отправляется также на выход 0. Для этого добавляются нормально открытый контакт C1 и катушка Y0. Задается комментарий к контакту C1 – «Счетчик импульсов».

Сброс счетчика осуществляется прикладной инструкцией RST.

Устанавливается сброс счетчика 1 по включению выхода таймера 0. Для этого устанавливается область ввода справа от контакта T0 и нажимается кнопка «F6» с клавиатуры. В появившемся окне редактирования прикладных инструкций (см. рис. 3.6) выбирается в списке «Тип инструкции» – «Выход». В списке «Прикладные инструкции» выбирается «RST». Единственный появившийся параметр S устанавливается равным C1 и нажимается OK.

После этого добавляется комментарий к контакту T0. Для этого щелкается дважды мышью по контакту T0 или, установив на него область ввода, нажимается клавиша «Enter» с клавиатуры. В появившемся окне «Инструкция ввода» нажимается кнопка «…» для вызова окна ввода параметров операнда (см. рис. 3.2). Вводится комментарий «Сброс счетчика импульсов» и нажимается OK.

Результат добавления счетчика представлен на рис. 3.9.

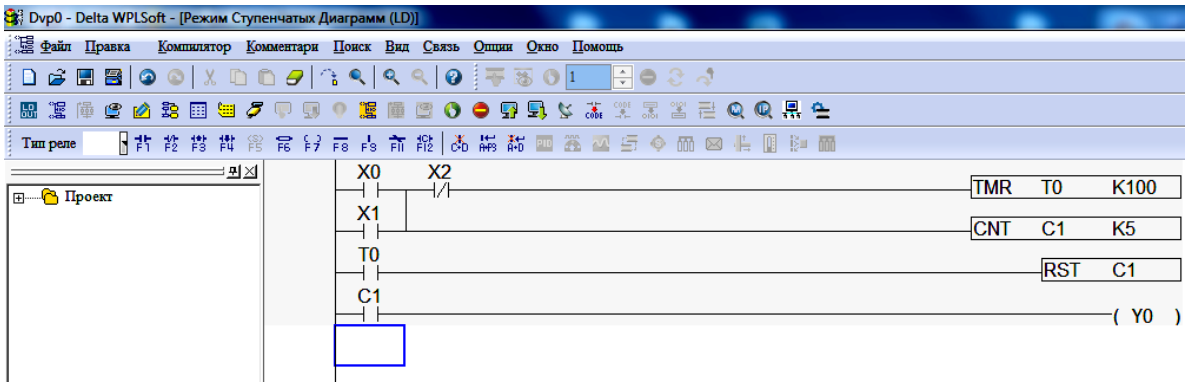


Рис. 3.9. Результат добавления счетчика 1

3.4. Добавление RS-триггера

Прикладные инструкции SET и RST позволяют создавать RS-триггер. Инструкция SET устанавливает операнд в активное состояние, а RST – сбрасывает.

Пусть импульс по входу 3 включает выход 3, а импульс по входу 4 – его сбрасывает.

Устанавливается область ввода под контактом C1 и нажимается кнопка «F3» с клавиатуры. Появляется окно ввода параметров операнда (см. рис. 3.2). Тип операнда – контакт с включением по переднему фронту сигнала. Вводится тип операнда – X, номер – 3, комментарий – «S». Смещается область ввода вниз и добавляется аналогичный контакт X4 с комментарием «R».

Затем помещается область ввода справа от X3 и нажимается кнопка «F6» с клавиатуры. В появившемся окне редактирования прикладных инструкций (см. рис. 3.6) выбирается в списке «Тип инструкции» – «Выход». В списке «Прикладные инструкции» – «SET». Устанавливается параметр S равным Y3 и нажимается OK.

После этого область ввода помещается справа от X4 и добавляется прикладная инструкция RST, параметр S устанавливается равным Y3. Результаты редактирования приведены на рис. 3.10.

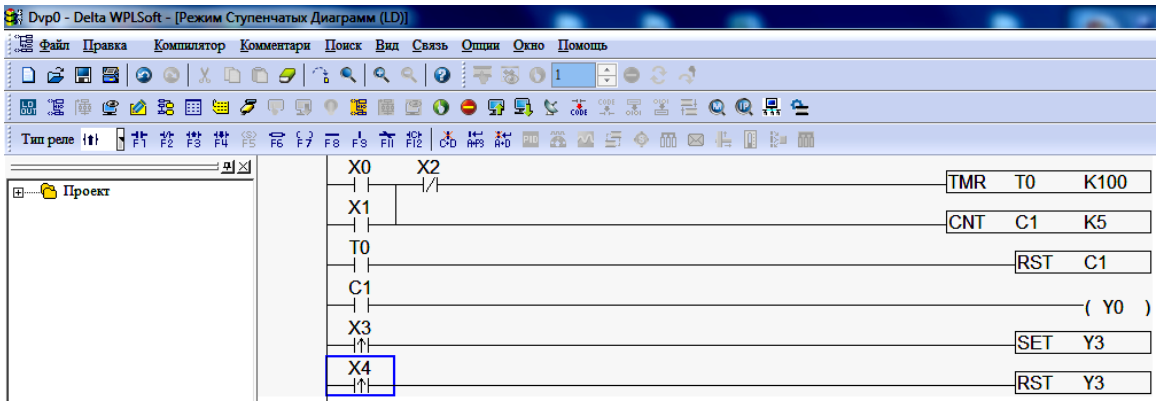


Рис. 3.10. Результат добавления RS-триггера на выход 3

3.5. Импульсные выходы. Добавление пустых строк

Допустим, необходимо генерировать один короткий импульс на выход 1 при наличии высокого уровня на входах 0 или 1 (например, при нажатии на кнопку). Для этих целей можно использовать импульсные прикладные инструкции:

- PLS – генерирует один импульс в целевом устройстве (внутреннее реле или выход) при появлении высокого уровня на входе.
- PLF – генерирует один импульс в целевом устройстве (внутреннее реле или выход) при появлении низкого уровня на входе.

Прикладные инструкции PLS/PLF добавляются аналогично прикладным инструкциям SET/RST, разница заключается только в выборе параметра «Прикладные инструкции» – его необходимо задать в соответствии с требующейся инструкцией (либо PLS, либо PLF).

Устанавливается область ввода справа от X1, нажимается кнопка «F6» с клавиатуры и добавляется инструкция PLS. Параметр D устанавливается равным Y1.

Считать импульсы счетчик 1 теперь может по выходу 1, причем делать это он должен до инструкции сброса, иначе в случае короткого импульса на входах 0 или 1 сброс счетчика может не сработать. Для этого необходимо вставить пустую строку перед инструкцией сброса.

Устанавливается поле ввода на контакт T0 и выбирается пункт меню Правка → Вставить строку. Появится необходимая пустая строка. Затем добавляется нормально открытый контакт Y1 и прикладная инструкция конфигурации счетчика C1. Задается комментарий «Импульсный выход 1» для Y1, счетчик конфигурируется, как и прежде, на 5 импульсов.

Добавляется еще один импульсный выход, который будет генерировать короткий импульс при срабатывании таймера 0.

Предложено установить таким выходом – выход 2. Для этого добавляется пустая строка выше контакта C1, затем устанавливается поле ввода справа от контакта T0 и нажимается кнопка «F9» с клавиатуры, чтобы добавить вертикальную черту. Затем перемещается поле ввода на одно деление вниз и добавляется прикладная инструкция PLS для выхода 2.

Результат добавления импульсных выходов представлен на рис. 3.11.

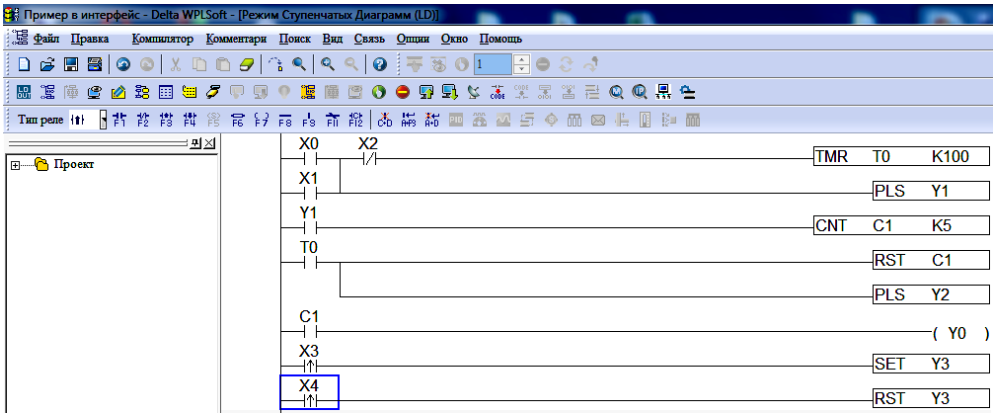


Рис. 3.11. Результат добавления импульсных выходов

3.6. Отключаемые блоки на ступенчатой диаграмме

Включение или выключение отключаемых блоков ступенчатой диаграммы осуществляется прикладными инструкциями MC и MCR соответственно. При этом границы отключаемого блока задают сами инструкции: MC начинает блок, а MCR его заканчивает.

Инструкция MC принимает на входе условие включения блока. Если уровень сигнала на входе высокий, блок будет включен, если низкий – выключен.

Установим выключаемым блок ступенчатой диаграммы, отвечающий за обработку входов 1 и 2. Функционально он никак не связан с последующей частью диаграммы, поэтому его отключение будет вполне корректным. Работой отключаемого блока будет управлять вход 0.

Добавляются пустые строки над контактами X0 и X3 (на границах отключаемого блока). Перемещается область ввода на контакт X0 и нажимается кнопка «Delete» клавиатуры. Контакт будет удален

с диаграммы. Затем снова добавляется нормально открытый контакт X0, но уже в верхнюю пустую строку. Справа от него добавляется инструкция MC. Для этого нажимается кнопка «F6» с клавиатуры, в появившемся окне редактирования прикладных инструкций выбирается из списка прикладная инструкция MC. Единственный параметр N (номер слоя) устанавливается равным 0 и нажимается OK. Аналогичным образом добавляется инструкция MCR в нижнюю пустую строку.

Теперь, для того чтобы активировать блок, обрабатывающий входы 1 и 2, необходимо подать высокий уровень сигнала на вход 0. Это может оказаться не очень удобным, так как блок будет нормально выключенным. Чтобы сделать его нормально включенным, необходимо заменить нормально открытый контакт X0 на нормально закрытый.

Перемещается область ввода на контакт X0, и в нижнем левом углу панели быстрого запуска из списка типов контактов (рис. 3.12) выбирается нормально закрытый контакт.

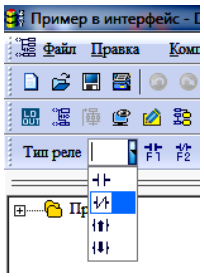


Рис. 3.12. Список типов контактов (реле)

Итоговая ступенчатая диаграмма представлена на рис. 3.13.

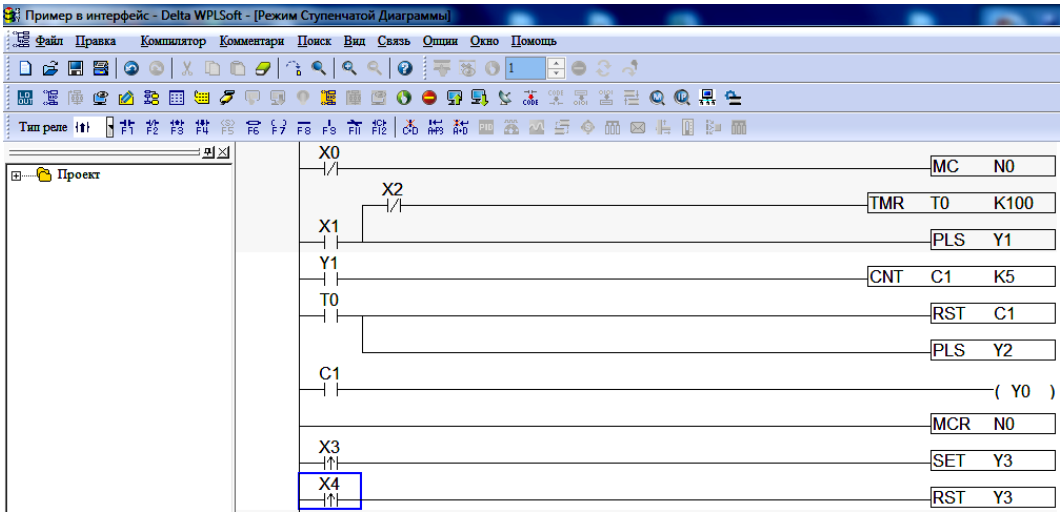


Рис. 3.13. Добавление отключаемого блока

При отключении блока ступенчатой диаграммы вся входящая в блок логика будет игнорироваться ПЛК. Это приведет к тому, что состояние всех операндов, входящих в блок, в частности выходов, будет оставаться неизменным до нового включения блока. В связи с этим не следует ожидать, что при выключении выделенного блока выходы 0, 1 и 2 погаснут, напротив, выходы сохранят свое состояние до включения блока.

3.7. Инверсия. Логический элемент ИЛИ-НЕ

Создадим логический элемент ИЛИ-НЕ для входов 5,6 и выхода 4. Для этого добавляется на диаграмму два нормально разомкнутых контакта X5 и X6 один под другим. Задаются комментарии «A» и «B», соответственно. Затем помещается область ввода справа от контакта X5 и нажимается кнопка «F9» с клавиатуры. Контакты X5 и X6 соединятся вертикальной чертой. Затем, не перемещая область ввода, добавляется катушка Y4.

Таким образом, создается элемент ИЛИ, остается добавить инверсию. Для этого нажимается кнопка «F11» с клавиатуры. Косая черта означает инверсию участка ступенчатой диаграммы, находящейся левее черты. Дважды щелкнув по катушке Y4, задается комментарий к устройству – «NOR».

Ступенчатая диаграмма элемента ИЛИ-НЕ представлена на рис. 3.14.



Рис. 3.14. Ступенчатая диаграмма элемента ИЛИ-НЕ

3.8. Кнопки редактирования вертикальных соединений

Редактирование вертикальных соединений подразумевает выполнение двух операций: создание соединений и их удаление. При выполнении обеих этих операций область ввода должна располагаться над соединением, как показано на рис. 3.15.

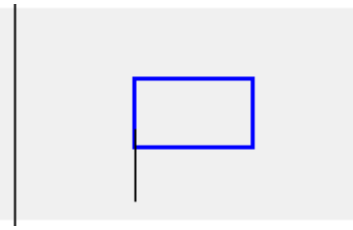


Рис. 3.15. Положение области ввода при редактировании вертикальных соединений

Ранее неоднократно использовалась кнопка для создания вертикальных соединений. Для удаления таких соединений используется сочетание клавиш «Ctrl+D» с клавиатуры. При этом вертикальные соединения в отличие от горизонтальных не удаляются кнопкой «Delete».

Следующие комбинации клавиш удобно использовать для последовательного редактирования нескольких вертикальных соединений:

- Комбинация «Alt+F9» с клавиатуры создает вертикальное соединение и смещает область ввода на одно деление вниз.
- Комбинация «Alt+D» с клавиатуры удаляет вертикальное соединение и смещает область ввода на одно деление вниз.

3.9. Отображение комментариев

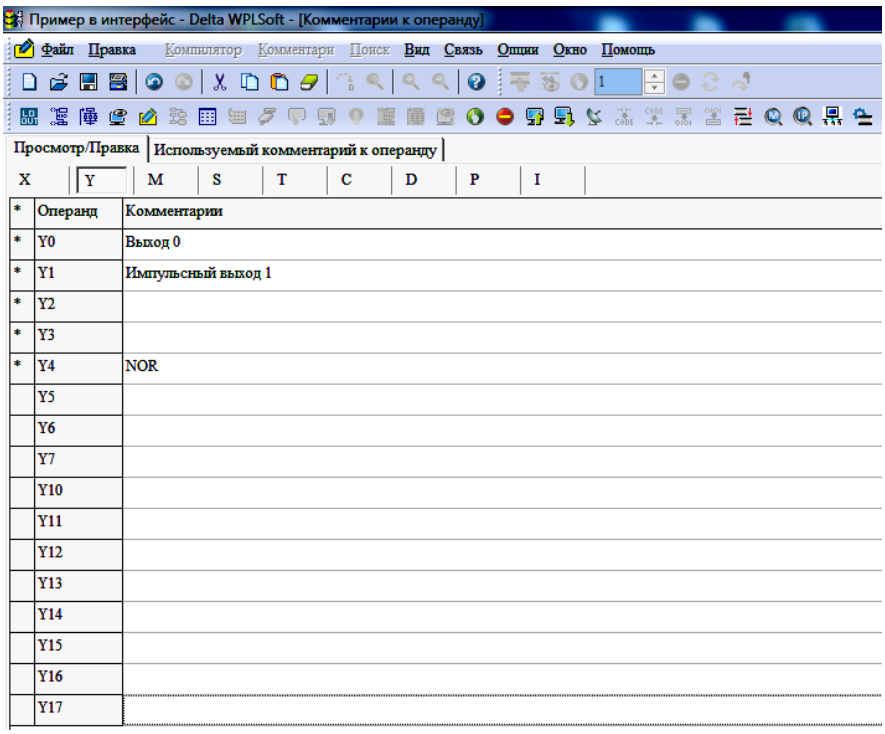
Редактируя демонстрационный проект, постоянно создавались комментарии к добавляемым операндам. Однако эти комментарии никак не отображались на диаграмме, чтобы не загромождать область редактирования. Теперь же, когда демонстрационный проект полностью завершен, комментарии можно показать. Для этого выбирается пункт меню Вид → Показать комментарии.

При этом не все выходы имеют комментарии из-за того, что комментарии прописывались к устройствам только при создании контактов и катушек. Чтобы добавить недостающие комментарии, вызывается пункт меню Вид → Правка комментариев операнда. Затем появляется окно редактирования комментариев и нажимается кнопка Y, окно должно приобрести вид, показанный на рис. 3.16.

Звездочками слева отмечены используемые в проекте операнды.

Добавляются следующие комментарии к операндам:

- Y2 – Импульсный выход 2.
- Y3 – RS-триггер.



*	Операнд	Комментарии
*	Y0	Выход 0
*	Y1	Импульсный выход 1
*	Y2	
*	Y3	
*	Y4	NOR
	Y5	
	Y6	
	Y7	
	Y10	
	Y11	
	Y12	
	Y13	
	Y14	
	Y15	
	Y16	
	Y17	

Рис. 3.16. Окно редактирования комментариев

После этого необходимо перейти к окну редактирования ступенчатых диаграмм. Для этого по нему щелкается мышью или выбирается пункт меню Вид → Ступенчатая диаграмма. Диаграмма должна принять вид, показанный на рис. 3.17.

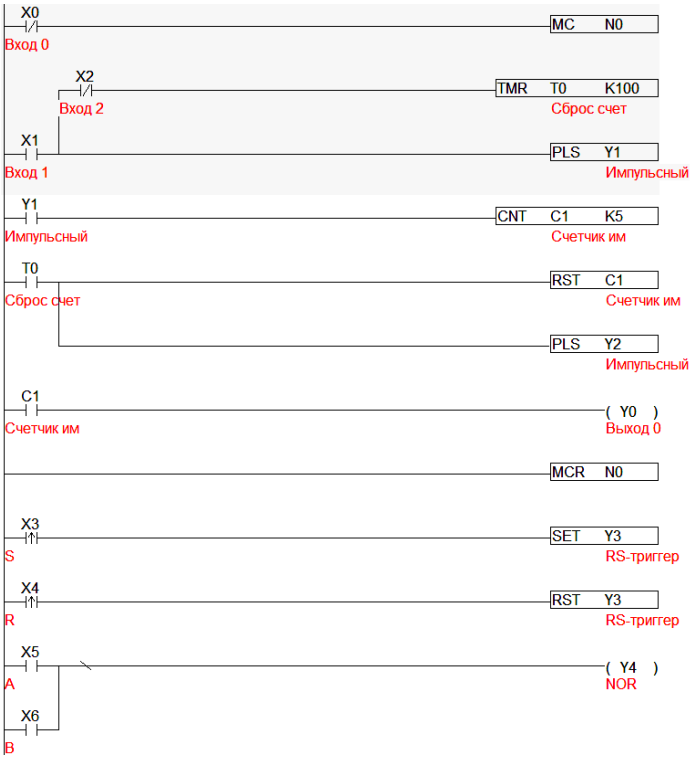


Рис. 3.17. Итоговая ступенчатая диаграмма

3.10. Компиляция проекта

После создания ступенчатой диаграммы логики демонстрационного проекта необходимо перейти к окну отображения набора инструкций, вызвав пункт меню Вид → Список инструкций. При этом будет обнаружено, что его содержимое по-прежнему не изменилось.

Чтобы перевести ступенчатую диаграмму в набор инструкций, необходимо скомпилировать проект. Это необходимая операция, и без нее вся созданная логика в виде ступенчатой диаграммы представляет собой только картинку.

За компиляцию проекта отвечает раздел меню Компилятор. Он содержит различные варианты преобразования отображений логики в список инструкций и обратно. При этом используется пункт меню Компилятор → РКС(LD) = > Инструкции. В том случае, если активно окно отображения списка инструкций, этот пункт меню будет неактивным (серым). Следует перейти к окну редактирования ступенчатых диаграмм и вызвать требуемый пункт меню.

Если в процессе компиляции появится ошибка, информация об этом будет в возникшем снизу окне (рис. 3.18).

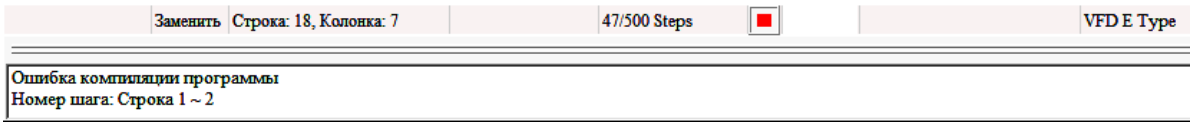


Рис. 3.18. Ошибка компиляции

Поле ввода автоматически переводится в ту область диаграммы, где обнаружена ошибка. В созданном демонстрационном проекте это вторая строка. Действительно, в этом месте диаграмма формально неверна. Необходимо исправить ее, как показано на рис. 3.19, после чего повторить процесс компиляции.

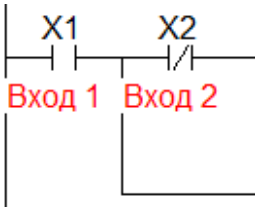


Рис. 3.19. Исправленный участок ступенчатой диаграммы

Через некоторое время появится всплывающее окно, информирующее об окончании процесса компиляции, чтобы закрыть его, нажимается кнопка OK.

При этом на диаграмме слева появятся синие числа, которые означают номера инструкций, начинающих очередную ветвь диаграммы (как правило, это разновидность инструкции LD).

После этого окно отображения списка инструкций должно выглядеть, как представлено на рис. 3.20.

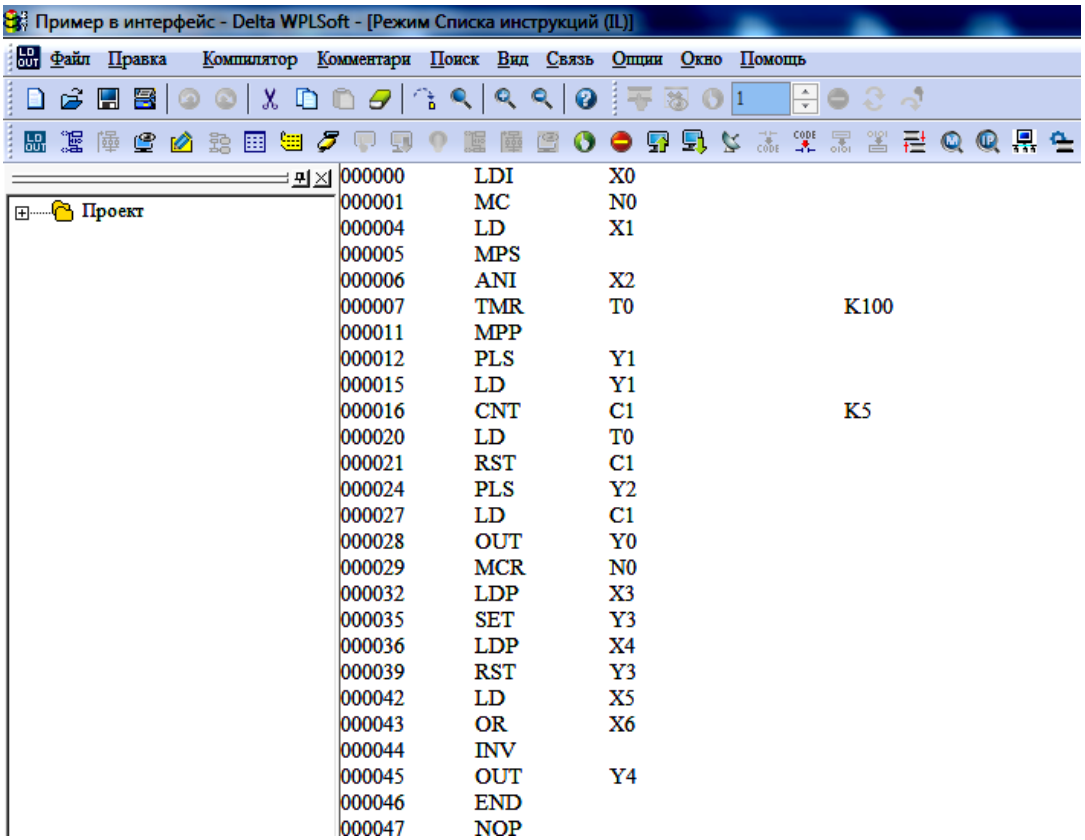


Рис. 3.20. Набор инструкций демонстрационного проекта

Синей полоской отмечена инструкция, выделенная в окне редактирования ступенчатых диаграмм.

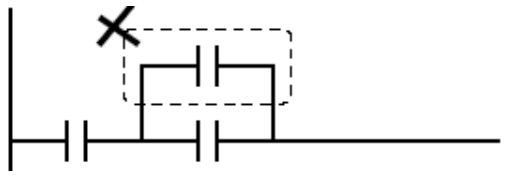
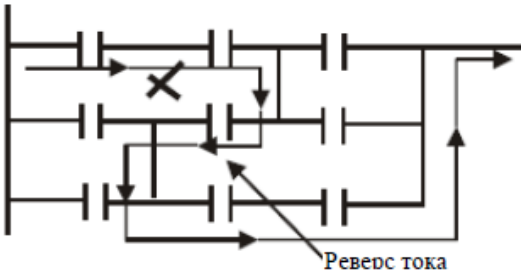
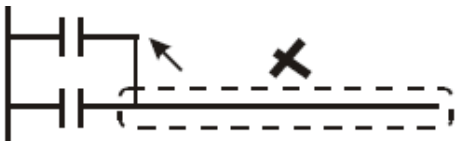
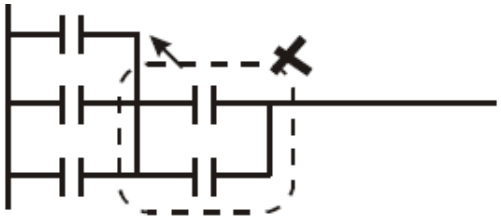
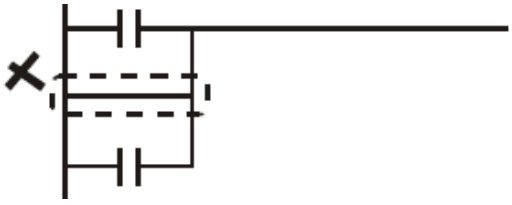
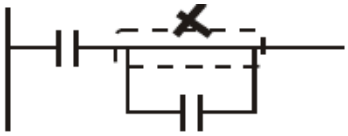
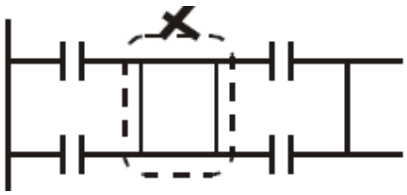
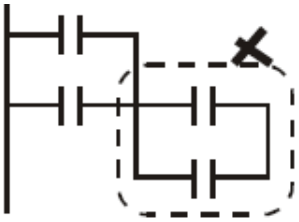
На этом создание демонстрационного проекта завершено, и его необходимо сохранить в удобном месте. После этого закрывается проект и производится выход из редактора WPLSoft.

3.11. Типичные ошибки при написании ступенчатых диаграмм

Ввиду особенностей работы процессора ПЛК при написании ступенчатых диаграмм необходимо придерживаться определенных правил. В табл. 3.1 приведены типичные ошибки, которые вызывают сбои в работе контроллера.

Таблица 3.1

Типичные ошибки при написании ступенчатых диаграмм

	Нельзя подключать параллельный контакт сверху
	«Ток» через контакты должен протекать только слева направо. Реверсивное направление не допускается
	Командная линия должна располагаться выше
	Блок ИЛИ должен быть расположен выше
	Нельзя выполнить операцию параллельно пустой линии
	
	В среднем блоке отсутствуют объекты
	Блоки должны располагаться на одном уровне

Контрольные вопросы

1. Какие имеются типы контактов операндов ПЛК?
2. Каков принцип задания параметров таймеров ПЛК?
3. Каков принцип задания параметров счетчиков ПЛК?
4. В чем назначение и каков способ задания RS-триггера.
5. Для чего используются импульсные выходы?
6. Поясните принцип работы и способ задания отключаемых блоков ступенчатых диаграмм.
7. Как обозначается инверсия на ступенчатых диаграммах?
8. Для чего необходима компиляция проекта?
9. Перечислите ошибки написания ступенчатых диаграмм, вызывающие сбой в работе ПЛК.

4. МОНИТОРИНГ ВЫПОЛНЕНИЯ ПРОГРАММ

Данный раздел посвящен проверке работоспособности созданного проекта с помощью встроенных средств программирования и отладки WPLSoft.

Редактор WPLSoft позволяет проводить имитацию выполнения программы пользователя с индикацией работы внутренних устройств контроллера (операндов). Элементы в программе, имеющие статус включенного состояния, на релейно-ступенчатой диаграмме подсвечиваются зеленым цветом.

Симулятор предоставляет пользователям функцию тестирования программы без подключения к ПЛК. Однако иногда фактический результат может быть не совсем таким же, как при подключении к ПЛК.

Для запуска симулятора создается новый или открывается существующий файл с демонстрационным проектом. При этом нажимается значок  на панели инструментов, как показано на рис. 4.1.

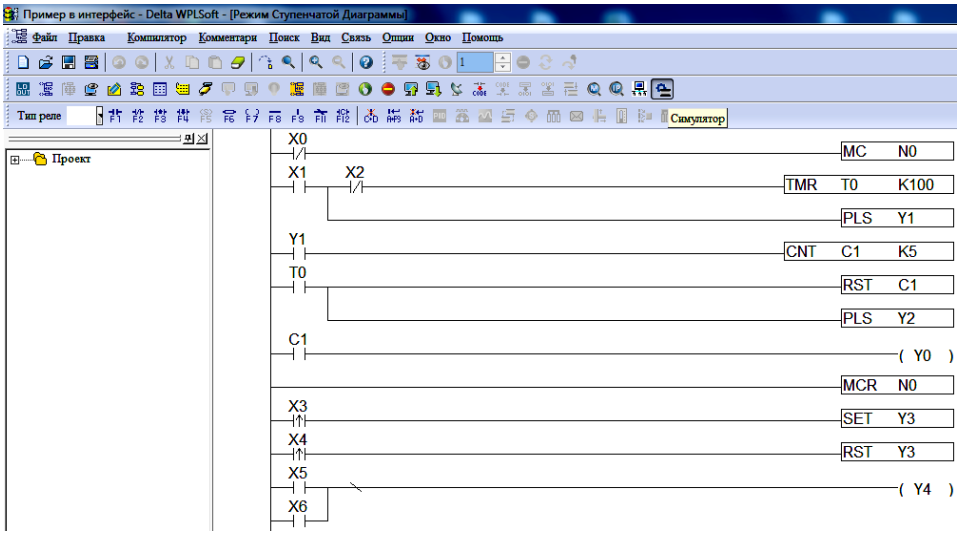


Рис. 4.1. Запуск симулятора

После запуска симулятора WPLSoft строка состояния примет вид как на рис. 4.2. В этой строке отображается состояние работы ПЛК.

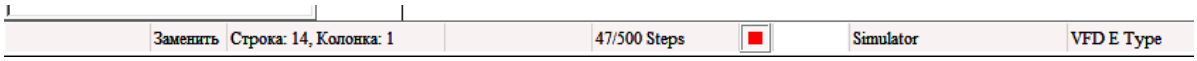


Рис. 4.2. Строка состояния симулятора

Для контроля состояния работы ПЛК в окне ПК запускается функция «Режим On-line» после нажатия на значок  на панели

инструментов, как показано на рис. 4.3. При этом рабочий режим будет таким же, как и в случае фактической связи с ПЛК.

После запуска функции «Режим On-line» зеленая часть, отображаемая в окне, является индикацией того, что контакт находится в состоянии ВКЛ или выходная катушка включена. Напротив, если зеленый цвет не отображается на контакте или на выходной катушке, это означает, что этот операнд в настоящее время выключен.

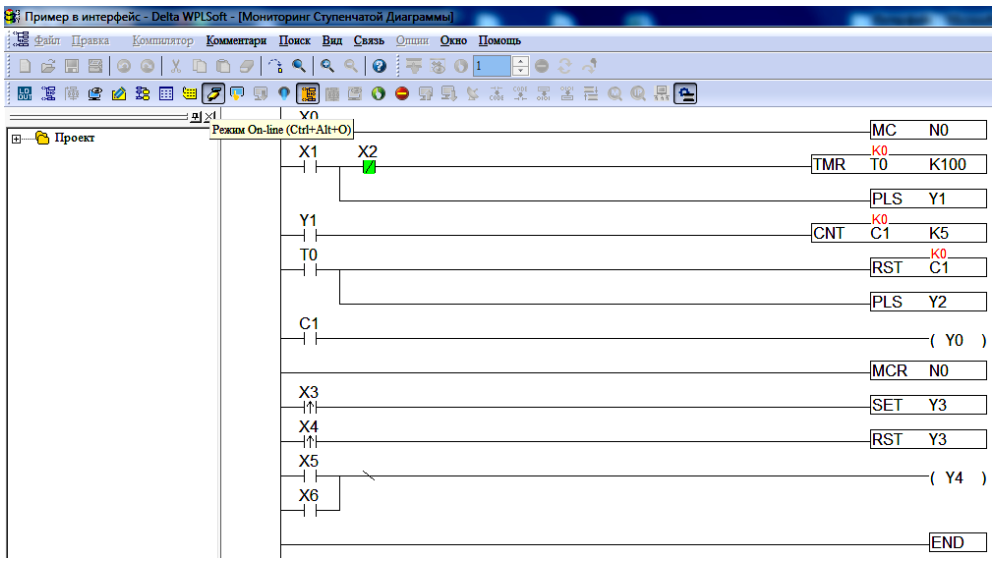


Рис. 4.3. Запуск функции «Режим On-line»

Для запуска работы ПЛК нажимается «Связь» > «Работа», как показано на рис. 4.4, или используется сочетание клавиш (Ctrl) + (F8), или выбирается соответствующий значок на панели инструментов, чтобы получить диалоговое окно подтверждения. После этого нажимается «Да», чтобы установить ПЛК в состояние RUN (Работа).

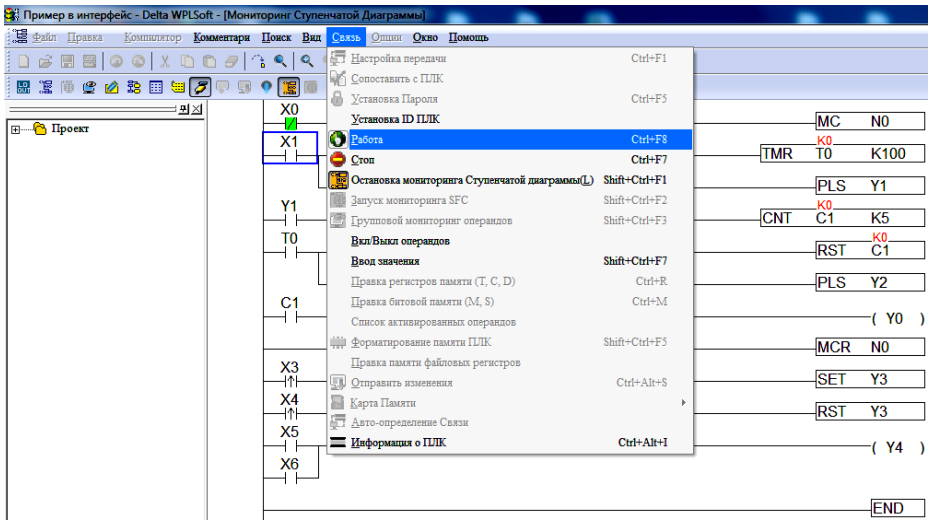


Рис. 4.4. Запуск/останов работы ПЛК

Чтобы остановить работу ПЛК, нажимается «Связь» > «Стоп», как показано на рис. 4.4, или используется сочетание клавиш (Ctrl) + (F7), или выбирается соответствующий значок на панели инструментов, чтобы получить диалоговое окно подтверждения. После этого нажимается «Да», чтобы установить ПЛК в состояние «STOP» (Стоп).

Если ПЛК находится в состоянии RUN (Работа) появляется возможность принудительного включения или выключения определенных операндов. Для этого курсор перемещается на операнд и отмечается правой кнопкой мыши, чтобы получить всплывающее меню, показанное на рис. 4.5. Данное меню позволяет изменять текущее состояние операнда.

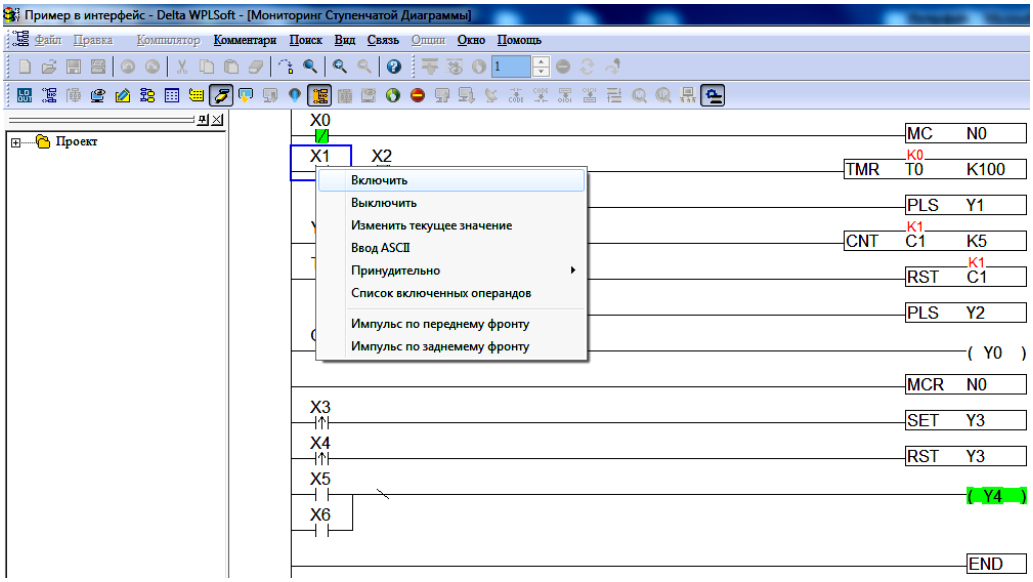


Рис. 4.5. Меню изменения состояния операнда

Контрольные вопросы

1. Для какой цели проводится мониторинг выполнения программ?
2. В чем особенности проведения мониторинга при использовании редактора WPLSoft?
3. Какова последовательность проведения мониторинга программы без подключения к ПЛК?
4. Какова последовательность проведения мониторинга программы с подключением к ПЛК?

5. ОПИСАНИЕ ПРИКЛАДНЫХ ИНСТРУКЦИЙ

5.1. Сравнение числовых данных (CMP)

На рис. 5.1 приведено окно прикладной инструкции для сравнения числовых данных.

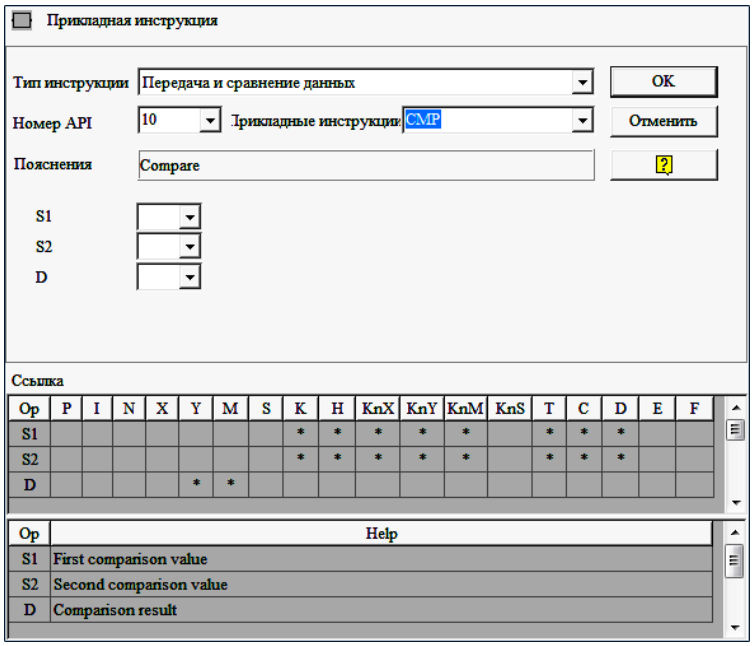


Рис. 5.1. Окно прикладной инструкции для сравнения числовых данных

Операнды:

S1: Первое сравниваемое число.

S2: Второе сравниваемое число.

D: Результат сравнения.

Описание:

1. Операнд D занимает три последовательных адреса.

2. Данные в обоих источниках S1 и S2 сравниваются друг с другом и результат сравнения (больше, меньше или равно) отображается (индицируется) благодаря использованию реле M или выхода Y. Определение, какой из этих операндов должен использоваться, выполняется по адресу результата D.

3. Данные в S1 и S2 обрабатываются как двоичные данные.

Пример программы представлен на рис. 5.2:

1. В адресе результата D в этом примере указан выход Y0. Соответствующие результаты сравнения автоматически присваиваются приращиваемым на единицу последующим адресам выхода Y0, Y1, Y2, которые имеют следующие значения:

Y0: включен, если K10 > значения регистра D10;
Y1: включен, если K10 = значению регистра D10;
Y2: включен, если K10 < значения регистра D10.

2. Y0, Y1, Y2 не изменяются, если входное условие X10 выключено.

3. Для сброса результатов сравнения используются команды RST (рис. 5.2).

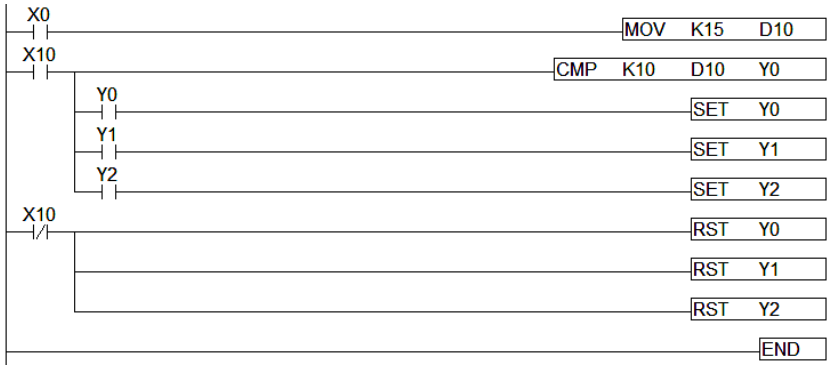


Рис. 5.2. Пример программы для сравнения числовых данных

5.2. Зонное сравнение числовых данных (ZCP)

На рис. 5.3 приведено окно прикладной инструкции для сравнения числовых данных.

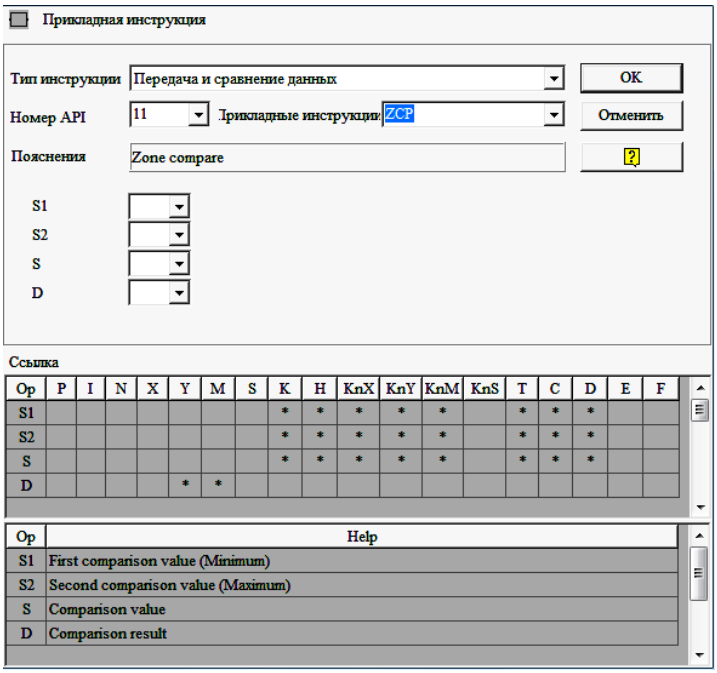


Рис. 5.3. Окно прикладной инструкции для зонного сравнения числовых данных

Операнды:

S1: Первое сравниваемое число (нижний предел).

S2: Второе сравниваемое число (верхний предел).

S: Сравниваемое число.

D: Результат сравнения.

Описание:

1. Операнд S1 должен быть меньше операнда S2.

2. Операнд D занимает три последовательных адреса.

3. Данные в источнике S сравниваются с данными в обоих источниках S1 и S2. Результат сравнения (больше, меньше или равно) отображается (индицируется) благодаря использованию реле M или выхода Y. Определение, какой из этих операндов должен задействоваться, выполняется в регистре данных (по адресу цели) – D.

4. Данные в S1, S2 и S обрабатываются как двоичные данные.

Пример программы представлен на рис. 5.4:

1. В адресе цели (D) в этом примере указано реле M0. Соответствующие результаты сравнения автоматически присваиваются приращиваемым на единицу последующим адресам реле M0, M1, M2, которые имеют следующие значения:

M0: включен, если К4 > накопленного в счетчике C1 значения;

если накопленное значение счетчика C1 находится в области от 4 до 7, то реле M1 включено;

M2: включен, если накопленное в счетчике C1 значение > К7.

2. M0, M1, M2 не изменяются, если входное условие X0 выключено.

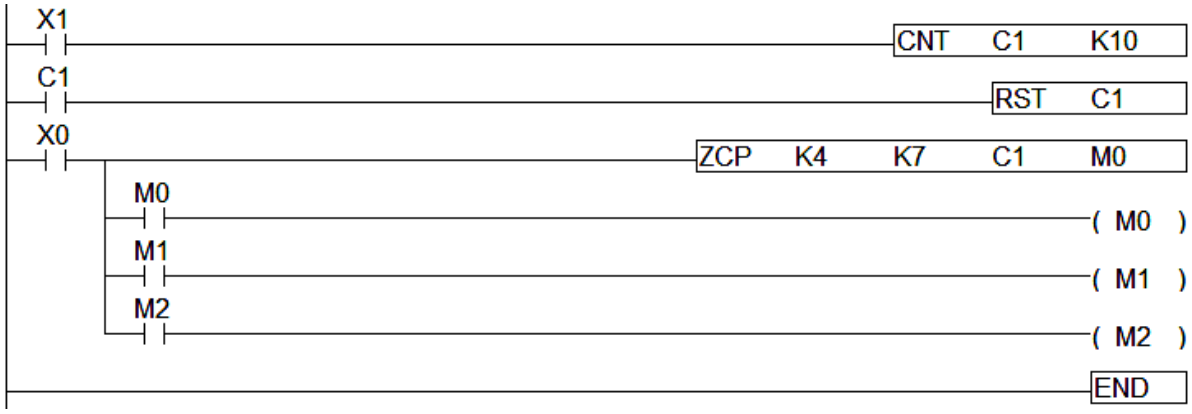


Рис. 5.4. Пример программы для зонного сравнения числовых данных

5.3. Пересылка данных (MOV)

На рис. 5.5 приведено окно прикладной инструкции.

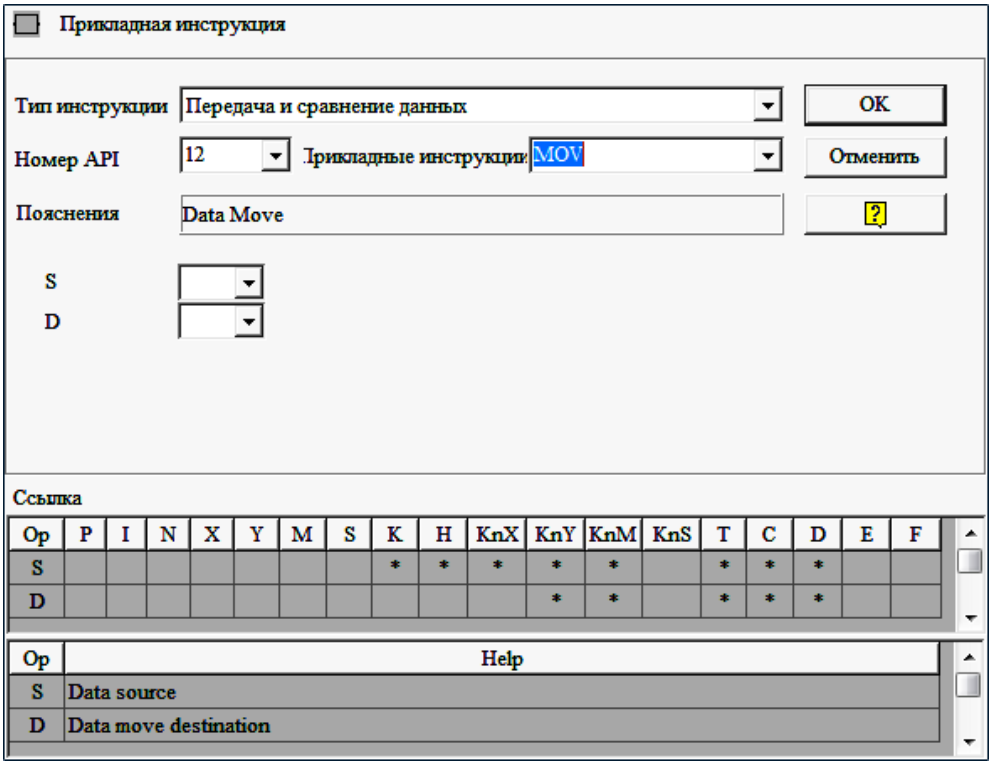


Рис. 5.5. Окно прикладной инструкции для переменных данных

Операнды:

S: Источник данных.

D: Данные цели.

Описание:

1. Инструкция служит для передачи данных от источника S к цели D. Содержимое источника S при этом не изменяется.

2. Данные в источнике S при выполнении MOV-инструкции автоматически интерпретируются как двоичные значения.

3. Инструкции выполняются в каждом цикле программы. Этого можно избежать благодаря использованию вставленной впереди импульсной функции (PLS- или PLF-инструкции или же параметра «P»).

Пример программы представлен на рис. 5.6:

1. Если входное условие X0 включено, то значение регистра D0 будет равно 100. Если X0 выключен, значение D0 не изменится.

2. Если входное условие X1 включено, то регистру D1 будет передаваться текущее значение таймера T0. Если X1 выключен, значение D1 не изменится.

3. Если входное условие X2 включено, то регистру D2 будет передаваться текущее значение счетчика C0.

4. Для сброса счетчика C0 используется команда RST.



Рис. 5.6. Пример программы для переменных данных

5.4. Пересылка блока данных (BMOV)

На рис. 5.7 приведено окно прикладной инструкции.

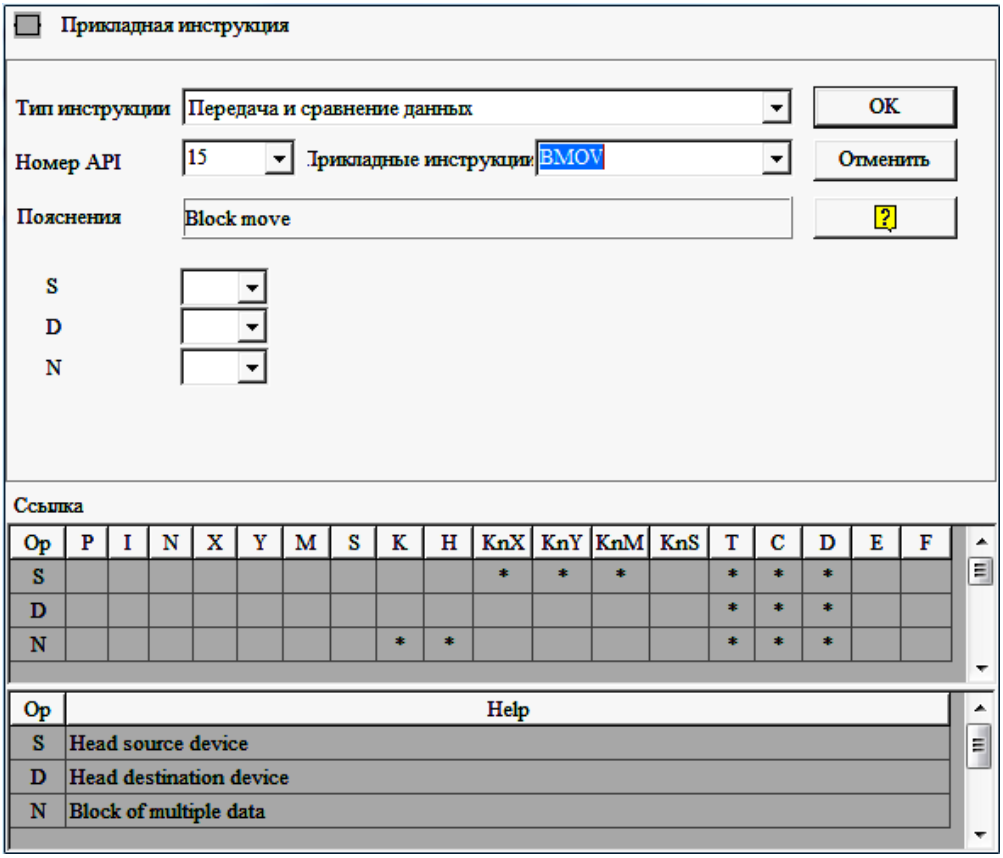


Рис. 5.7. Окно прикладной инструкции для пересылки блока данных

Операнды:

S: Источник данных.

D: Цель.

N: Число передаваемых данных.

Описание:

1. Диапазон операнда N = 1 – 512.

2. Переносится (передается) предварительно заданное количество операндов.

3. Для передачи предварительно задаются стартовый адрес S, адрес цели D и число переносимых слов N.

4. Если величина пакета данных превышает величину областей цели или источника, то передаются только слова, которые могут поместиться в области.

Пример программы представлен на рис. 5.8.

Когда X10 включен, содержимое регистров D0–D3 будет копироваться в регистры D20–D23.

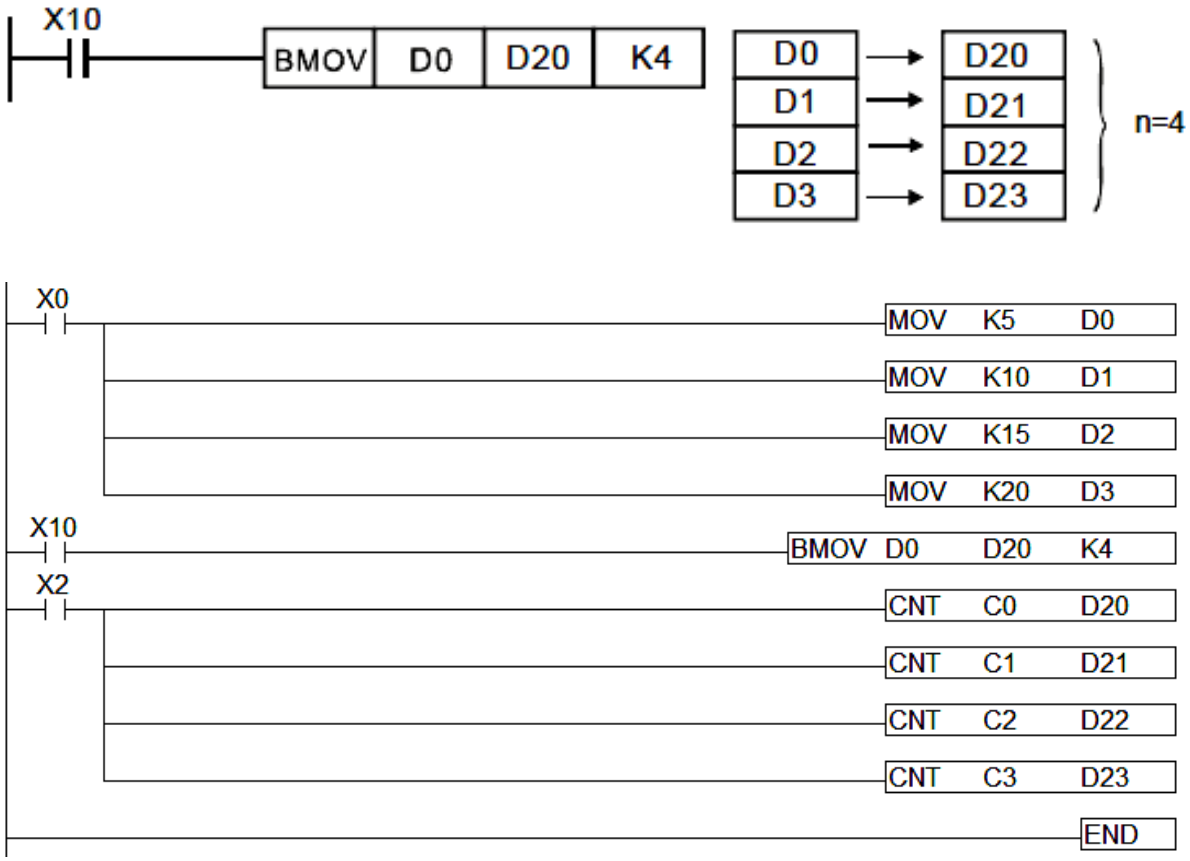


Рис. 5.8. Пример программы для пересылки блока данных

5.5. Сложение двух чисел (ADD)

На рис. 5.9 приведено окно прикладной инструкции.

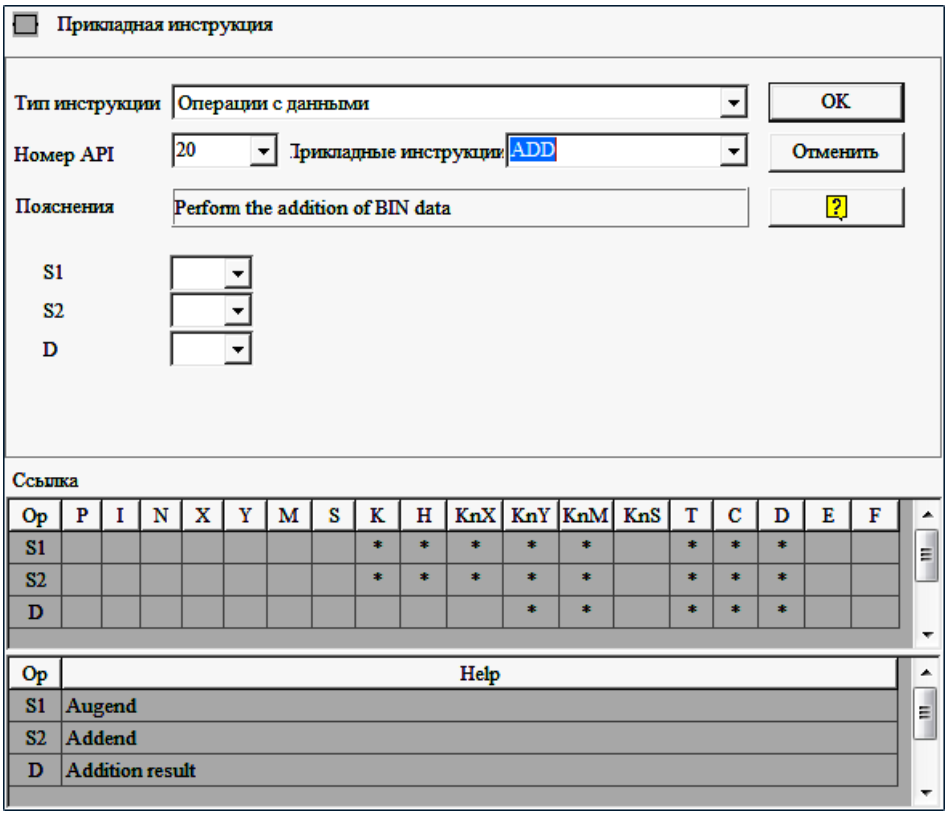


Рис. 5.9. Окно прикладной инструкции для сложения двух чисел

Операнды:

S1: Первое слагаемое.

S2: Второе слагаемое.

D: Сумма.

Описание:

1. S1 + S2 = D. Двоичные данные в адресах источников S1 и S2 суммируются. Результат суммирования запоминается в адресе цели D.

2. В старшем бите запоминается знак числа суммирования:

0: знак положительного числа.

1: знак отрицательного числа.

3. При определенных результатах счета после исполнения инструкции включаются специальные реле (флаги):

– флаг нуля (Zero) – специальное реле M1020: если результатом сложения является 0, включается флаг нуля;

– флаг заимствования (Borrow) – специальное реле M1021: если результатом сложения явилось число меньшее –32 767, включается флаг заимствования;

– флаг переноса (Carry) – специальное реле M1022: если результатом сложения явилось число, большее 32 767, включается флаг переноса.

Пример программы представлен на рис. 5.10:

1. Если включен X0, то суммируются значения данных в регистрах D0 и D10. Результат суммирования запоминается в регистре данных D20.

2. Для сброса значения регистра D20 используется команда RST.

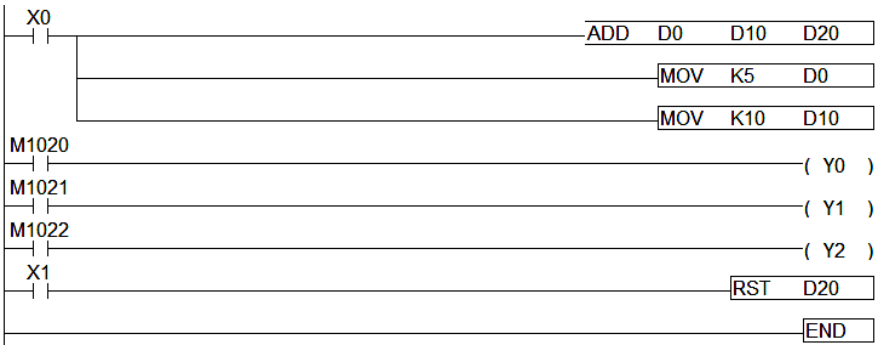


Рис. 5.10. Пример программы для сложения двух чисел

5.6. Вычитание двух чисел (SUB)

На рис. 5.11 приведено окно прикладной инструкции.

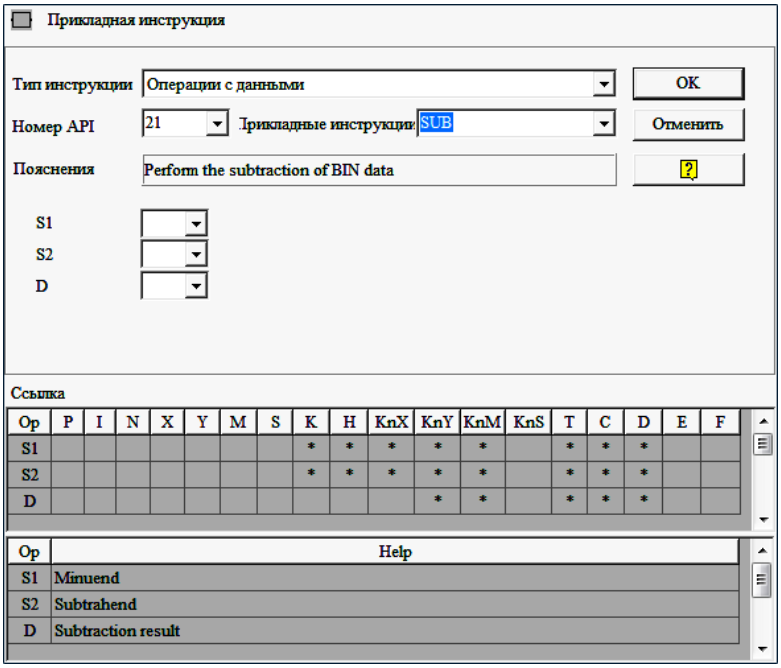


Рис. 5.11. Окно прикладной инструкции для вычитания двух чисел

Операнды:

S1: Уменьшаемое.

S2: Вычитаемое.

D: Разность.

Описание:

1. S1 − S2 = D. Значение данных в S2 вычитается из значения данных S1. Результат вычитания запоминается в адресе цели D. Операция выполняется в BIN-формате.

2. В старшем бите запоминается знак числа вычитания:

0: знак положительного числа.

1: знак отрицательного числа.

3. При определенных результатах счета после исполнения инструкции включается специальное реле (флаг).

1) Если результат операции = 0, включится Zero flag, специальное реле M1020 = ON.

2) Если результат операции меньше -32768, включится Borrow flag, специальное реле M1021 = ON.

3) Если результат операции больше 32767, включится Carry flag, специальное реле M1022 = ON.

Пример программы представлен на рис. 5.12:

1. Если включен X0, то находится разность значений данных в регистрах D0 и D10. Результат вычитания запоминается в регистре данных D20.

2. Для сброса значения регистра D20 используется команда RST.

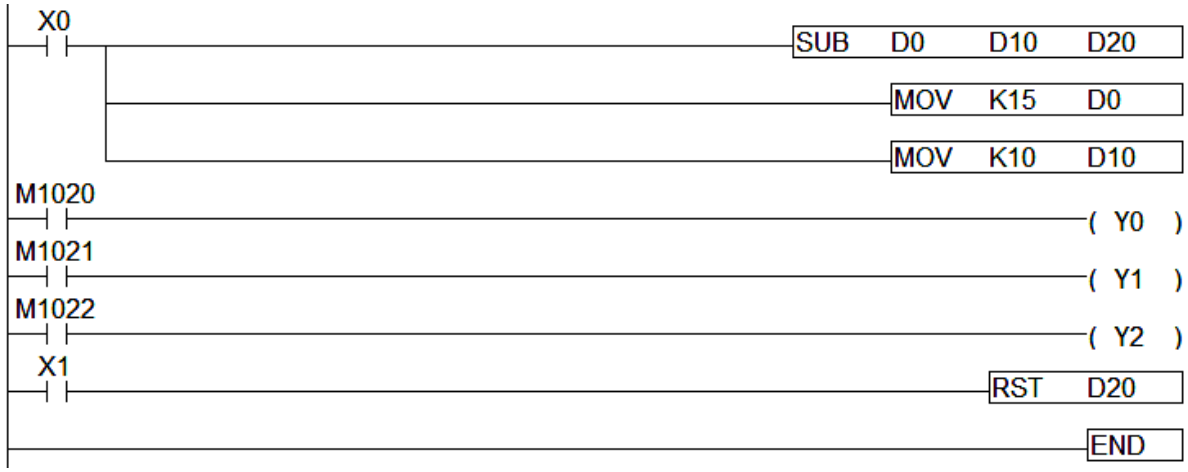


Рис. 5.12. Пример программы для вычитания двух чисел

5.7. Умножение двух чисел (MUL)

На рис. 5.13 приведено окно прикладной инструкции.

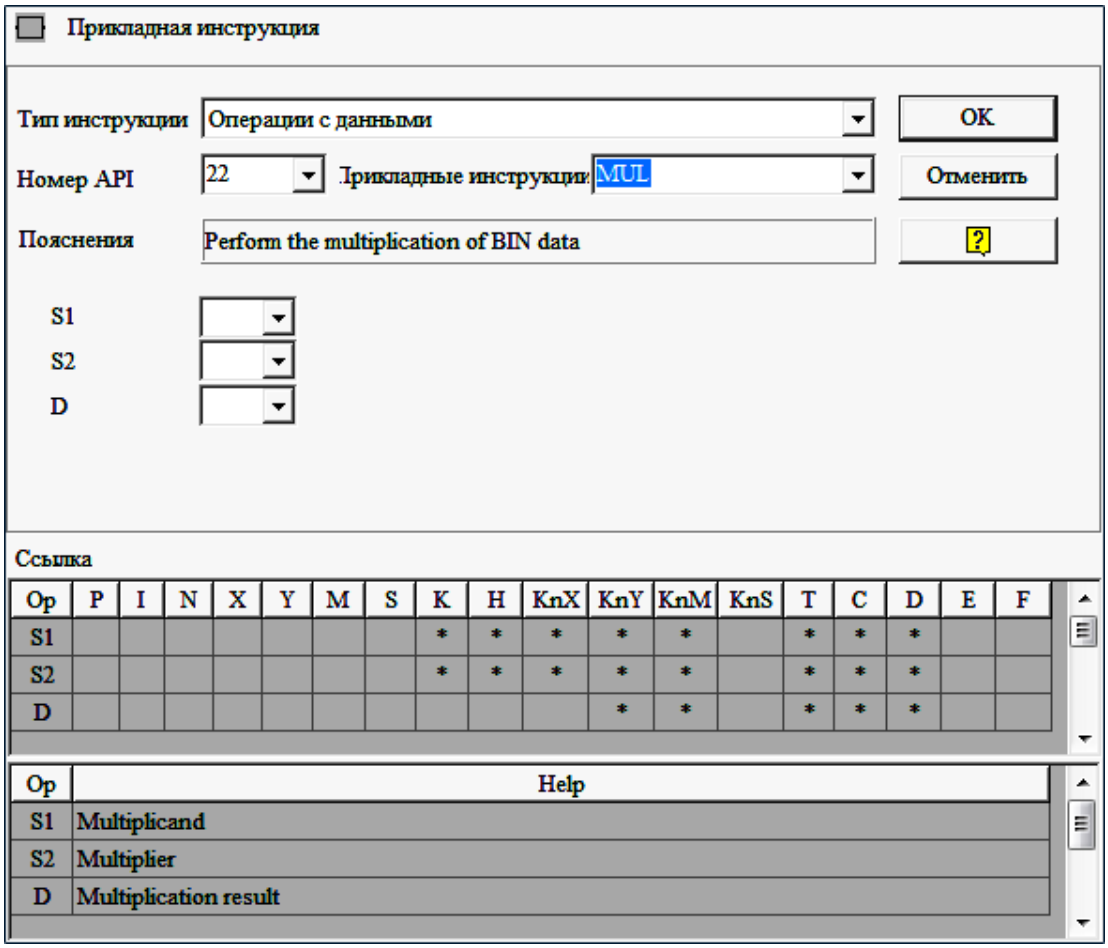


Рис. 5.13. Окно прикладной инструкции для умножения двух чисел

Операнды:

S1: Множимое.

S2: Множитель.

D: Произведение.

Описание:

В 16-битном режиме операнд D занимает 2 адреса S1 × S2 = D. Данные в S1 и S2 перемножаются между собой. Результат умножения запоминается по адресу операнда, указанного в D и в следующем за ним адресе операнда. Операция выполняется в BIN-формате.

Пример программы представлен на рис. 5.14:

Если включен X0, то находится произведение значений данных в регистрах D0 и D10. Результат умножения запоминается в регистре данных D20, а знак результата умножения в регистре, следующем за ним – D21.



Рис. 5.14. Пример программы для умножения двух чисел

5.8. Деление двух чисел (DIV)

На рис. 5.1 приведено окно прикладной инструкции.

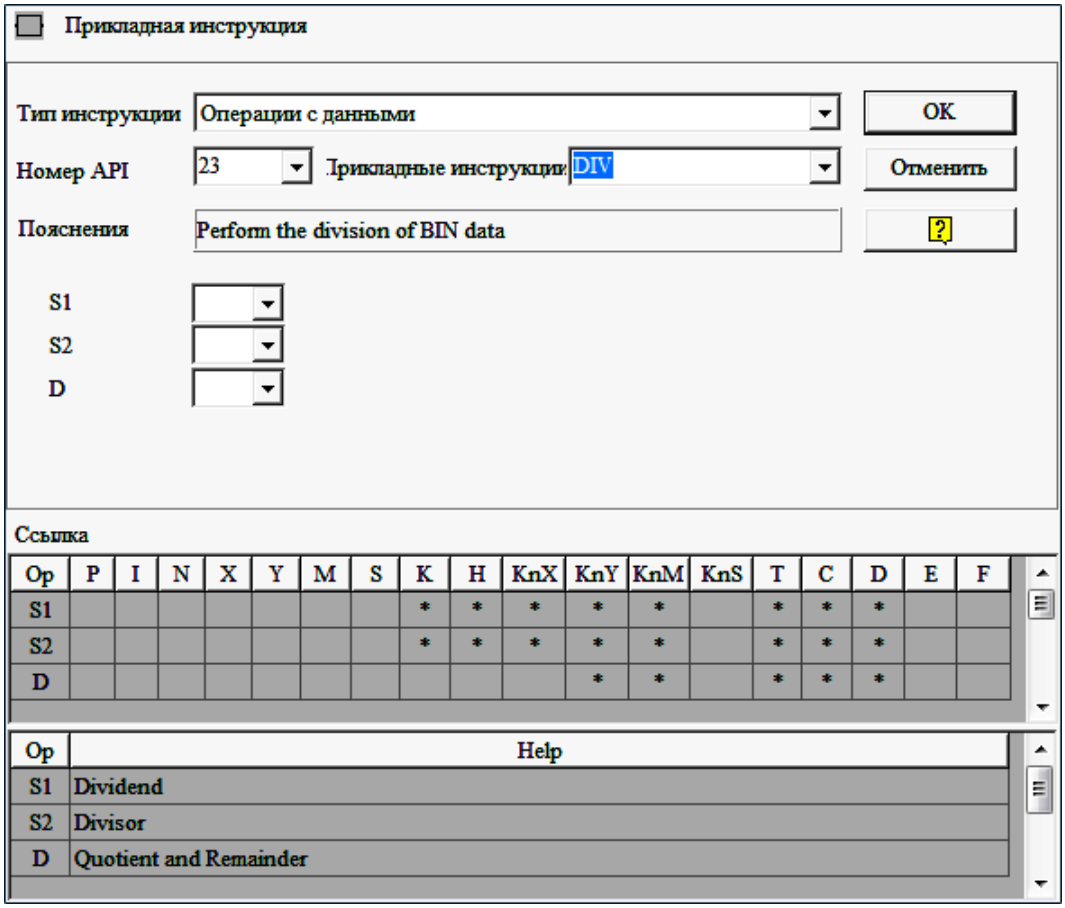


Рис. 5.15. Окно прикладной инструкции для деления двух чисел

Операнды:
S1: Делимое.
S2: Делитель.
D: Частное и остаток.

Описание:

В 16-битном режиме операнд D занимает 2 адреса S1 : S2 = D. Выполняется деление данных в S1 и S2. Данные в S1 соответствуют делимому, в S2 – делителю. Результат деления запоминается по адресу операнда, указанного в D. Остаток делимого запоминается в следующем адресе операнда. Операция выполняется в BIN-формате.

Пример программы представлен на рис. 5.16.

Если включен X0, то выполняется деление значений данных в регистрах D0 и D10. Результат деления запоминается в регистре данных D20, а остаток деления запоминается в D21.



Рис. 5.16. Пример программы для деления двух чисел

5.9. Инкрементирование (увеличение на 1 INC)

На рис. 5.17 приведено окно прикладной инструкции.

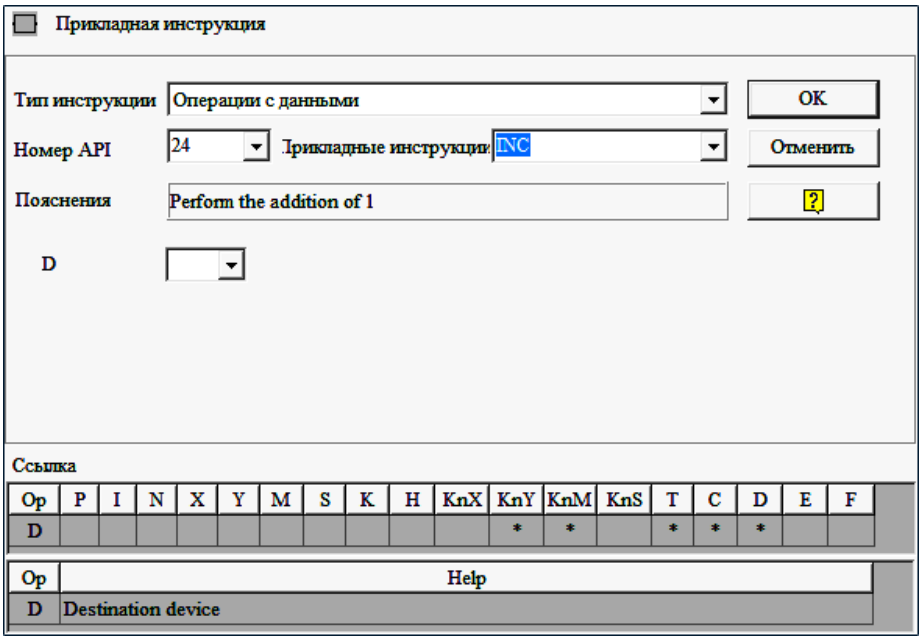


Рис. 5.17. Окно прикладной инструкции для инкрементирования

Операнды:

D: Цель.

Описание:

1. К числовому значению данных прибавляется число 1, как только выполнится входное условие.

2. Инструкция выполняется в каждом цикле программы. Этого можно избежать благодаря введению впереди функций импульса (PLS или PLF-инструкции) или применив командный параметр P.

3. Если при 16-битовом операнде значение 1 добавится к числу 32 767, то запишется значение –32 768. При этом не появляется никакого предупреждения (флага).

Пример программы представлен на рис. 5.18.

Значение данных в регистре D0 при наличии входного сигнала X0 увеличится на число 1.

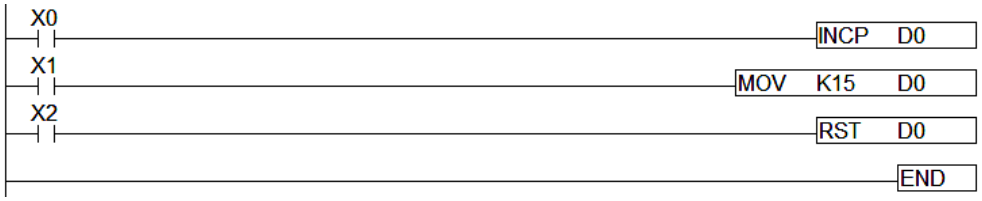


Рис. 5.18. Пример программы для инкрементирования

5.10. Декрементирование (уменьшение на 1 DEC)

На рис. 5.19 приведено окно прикладной инструкции.

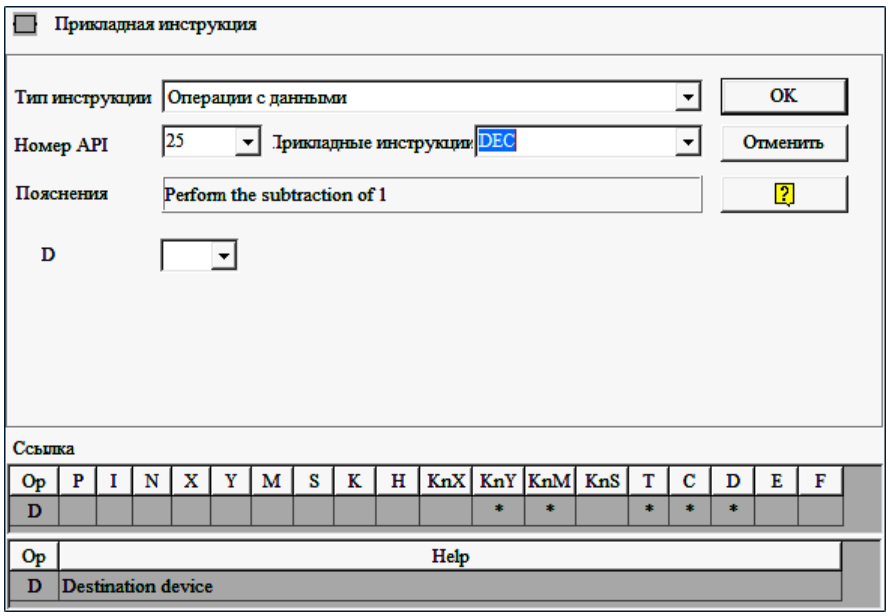


Рис. 5.19. Окно прикладной инструкции для декрементирования

Операнды:

D: Цель.

Описание:

1. Из значения числа, имеющегося в регистре D, вычитается число 1, как только выполнится входное условие.

2. Инструкция выполняется в каждом цикле программы. Этого можно избежать благодаря введению впереди функций импульса (PLS или PLF-инструкции) или применив командный параметр P.

3. Если при 16-битовом операнде значение 1 отнимется от числа –32 768, то запишется значение 32 768. При этом не появляется никакого предупреждения (флага).

Пример программы представлен на рис. 5.20.

Значение данных в регистре D0 при наличии входного сигнала X0 уменьшится на число 1.

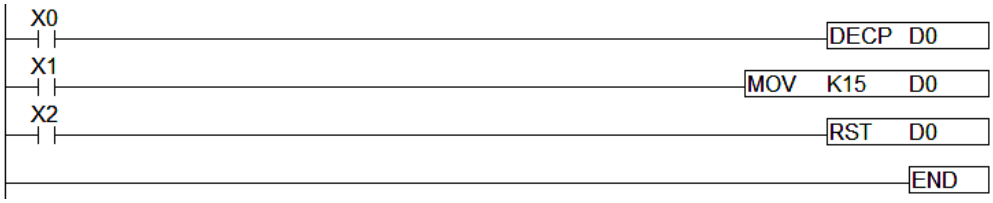


Рис. 5.20. Пример программы для декрементирования

5.11. Кольцевой сдвиг влево/вправо (ROL/ROR)

На рис. 5.21 приведено окно прикладной инструкции.

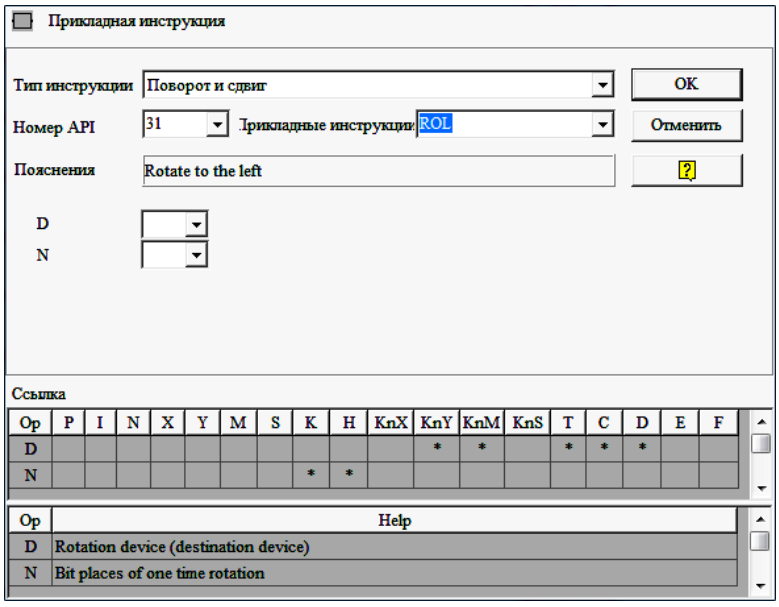


Рис. 5.21. Окно прикладной инструкции кольцевого сдвига влево/вправо

Операнды:

D: Операнд для кольцевого сдвига.

N: Количество сдвигаемых бит.

Эта команда сдвигает битовое состояние операнда D на N мест (бит).

Пример программы представлен на рис. 5.22.

Каждый раз, когда контакт реле M1 переходит из состояния «0» в состояние «1», происходит переключение контактов внутренних реле, указанных в прикладной инструкции ROL, на заданное количество мест.

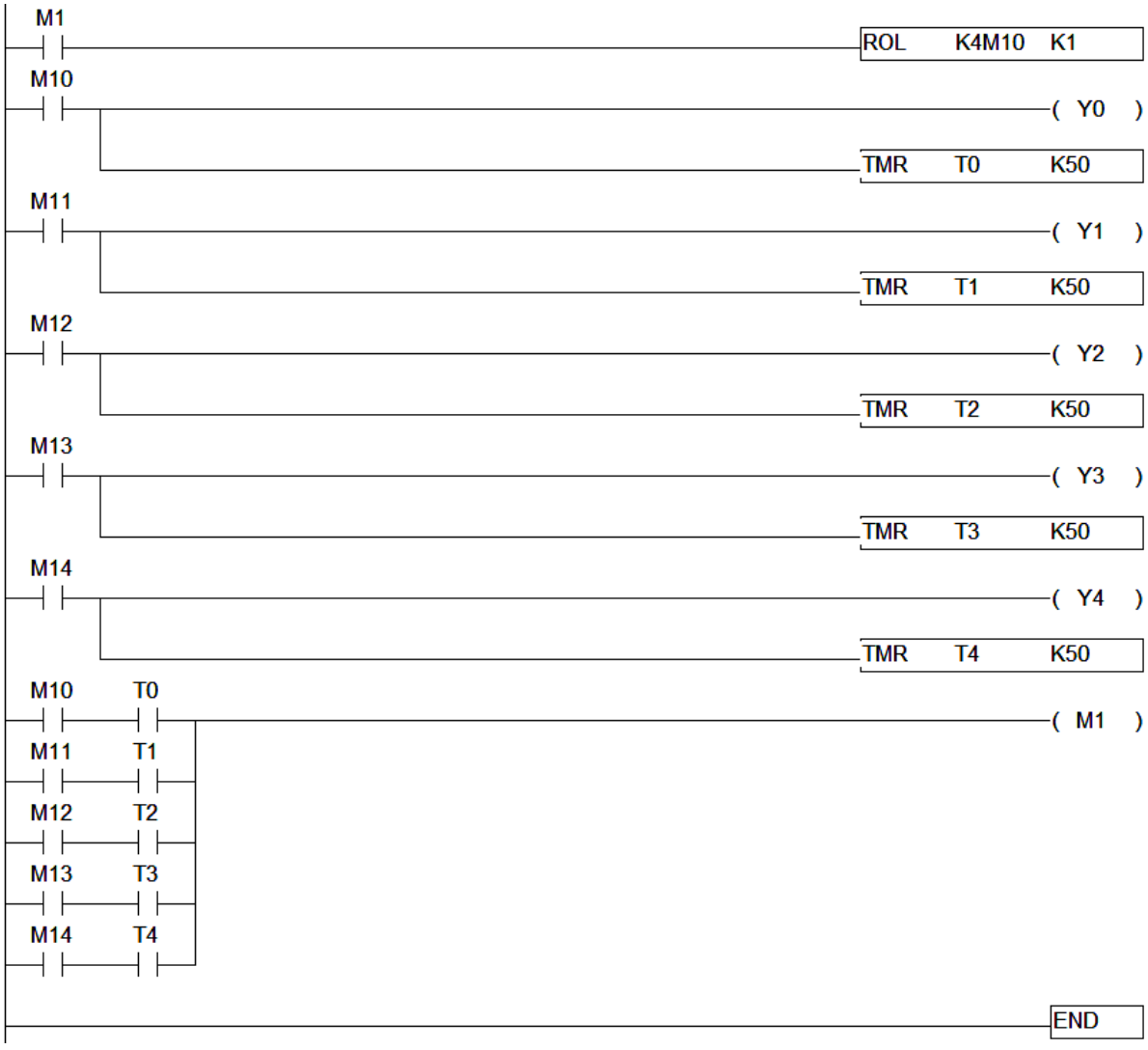


Рис. 5.22. Пример программы для кольцевого сдвига влево/вправо

5.12. Высокоскоростная обработка (высокоскоростной счет DHSCS)

На рис. 5.23 приведено окно прикладной инструкции.

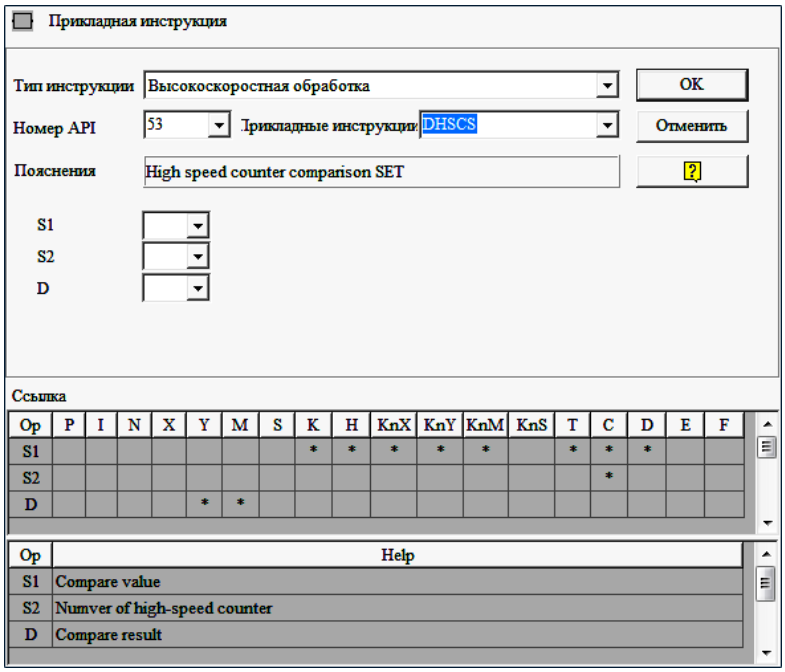


Рис. 5.23. Окно прикладной инструкции для высокоскоростной обработки

Операнды:

S1: Устанавливаемое значение для сравнения.

S2: Значение счетчика C235.

D: Результат сравнения.

Высокоскоростной счет возможен только при использовании платы расширения EME-PG01. Для автоматического счета устанавливается заданное значение счетчика с помощью команды DHSCS и включается специальное реле M1028. Контакт высокоскоростного счетчика C235 замкнется, когда текущее значение будет равно заданному значению. Для сброса счетчика C235 используется специальное реле M1029.

В специальном регистре D1044 выбирается один из трех режимов счета:

1. A-B фазный режим (D1044=0): при использовании 2-фазного датчика импульсов со сдвигом сигналов A и B на 90^{o}. Датчик импульсов должен быть подключен к клеммам $\overline{A}$, $\overline{B}$ и GND.

2. Импульсы + сигнал направления (D1044=1): по входу A идут счетные импульсы, а по B сигнал, определяющий направления счета (инкрементирование или декрементирование). Датчик импульсов должен быть подключен к клеммам $\overline{A}$, $\overline{B}$ и GND.

3. Импульсы + флаг направления (D1044=2): по входу идут счетные импульсы, а направление счета определяется специальным реле M1030. Датчик импульсов должен быть подключен к клеммам $\overline{B}$ и GND.

Пример программы представлен на рис. 5.24.

1. При установке реле M100 = 0 выбирается режим счета A, B импульсов, для M101=1 выбирается режим счета импульсов + направление, для M102=2 режим счета импульсов + флаг направления.

2. Специальное реле M1030 используется для установки направления счета.

3. При изменении сигнала реле M0 с выключенного на включенное состояние происходит выполнение команды сравнения высокоскоростного счетчика DHSCS. При изменении значения счетчика C235 с H1004F на H10050 или с H10051 на H10050 реле M2 будет включено.

4. При включении реле M1 происходит выполнение команды сравнения высокоскоростного счетчика DHSCS. При изменении значения счетчика C235 с H2 на H3 или с H4 на H3 реле M3 будет включено.

5. Специальное реле M1028 разрешает (M1028 = 1) или запрещает (M1028=0) функцию работы высокоскоростного счетчика. Специальное реле M1018 используется для запуска функции счета (M1018=1).

6. Специальный регистр D1025 сохраняет младший байт текущего значения высокоскоростного счетчика C235, а специальный регистр D1026 сохраняет старший байт.

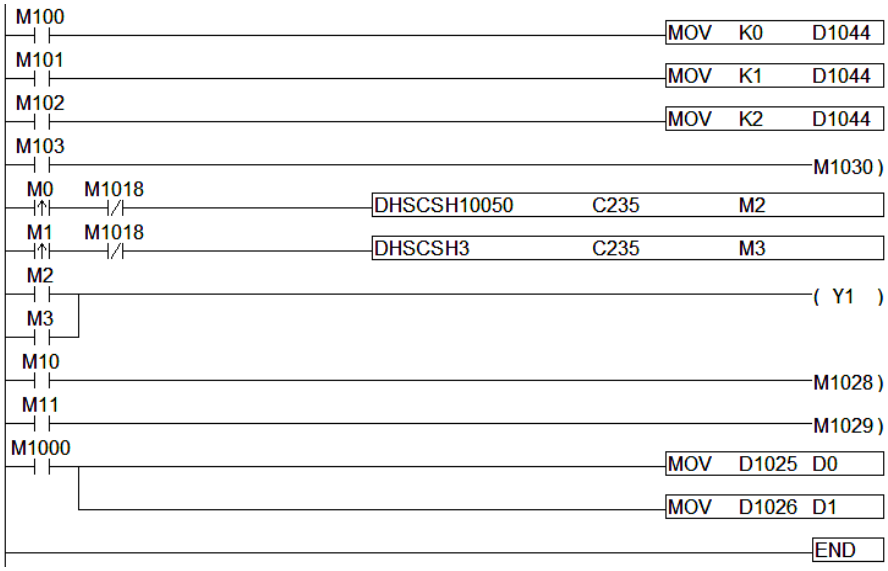


Рис. 5.24. Пример программы для высокоскоростной обработки

5.13. Управление параметрами ПИД-регулятора преобразователя частоты (FPID)

На рис. 5.25 приведено окно прикладной инструкции.

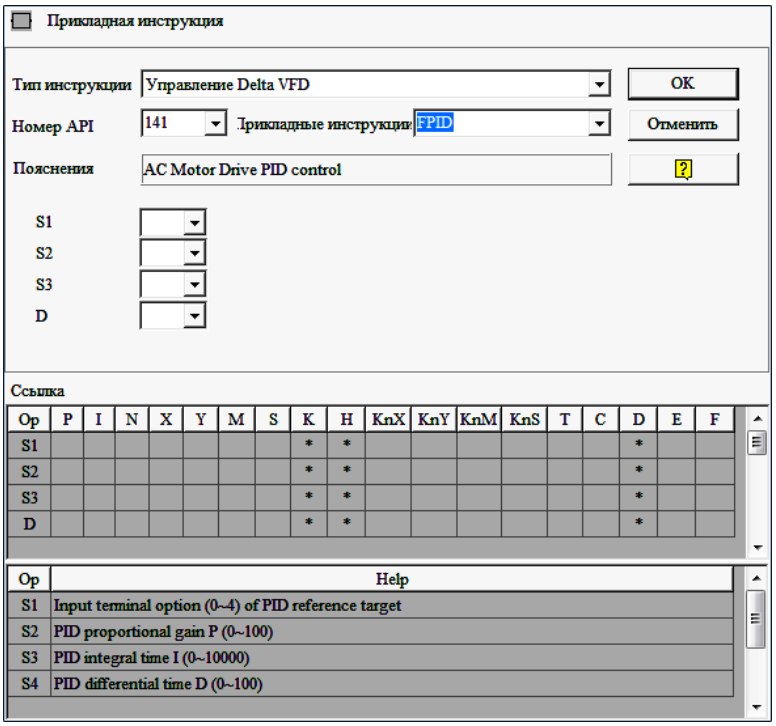


Рис. 5.25. Окно прикладной инструкции для управления параметрами ПИД-регулятора преобразования частоты

Операнды:

S1: Выбор источника сигнала задания для ПИД-регулятора.

S2: Коэффициент передачи пропорциональной составляющей.

S3: Коэффициент передачи интегральной составляющей.

S4: Коэффициент передачи дифференциальной составляющей.

Прикладная инструкция позволяет напрямую управлять параметрами 10.00, 10.02, 10.03 и 10.04 ПИД-регулятора преобразователя частоты.

Пример программы представлен на рис. 5.26.

1. Реле M0 = 1 устанавливает значение S1 = 0 (ПИД-регулятор отключен), S2 = 0, S3 = 1 (единицы измерения = 0,01 с) и S4 = 1 (единицы измерения = 0,01 с).

2. Реле M1 = 1 устанавливает значение S1 = 0 (ПИД-регуятор отключен), S2 = 1 (единицы измерения = 0,01 с), S3 = 0 и S4 = 0.

3. Реле M2 = 1 устанавливает значение S1 = 1 (источник задания сигнала для ПИД-регулятора цифровая клавиатура пульта управления), S2 = 1 (единицы измерения = 0,01 с), S3 = 0 и S4 = 0.

4. Специальный регистр D1027 – выходной регистр ПИД-регулятора (заданная частота ПИД-регулятора).

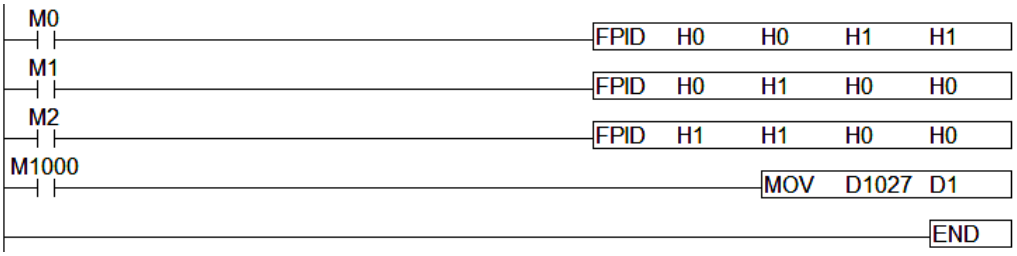


Рис. 5.26. Пример программы для управления параметрами ПИД-регулятора преобразования частоты

5.14. Управление выходной частотой преобразователя частоты (FREQ)

На рис. 5.27 приведено окно прикладной инструкции.

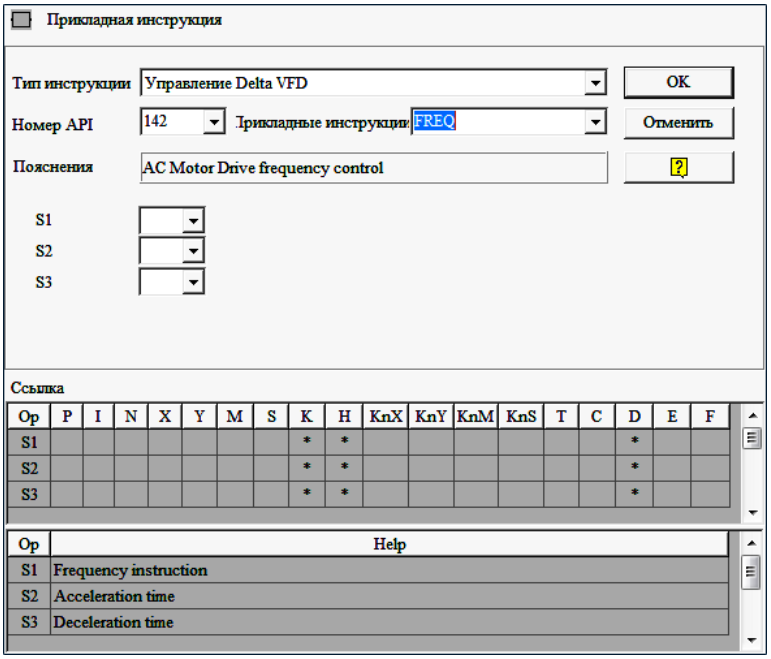


Рис. 5.27. Окно прикладной инструкции для управления выходной частотой преобразования частоты

Операнды:
S1: Заданная частота.
S2: Время разгона.
S3: Время замедления.

Команда FREQ позволяет управлять скоростью вращения привода и временем разгона/замедления. Специальные реле M1025 и

M1026 позволяют запускать/останавливать двигатель (команды RUN/STOP) и изменять направление вращения.

Пример программы представлен на рис. 5.28.

1. Специальное реле M1025 предназначено для запуска (M1025 = = 1) и останова (M1025=0) преобразователя частоты. Специальное реле M1026 для изменения направления вращения (M1026 = 1 – вперед, M1026 = 0 – назад). Специальное реле M1015 = 1, когда заданная частота достигнута.

2. При X0=1 будет задана частота K1500 (15,00 Гц) и разгон/замедление = 0.

3. При X1=1 будет задана частота K4000 (40,00 Гц), разгон K50 (5 с), замедление K60 (6 с).

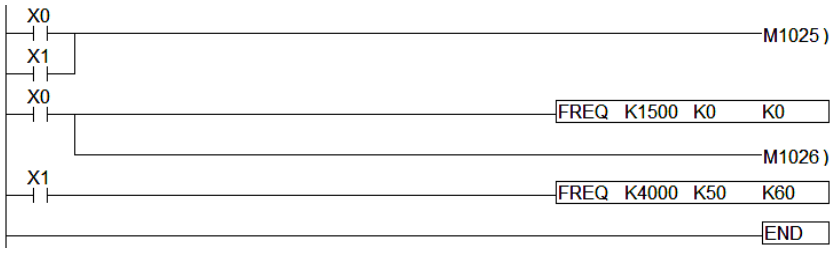


Рис. 5.28. Пример программы для управления выходной частотой преобразования частоты

5.15. Чтение/запись параметров преобразователя частоты (RPR/WPR)

На рис. 5.29 приведено окно прикладной инструкции.

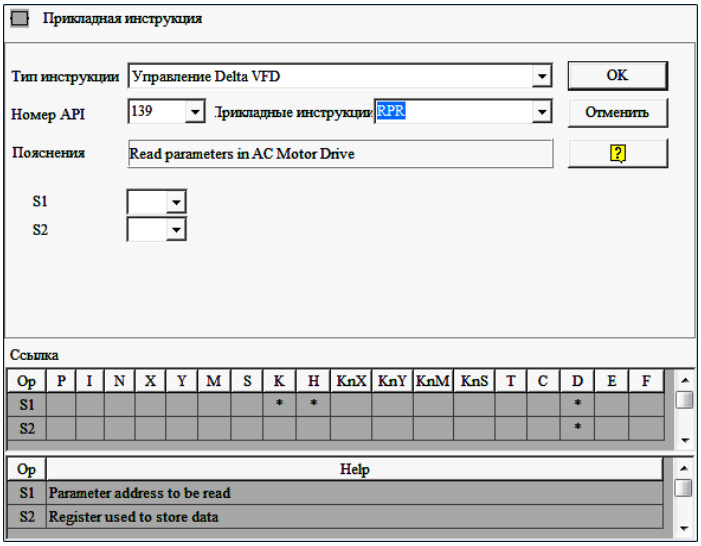


Рис. 5.29. Окно прикладной инструкции для чтения/записи параметров преобразователя частоты

Операнды:

S1: Адрес считываемых данных.

S2: Регистр сохранения данных.

Команды RPR и WPR могут использоваться для чтения/записи параметров преобразователя частоты и коммуникационных адресов.

На рис. 5.30 приведено окно прикладной инструкции.

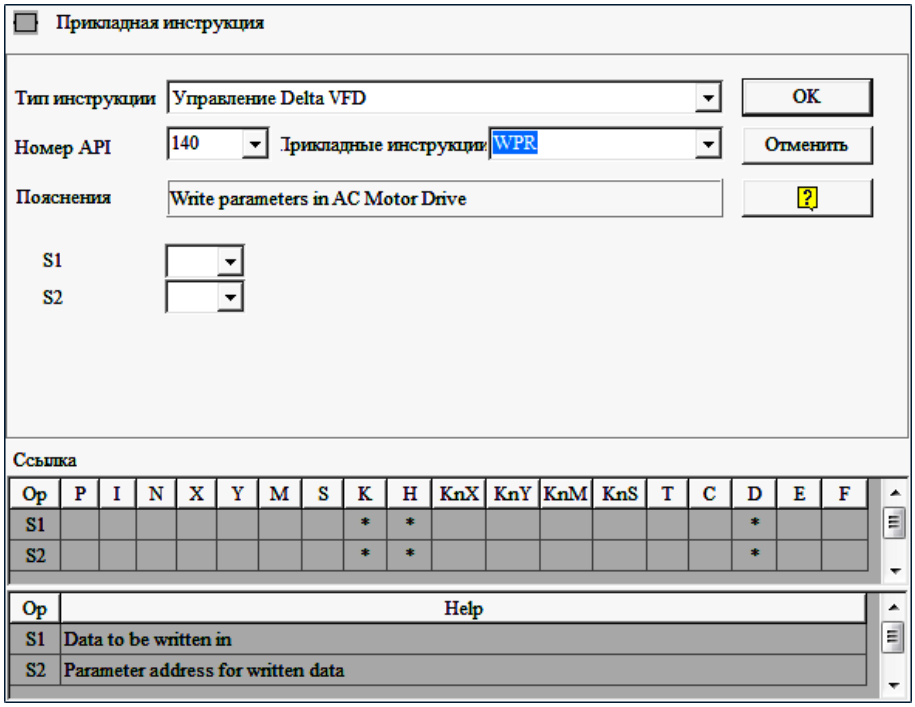


Рис. 5.30. Окно прикладной инструкции для чтения/записи преобразователя частоты и коммуникационных адресов

Операнды:

S1: Адрес источника данных.

S2: Регистр сохранения данных.

Пример программы представлен на рис. 5.31.

1. При M3=1 происходит чтение данных из коммуникационных адресов преобразователя частоты H2100, H2101 и запись в регистры D0, D1 соответственно.

2. При M0=1 производится запись данных из регистра D10 в коммуникационный адрес H2001.

3. При M1=1 производится запись значения H2 в коммуникационный адрес H2000 ПЧ – это команда пуск для ПЧ.

4. При M2=1 производится запись значения H1 в коммуникационный адрес H2000 ПЧ – это команда стоп для ПЧ.

5. При успешном выполнении записи данных в параметры или коммуникационные адреса ПЧ специальное реле M1017=1.



Рис. 5.31. Пример программы для чтения/записи параметров преобразователя частоты и коммуникационных адресов

Контрольные вопросы

1. Какие прикладные инструкции можно применять при использовании ПЛК «Delta Electronics» в редакторе WPLSoft?
2. Каков порядок проведения сравнения числовых данных?
3. Каков порядок проведения зонного сравнения числовых данных?
4. Последовательность выполнения пересылки данных.
5. Последовательность выполнения пересылки блока данных.
6. Последовательность сложения и вычитания двух чисел.
7. Последовательность умножения и деления двух чисел.
8. Порядок выполнения инкрементирования и декрементирования.
9. Порядок выполнения кольцевого сдвига влево/вправо.
10. Порядок проведения высокоскоростного счета.
11. Порядок выполнения прикладных инструкций при работе с преобразователем частоты.

6. ПРИМЕРЫ НАПИСАНИЯ ПРОГРАММ

Часто бывает необходимо использовать для старта и стопа кнопки без фиксации, но с самоблокировкой выхода. Примеры реализации таких схем представлены на рис. 6.1, 6.2 и 6.3.

6.1. Самоблокировка выхода с приоритетом кнопки СТОП

Когда X1=вкл., а X2=выкл., то выход Y1=вкл. до тех пор, пока X2 не разомкнется (рис. 6.1).

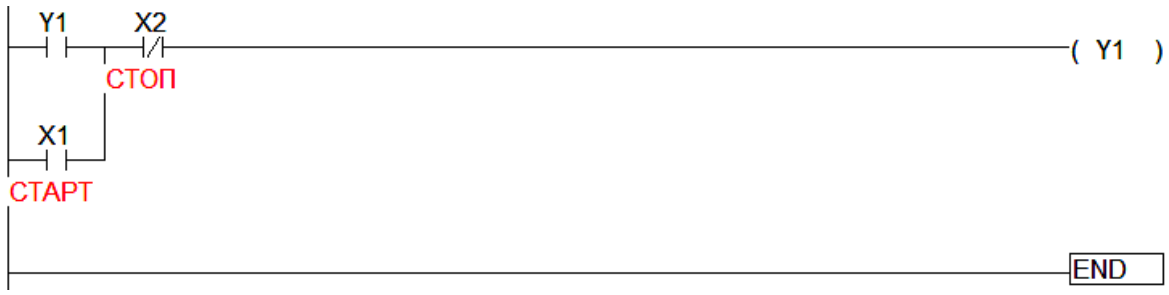


Рис. 6.1. Пример программы для самоблокировки выхода с приоритетом кнопки СТОП

6.2. Самоблокировка выхода с приоритетом кнопки СТАРТ

Когда X1=вкл., а X2=выкл., то выход Y1=вкл. Если X2 разомкнется, выход Y1 все равно останется включенным (рис. 6.2).

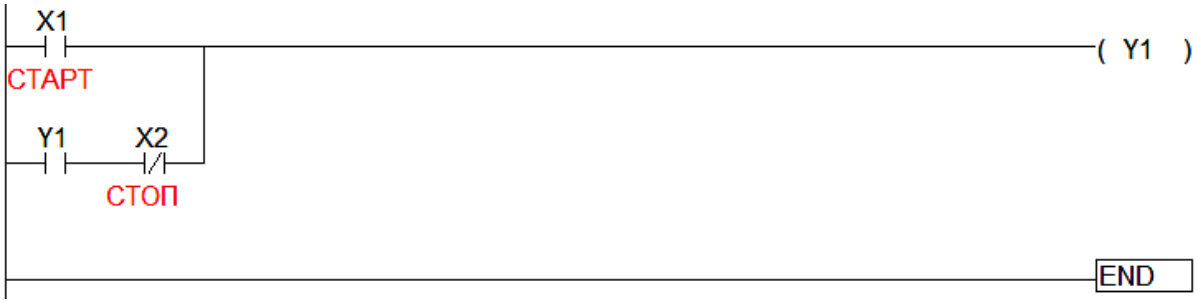


Рис. 6.2. Пример программы для самоблокировки выхода с приоритетом кнопки СТАРТ

6.3. Самоблокировка выхода с использованием команд SET и RST

ПЛК выполняет программу сверху вниз и, следовательно, приоритетом будет обладать команда, расположенная ниже.

Если одновременно попарно замкнуты контакты X1, X2 и X3, X4, то в верхней схеме выход Y1=0, а в нижней – Y2=1 (рис. 6.3).



Рис. 6.3. Пример программы для самоблокировки выхода с использованием команд SET и RST

6.4. Схема с условным управлением

Контакты X1 и X3 включают/выключают выход Y1 автономно, а X2 и X4 могут управлять состоянием выхода Y2 только при условии, что Y1 включен, т.е. выход Y1 является последовательным контактом (логическим И) для схемы примера, представленного на рис. 6.4.

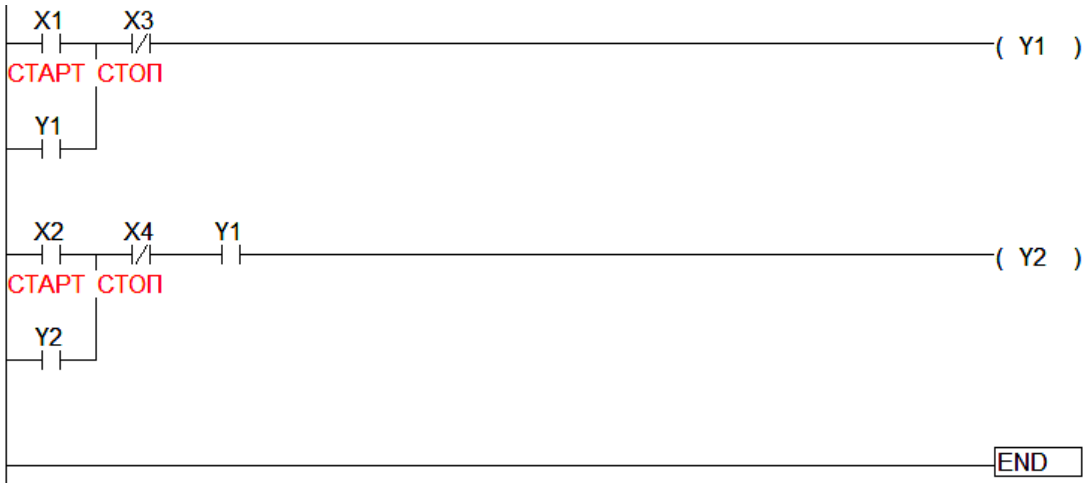


Рис. 6.4. Пример программы

6.5. Схема с взаимоблокировкой

Схема исключает одновременное включение двух выходов. Когда включен один выход, второй будет заблокирован. При одновременном замыкании контактов X1 и X2 приоритет будет иметь Y1 (рис. 6.5).

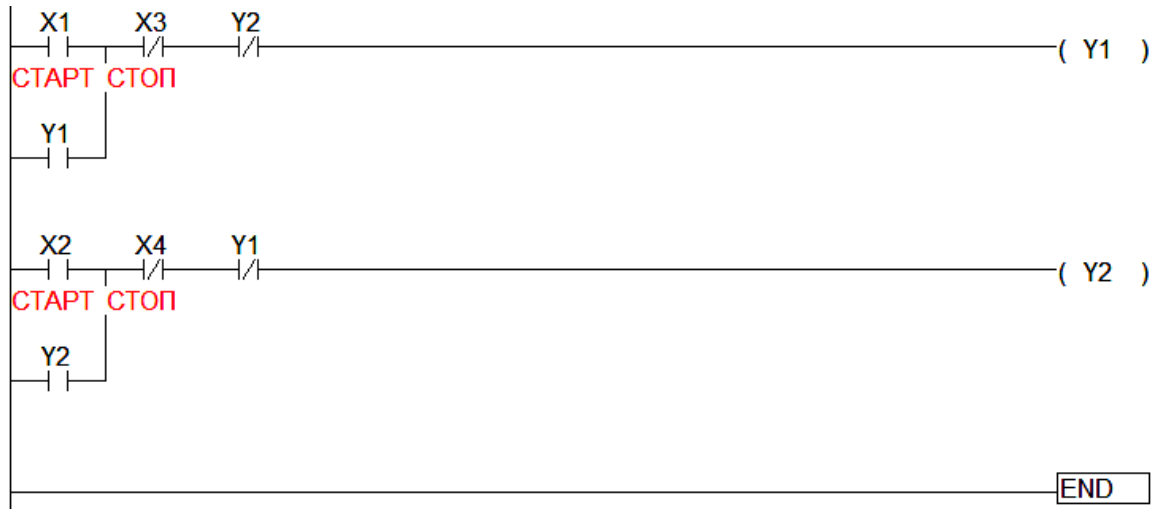


Рис. 6.5. Пример программы для схемы с самоблокировкой

6.6. Схема с последовательным управлением

Выход Y2 может быть включен, только если включен Y1, однако при включении Y2 выход Y1 будет отключен (рис. 6.6).

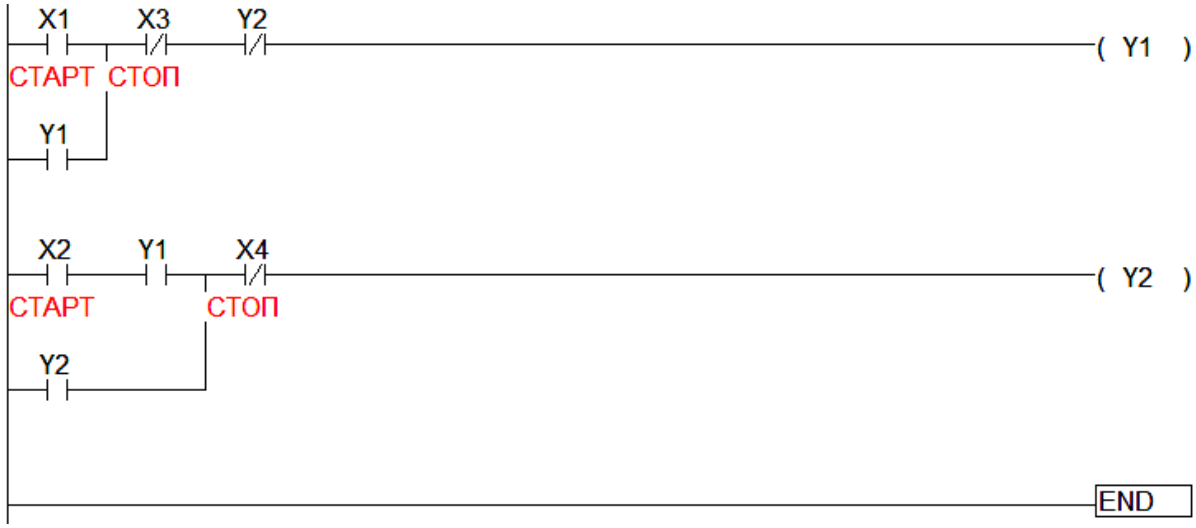


Рис. 6.6. Пример программы для схем
с последовательным управлением

6.7. Мерцающая схема

Используется для мигающей сигнализации с помощью лампы или динамика (рис. 6.7).

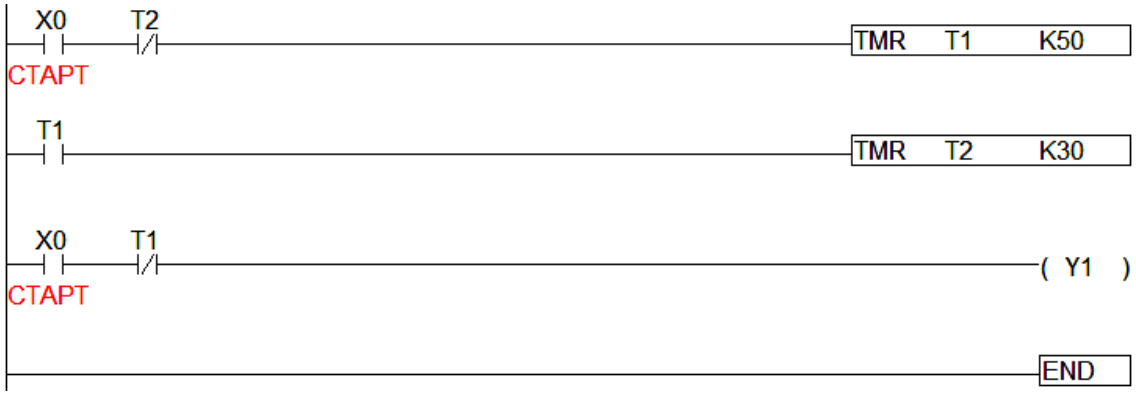


Рис. 6.7. Пример программы для мерцающей схемы

6.8. Триггерная схема

Каждое замыкание контакта X0 изменяет состояние выхода Y1 на противоположное. Эта схема еще называется импульсным реле (рис. 6.8).

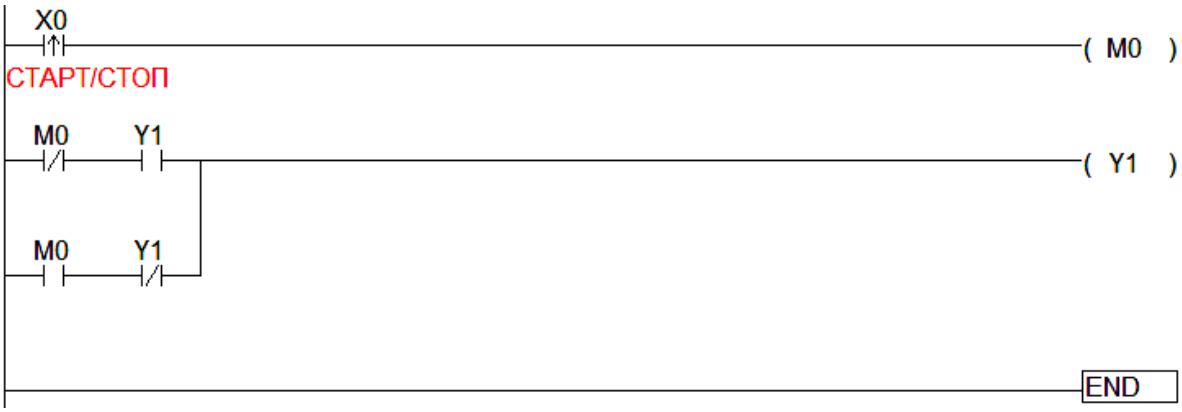


Рис. 6.8. Пример программы для триггерной схемы

Контрольные вопросы

1. Какие виды программ для ПЛК существуют в случаях использования для старта и стопа кнопки без фиксации, но с самоблокировкой выхода?
2. Приведите примеры реализации соответствующих программ.

6.9. Схема с задержкой таймера на выключение

Когда X0 = 1, выход Y1 включен. При выключении X0 выход Y1 выключится через 10 секунд, так как операнд заданного значения таймера T10 имеет дискретность 0,1 с, т.е. K100 = 10 с (рис. 6.9).

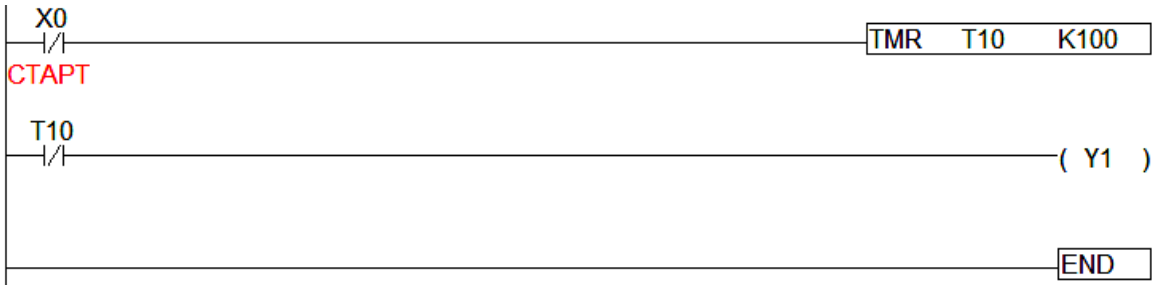


Рис. 6.9. Пример программы для схемы
с задержкой таймера на выключение

6.10. Схема с задержкой таймера на включение и выключение

На рис. 6.10 приведена схема с задержкой таймера на включение и выключение.

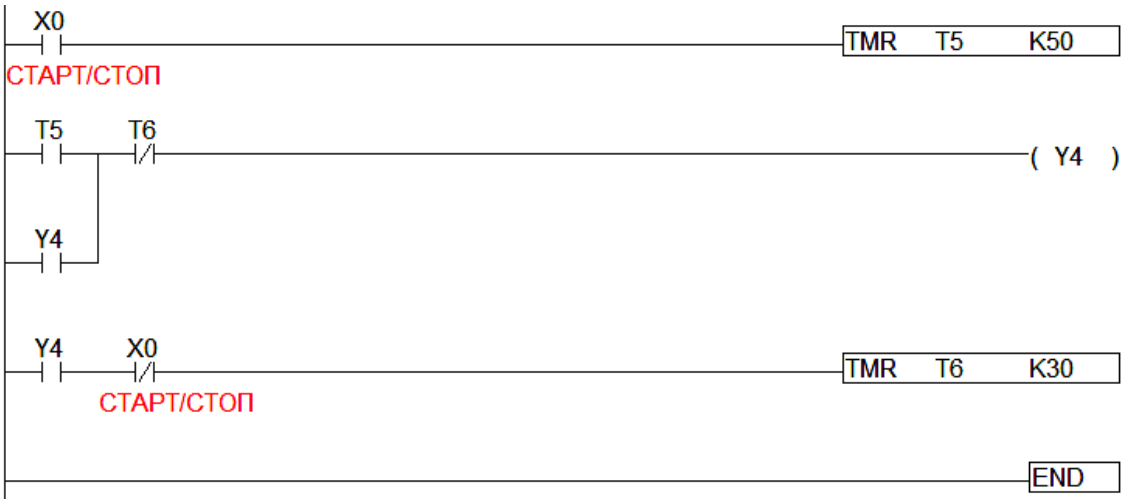


Рис. 6.10. Пример программы для схемы с задержкой таймера
на включение и выключение

7. ПРИМЕРЫ ПРОГРАММИРОВАНИЯ

7.1. Включение освещения с автоматическим отключением

В рабочей зоне автоматизированного оборудования для включения освещения необходимо ограниченное время. После включения освещения его нередко забывают отключить, что приводит к излишнему расходу электроэнергии.

Задача – обеспечить включение освещения на заданный интервал времени, например, в рабочей зоне станка с учетом следующего.

1. Перед защитным ограждением установлен датчик освещения F1, кнопка включения освещения ТАЙМЕР SB1 и тумблер СВЕТ SA1.

2. При кратковременном нажатии на кнопку ТАЙМЕР SB1 и при недостаточном естественном освещении светильник Q1 должен включаться на интервал времени 1 минуту. Этого времени достаточно, чтобы провести измерение детали или осуществить замену заготовки.

3. При удерживании нажатой кнопки ТАЙМЕР SB1 как минимум 3 секунды и при недостаточном естественном освещении светильник должен включаться на интервал времени 3 минуты. Этот режим может быть востребован при наладке или обслуживании станка.

4. С помощью тумблера СВЕТ SA1 включение и отключение освещения производится независимо от внешнего освещения в любой момент времени.

Для решения поставленной задачи принимаются следующие условия.

Выход датчика освещения F1 подключен к входу контроллера X0. При недостаточном освещении датчик формирует на своем выходе единичный логический сигнал. Соответственно при достаточном освещении (или при отсутствии датчика) на входе контроллера X0 будет сигнал логического нуля. Кнопка ТАЙМЕР SB1 подключена к входу X1. Замыкание кнопки ТАЙМЕР SB1 будет приводить к появлению на этом входе логической единицы. Тумблер СВЕТ SA1 подключен к входу X2. Замыкание тумблера СВЕТ SA1 будет приводить к появлению на этом входе логической единицы. Управление освещением (лампой) Q1 осуществляется с помощью контактов релейного выхода Y0.

Программа для приведенного случая представлена на рис. 7.1.

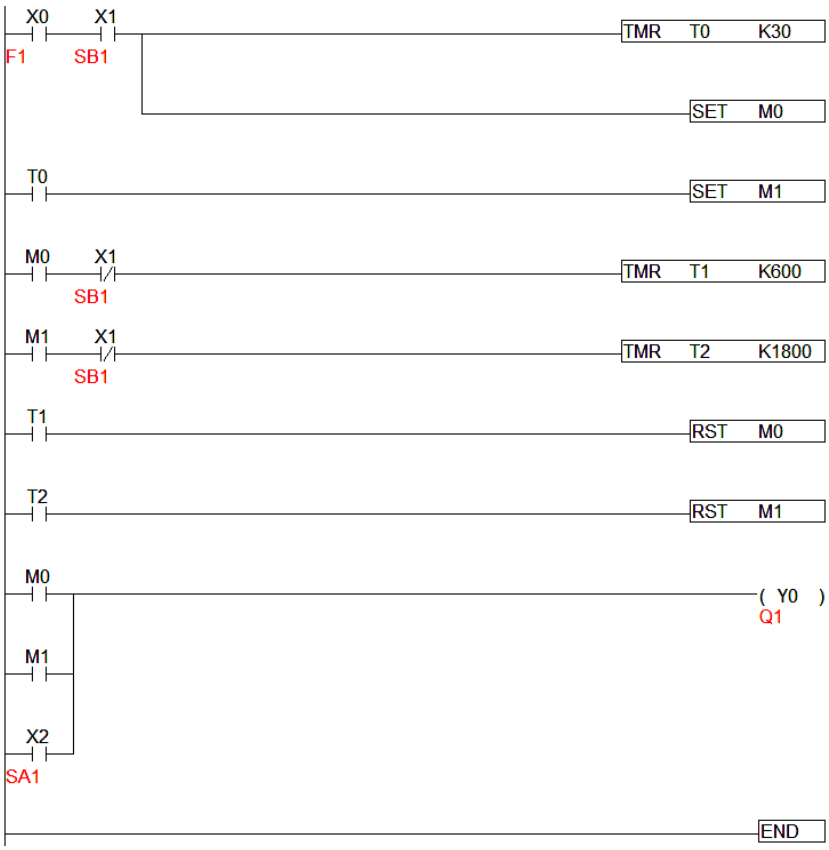


Рис. 7.1. Программа включения освещения
с автоматическим отключением

7.2. Управление уровнем смазочно-охлаждающей жидкости в баке станка

Необходимо реализовать систему управления заполнением бака смазочно-охлаждающей жидкости (СОЖ) из дополнительного бака фильтрации. Выполнение отдельных функций определяется состоянием датчиков уровня, некоторых функций – оператором. Должна быть световая индикация текущего состояния системы.

Алгоритм управления состоит в следующем. Имеется три датчика, определяющих текущий уровень СОЖ в баке гидросистемы: верхний, средний и нижний. Каждый датчик срабатывает (выдает на выходе уровень логической единицы) при превышении СОЖ соответствующего уровня. Ручное управление осуществляется с помощью двух кнопок: СТАРТ и СТОП. Когда бак пуст (уровень СОЖ ниже нижнего датчика уровня), должен постоянно гореть красный индикатор, когда полон (выше верхнего) – постоянно зеленый. Ведется управление двумя насосами. Запуск насосов разрешен, если бак не полон (уровень СОЖ ниже верхнего). Если при нажатии

кнопки СТАРТ уровень СОЖ ниже среднего – запускаются оба насоса, если при нажатии кнопки СТАРТ уровень СОЖ выше среднего – запускается один насос. Включение насосов сопровождается миганием зеленого индикатора. При заполнении бака (уровень СОЖ достигнет верхнего уровня) насосы автоматически выключаются. Если бак пуст (уровень СОЖ ниже нижнего уровня), выключить насосы нажатием кнопки СТОП нельзя.

Для реализации данной задачи ПЛК должен иметь пять дискретных входов и четыре релейных выхода.

Для решения поставленной задачи принимаются следующие условия.

Датчик нижнего уровня СОЖ в баке подключен к входу X1, датчик среднего уровня – к входу X2, датчик верхнего уровня – к входу X3. Кнопка СТОП подключена к входу X4, кнопка СТАРТ – к входу X5. Включением насоса № 1 производится управление с помощью выхода Y1, включением насоса № 2 – с помощью выхода Y2. Индикатор красного цвета подключен к выходу Y3, индикатор зеленого цвета – к выходу Y4.

Ручное управление осуществляется кнопками, которые формируют кратковременные управляющие сигналы.

Программа для приведенного случая представлена на рис. 7.2.

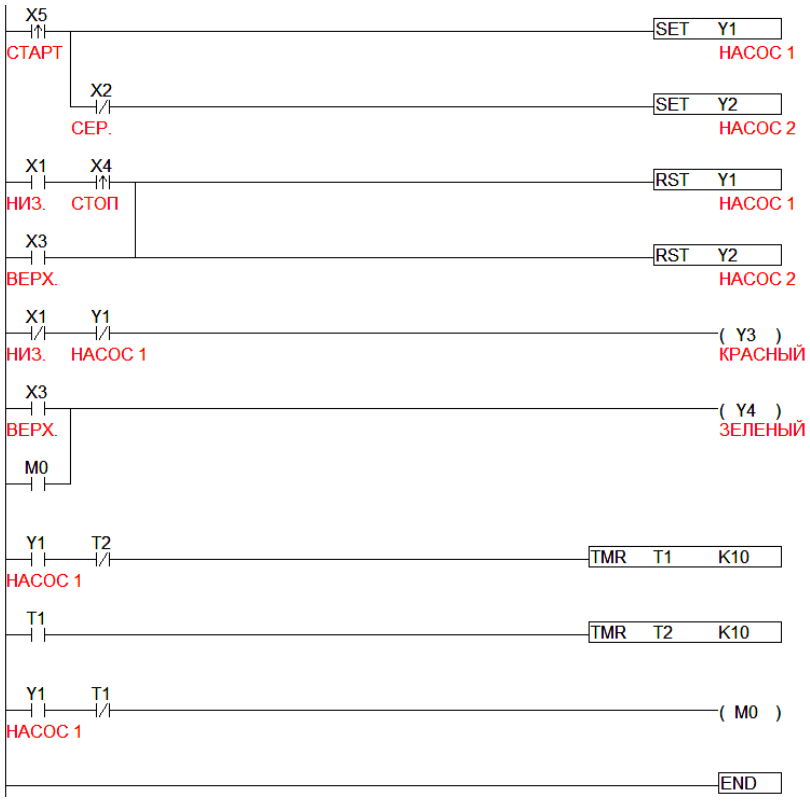


Рис. 7.2. Программа управления уровнем СОЖ в баке станка

7.3. Управление освещением

Контроллер используется для управления освещением у входа в производственное помещение и на лестничных площадках.

Требования. Освещение у входа в складское помещение включается, если сработал сумеречный датчик. Освещение на каждой из трех лестничных площадок срабатывает от датчика движения на соответствующей лестничной площадке. Освещение на каждой лестничной площадке должно гаснуть через 30 с после срабатывания датчика движения. С помощью отдельного выключателя освещение включается везде, например для проверки.

Используемые компоненты:

X1 – датчик движения на первой лестничной площадке;
X2 – датчик движения на второй лестничной площадке;
X3 – датчик движения на третьей лестничной площадке;
X4 – сумеречное реле;
X5 – ручное включение освещения;
Y1 – освещение на первой лестничной площадке;
Y2 – освещение на второй лестничной площадке;
Y3 – освещение на третьей лестничной площадке;
Y4 – освещение у входа.

Программа для приведенного случая представлена на рис. 7.3.

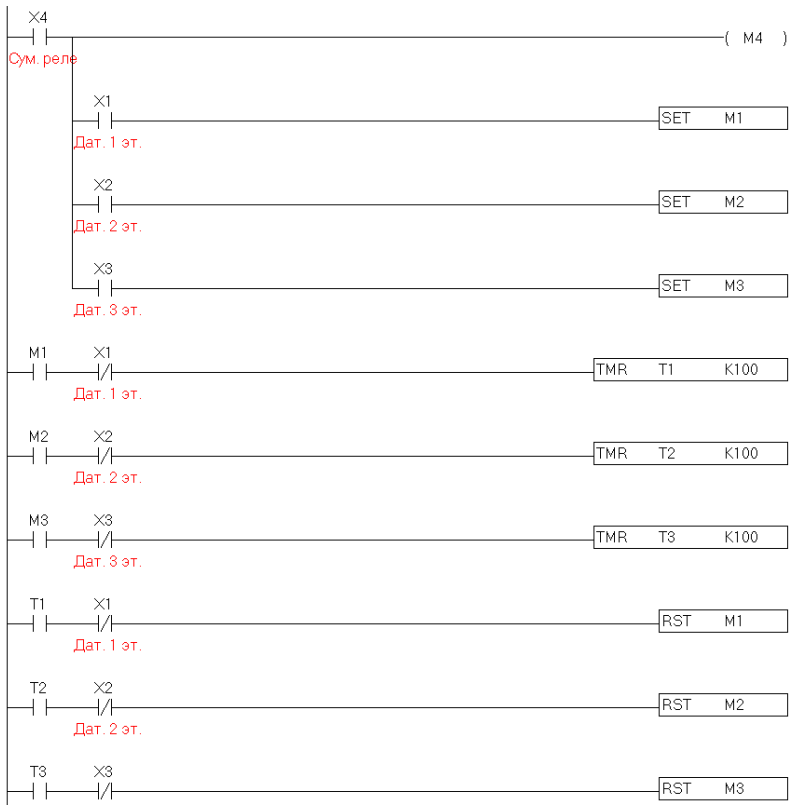


Рис. 7.3. Программа управления освещением (начало)

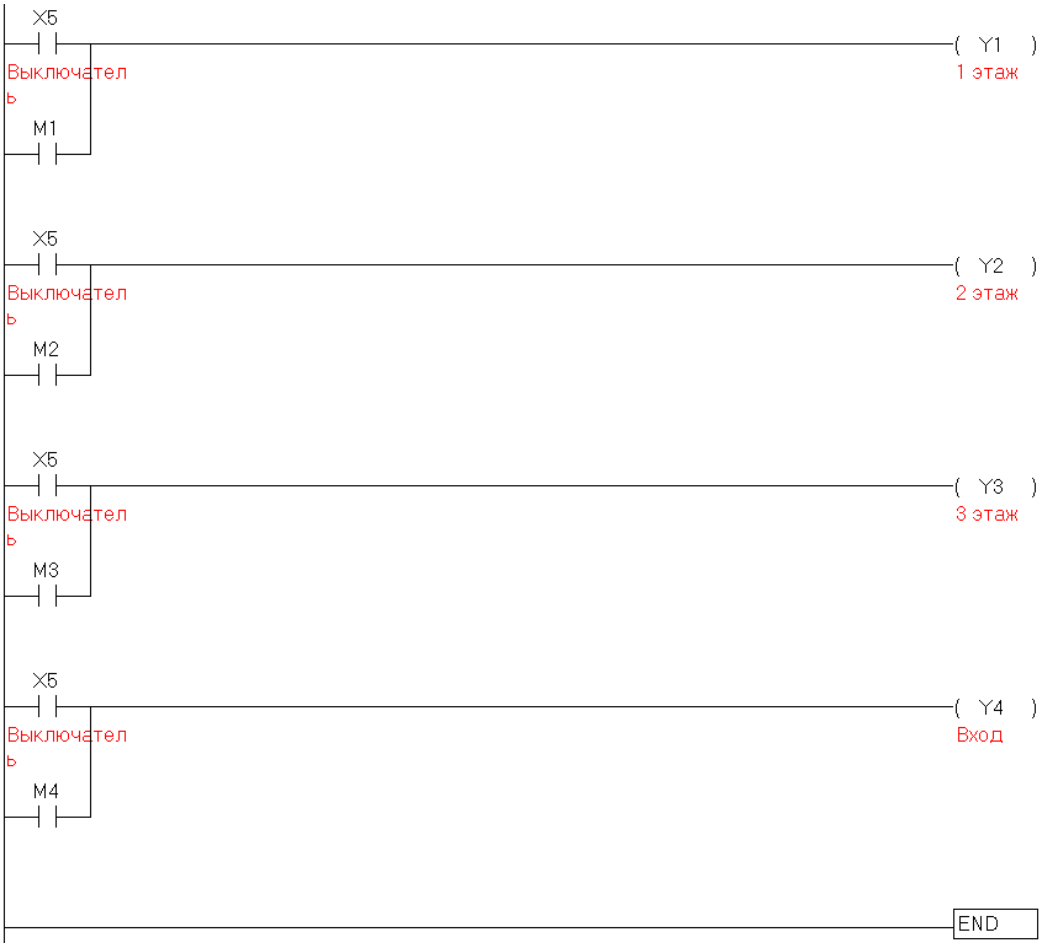


Рис. 7.3. Программа управления освещением (окончание)

7.4. Ступенчатый выключатель вентилятора

С помощью контроллера должно быть реализовано ступенчатое переключение четырех уровней мощности вентилятора промышленной вытяжки.

Требования. С помощью кнопочного переключателя «Увеличение» вентилятор запускается на уровне 1. Каждым следующим нажатием кнопки вентилятор переключается на один уровень мощности выше. Всего предусмотрено четыре уровня мощности. Этот 4-ступенчатый переключатель реализуется с помощью прикладной инструкции для кольцевого сдвига влево/вправо (ROL/ROR). Каждому уровню мощности соответствует срабатывание только одного соответствующего выхода. Каждым нажатием кнопки «Уменьшение» вентилятор может ступень за ступенью переключаться в обратном направлении.

Используемые компоненты:

X1 – кнопка «Увеличение»;

X2 – кнопка «Уменьшение»;
Y1 – контактор уровня 1;
Y2 – контактор уровня 2;
Y3 – контактор уровня 3;
Y4 – контактор уровня 4.
Программа для приведенного случая представлена на рис. 7.4.

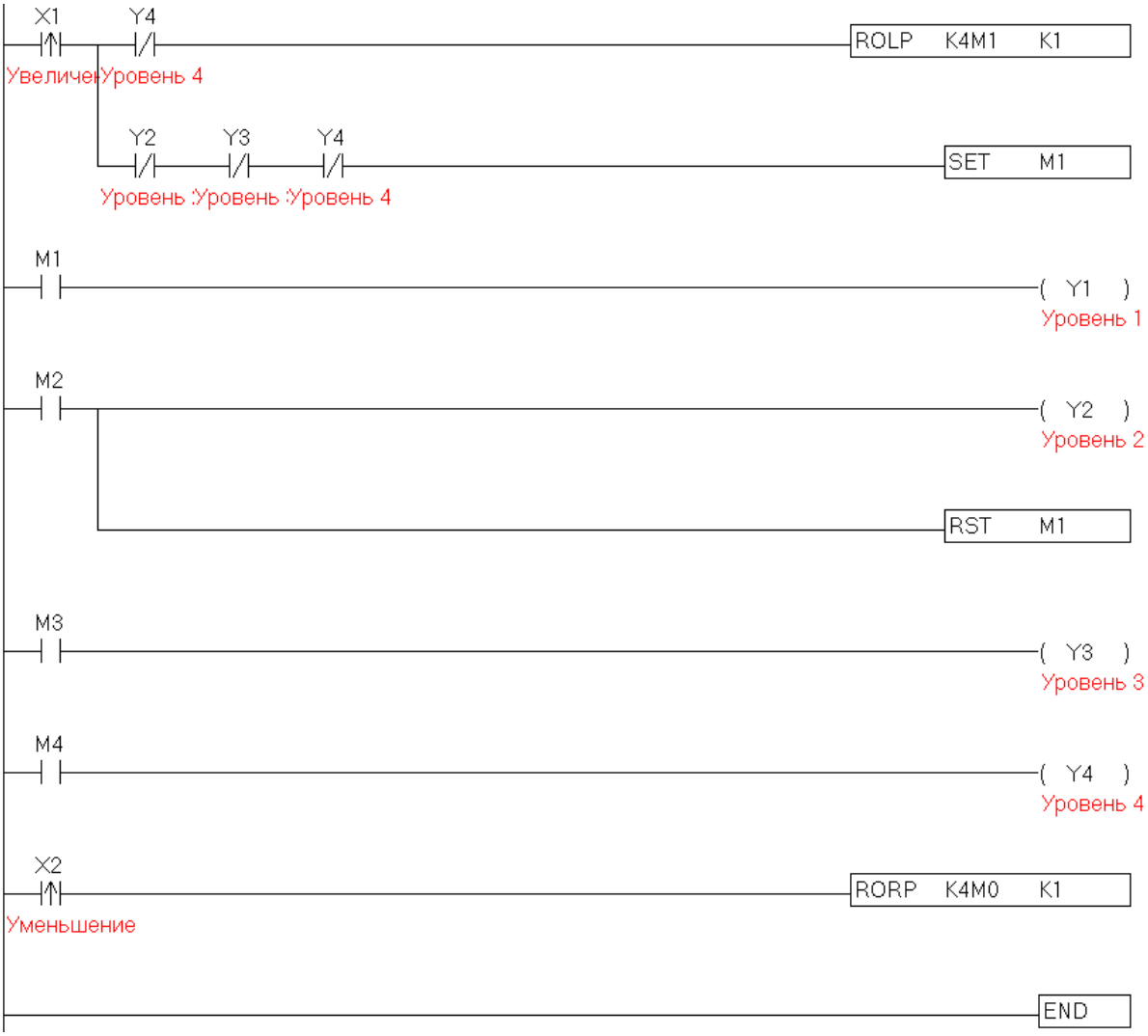


Рис. 7.4. Программа управления вентилятором промышленной вытяжки

7.5. Контроль длительности использования инструмента

С помощью контроллера должно быть обеспечено то, чтобы каждый инструмент станка мог работать только определенный интервал времени. Если заданное время превышено, то контроллер ав-

томатически запрещает работу инструмента. Это, например, очень важно для контроля и замены инструмента после превышения периода его стойкости.

Контроллер отдельно контролирует длительность суммарного времени включения каждого из трех инструментов (нахождение в рабочей позиции). Для каждого из инструментов может быть задано различное суммарное время работы.

Срабатывание тумблера на одном из трех дискретных входов определяет необходимость включения нужного инструмента с помощью соответствующего выхода и необходимость подсчета времени включения. Число включений каждого инструмента не ограничивается. Если суммарное время работы инструмента превышено, то контроллер его автоматически отключает. Длительность суммарного времени включения задается в секундах. Отсчет времени ведется за счет формирования с помощью датчика тактов (M1013) ежесекундных импульсов и их подсчета. Чтобы дать сигнал о том, что время включения какого-то из инструментов скоро истечет, к выходам контроллера подключены 3 сигнальные лампы, которые начинают мигать за 100 секунд до истечения времени соответствующего инструмента. Оператор нажатием кнопки «Сброс» может одновременно обнулить счетчики суммарного времени включения всех инструментов.

Используемые компоненты:

M1013 – датчик тактов с периодом 1 с (ON = 0,5 с, OFF = 0,5 с);

X1 – выключатель для инструмента 1;

X2 – выключатель для инструмента 2;

X3 – выключатель для инструмента 3;

X4 – кнопка «Сброс»;

Y1 – инструмент 1;

Y2 – инструмент 2;

Y3 – инструмент 3;

Y4 – сигнальная лампа инструмента 1;

Y5 – сигнальная лампа инструмента 2;

Y6 – сигнальная лампа инструмента 3.

Программа для приведенного случая представлена на рис. 7.5.

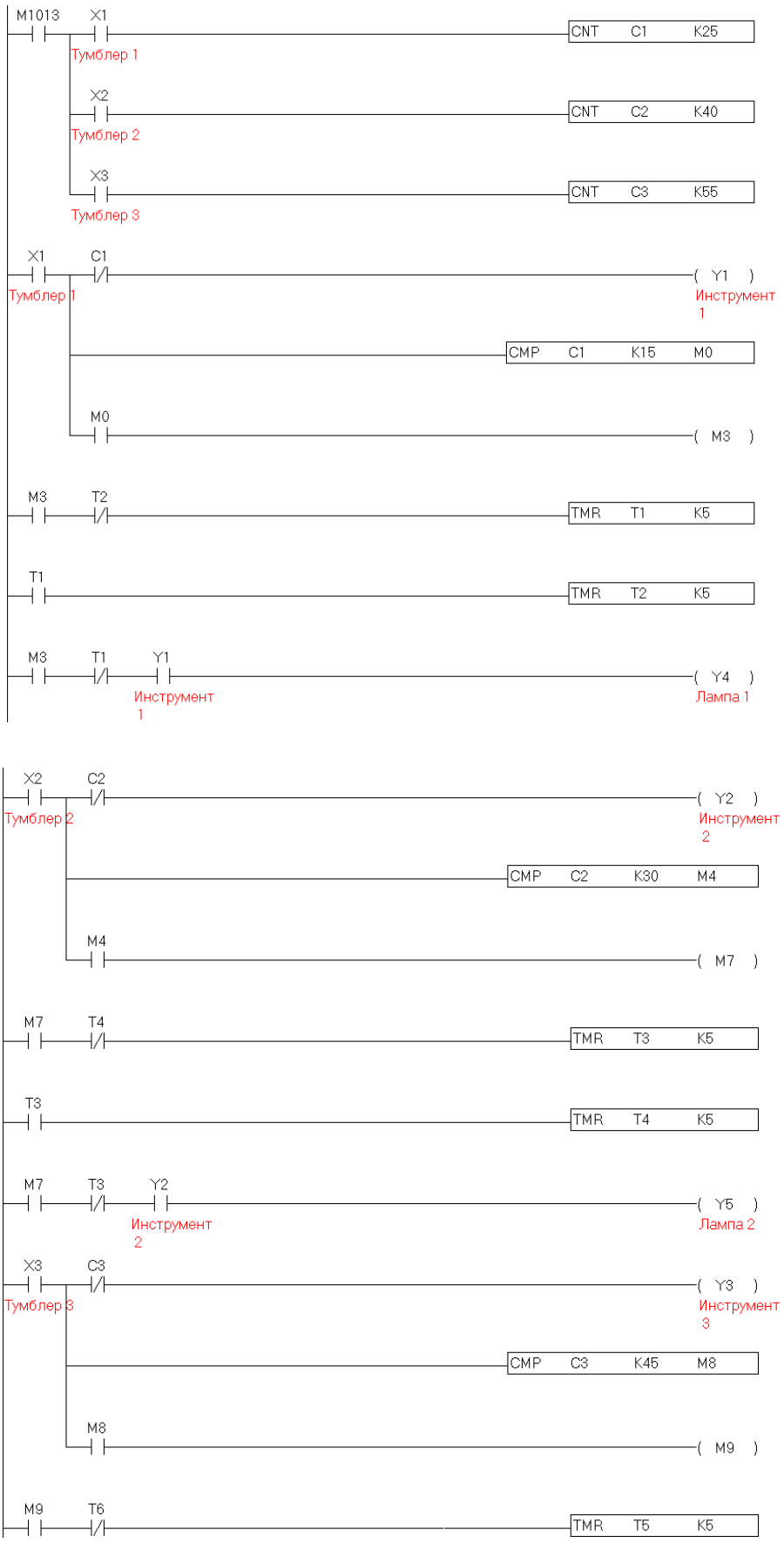


Рис. 7.5. Программа контроля длительности использования инструмента (начало)

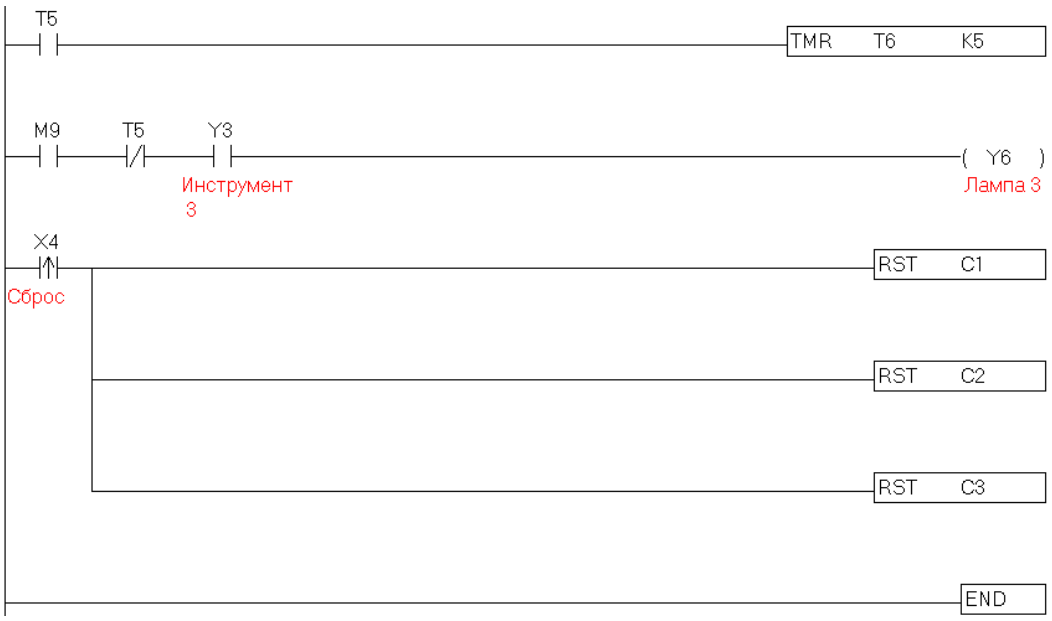


Рис. 7.5. Программа контроля длительности использования инструмента (окончание)

7.6. Управление системой подачи защитного газа автомата для наплавки

Защитный газ изолирует расплавленный металл от газов в воздухе, препятствуя их взаимодействию. Контроллер берет на себя функции управления системой подачи защитного газа автомата.

В режиме управления подача защитного газа осуществляется нажатием кнопки «Пуск». Подача защитного газа сопровождается горением индикатора. Прекращение подачи защитного газа осуществляется нажатием кнопки «Стоп». С момента нажатия кнопки «Стоп» начинается 3-минутный отсчет времени, который сопровождается миганием индикатора. С момента включения подачи газа до момента окончания времени удаления защитного газа включается блокировка открытия двери камеры. При отключении блокировки двери камеры индикатор гаснет.

Предусмотрена аварийная остановка процесса наплавки, которая может быть осуществлена в любой момент нажатием кнопки «Авария». По нажатию этой кнопки все действия системы подачи газа прекращаются (подача газа прекращается, время отсчета всех периодов времени сбрасывается в нуль, блокировка двери снимается).

Используемые компоненты:

X1 – кнопка «Пуск»;

X2 – кнопка «Стоп»;

X3 – кнопка «Авария»;
Y1 – включение газогенератора;
Y2 – блокировка двери;
Y3 – индикатор режима работы.
Программа для приведенного случая представлена на рис. 7.6.

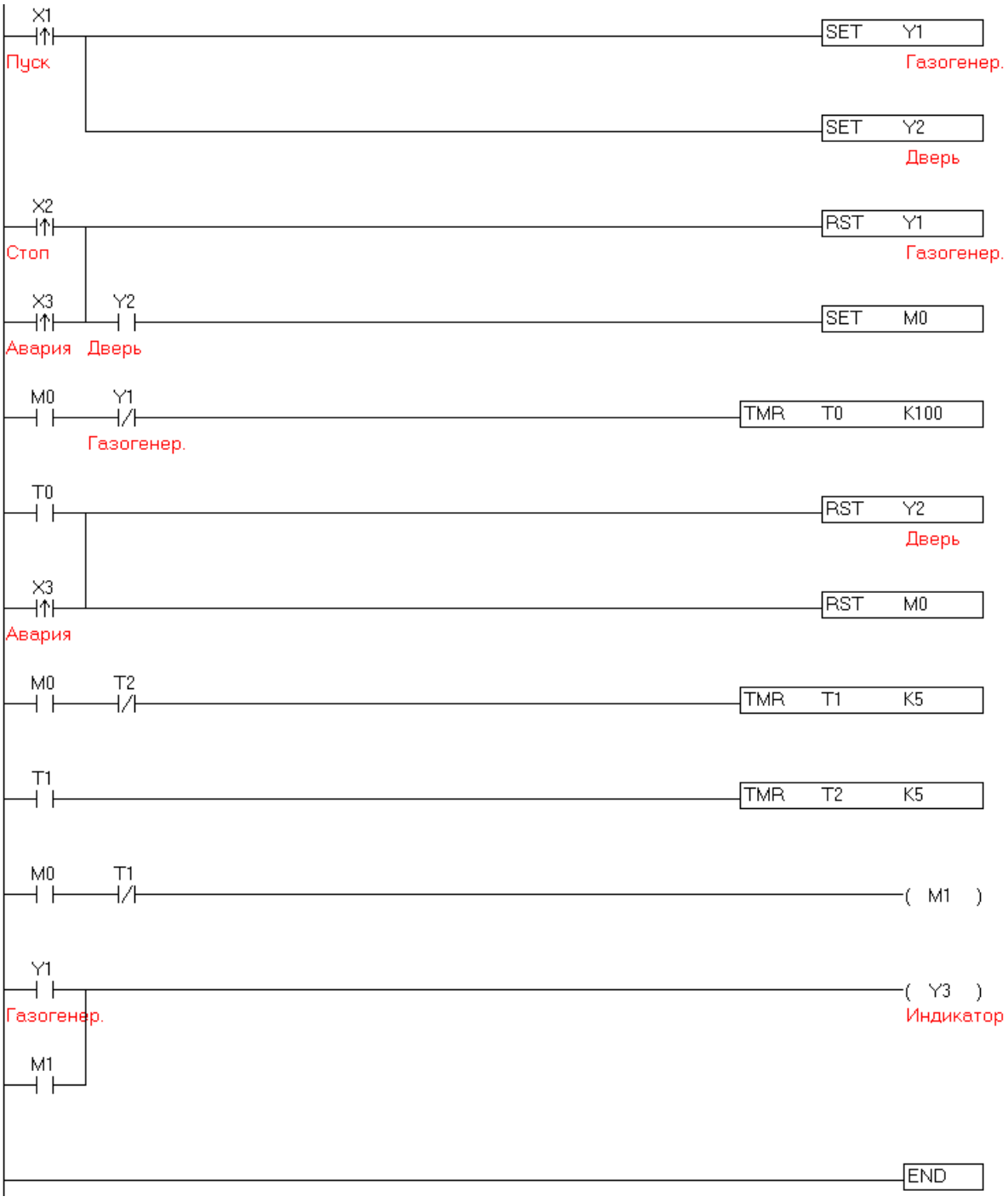


Рис. 7.6. Программа управления системой подачи защитного газа автомата для наплавки

7.7. Управление подъемным столом-перегружателем

С помощью контроллера реализовано управление подъемным столом-перегружателем. Для контроля зоны перемещения подъемного стола установлен ультразвуковой датчик.

Подъемный стол с помощью кнопок «Вверх» и «Вниз» может перемещаться вверх или вниз. Верхнее и нижнее конечное положение распознается соответствующим концевым датчиком. Если конечное положение достигнуто, то двигаться можно только в противоположном направлении. Стол останавливается в любой момент времени при срабатывании датчика «Аварийный останов». Стол останавливается если ультразвуковой датчик контроля зоны перемещения стола «Препятствие» распознает препятствие. При сработавшем датчике «Препятствие» стол можно перемещать в ручном режиме, если клавиша направления нажата дольше 3 секунд. Однако если одновременно сработал датчик «Аварийный останов», то стол перемещаться с помощью кнопок направления не может. Если стол движется вверх или вниз, то мигает предупредительный световой сигнал.

Используемые компоненты:

X1 – кнопка «Вверх»;

X2 – верхний концевой датчик;

X3 – кнопка «Вниз»;

X4 – нижний концевой датчик;

X5 – датчик «Аварийный останов»;

X6 – датчик «Препятствие»;

Y1 – стол двигается вверх;

Y2 – стол двигается вниз;

Y3 – предупреждающий световой сигнал.

Программа для приведенного случая представлена на рис. 7.7.

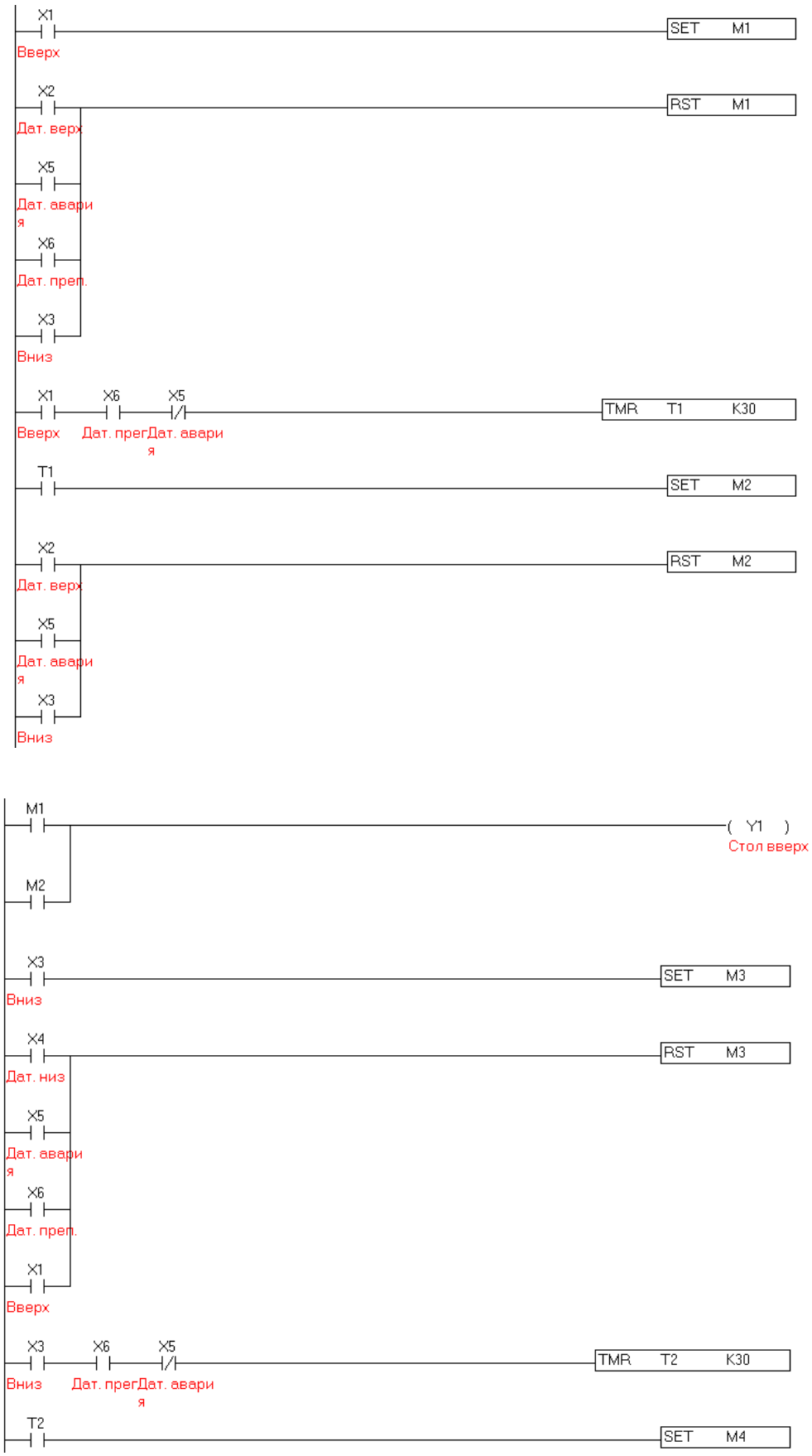


Рис. 7.7. Программа управления подъемным столом-перегружателем (начало)

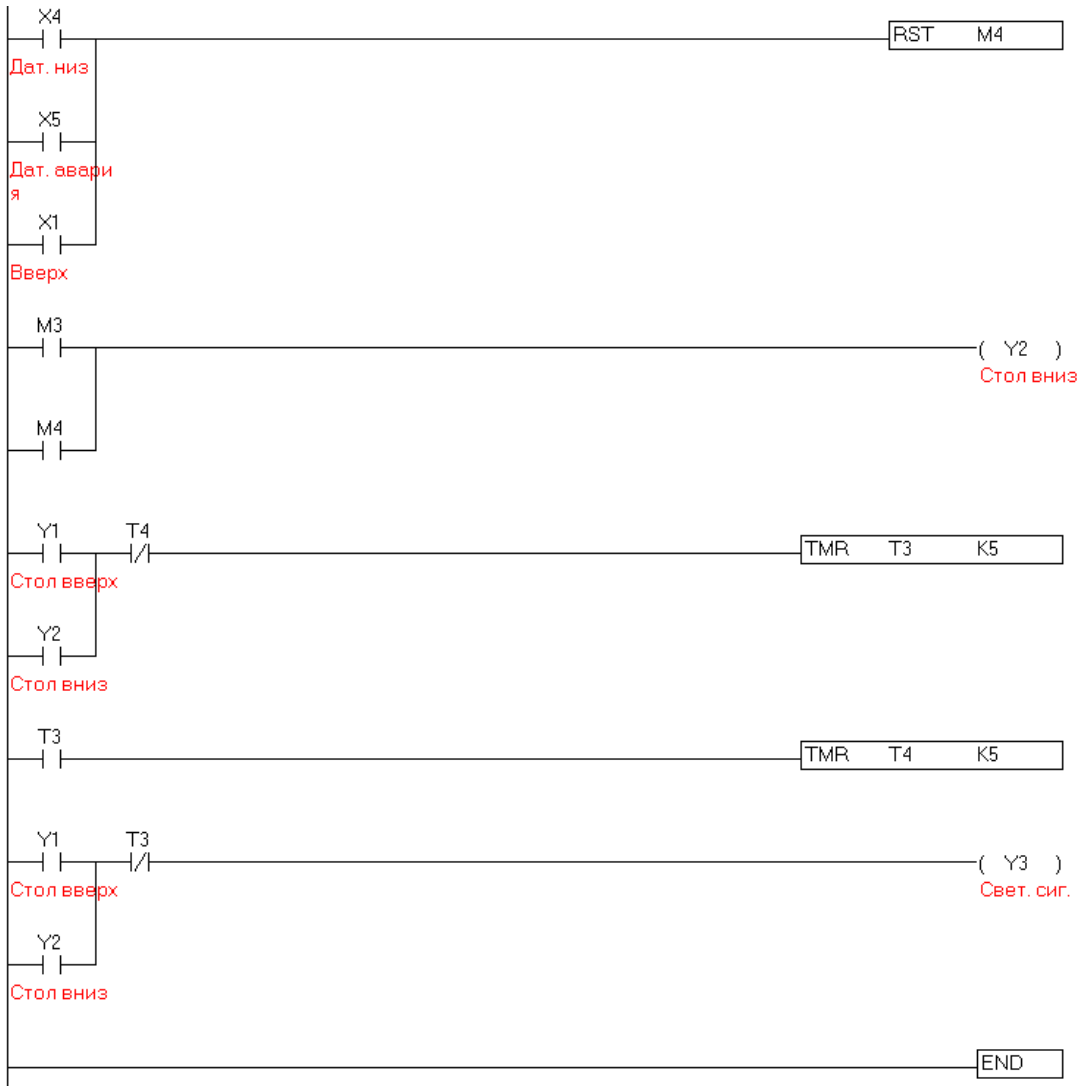


Рис. 7.7. Программа управления подъемным столом-перегружателем (окончание)

7.8. Последовательный запуск трех двигателей

С помощью контроллера необходимо реализовать последовательный пуск трех двигателей.

Запуск двигателя масляного насоса осуществляется сразу при нажатии кнопки «Пуск». Основной двигатель запускается после 10-секундной задержки. После пуска основного двигателя спустя 5 секунд запускается вспомогательный двигатель подачи СОЖ. При нажатии кнопки «Стоп» останавливаются все двигатели и сбрасываются таймеры.

Используемые компоненты:

X0 – кнопка «Пуск»;

X1 – кнопка «Стоп»;

Y0 – двигатель масляного насоса;
Y1 – основной двигатель;
Y2 – вспомогательный двигатель.
Программа для приведенного случая представлена на рис. 7.8.

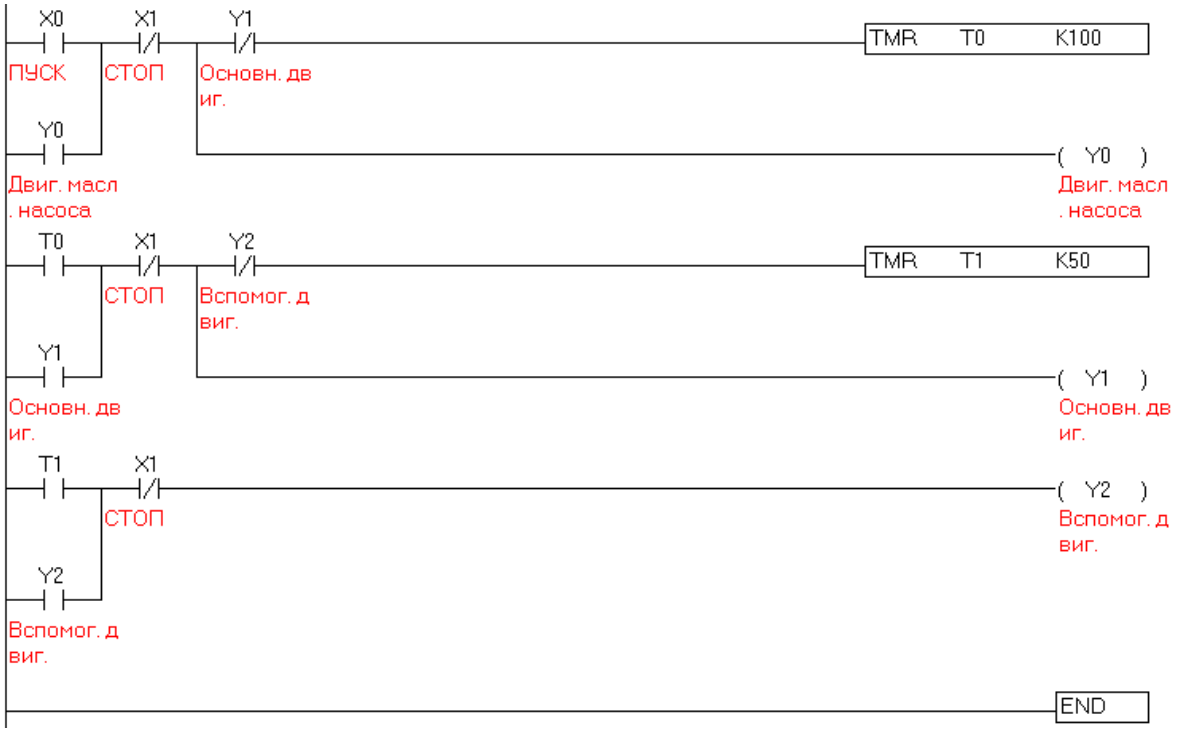


Рис. 7.8. Программа последовательного запуска трех двигателей

7.9. Педальный переключатель для выбора скоростей станка

На рабочем месте у станка с помощью педального переключателя можно выбирать различные режимы (скорости вращения двигателя) или отключать станок. Управление осуществляется с помощью контроллера.

Механический педальный переключатель ногой может переводиться в три положения: «Педаль отжата», «Педаль в среднем положении», «Педаль полностью нажата». Два последних положения фиксируются датчиками. При подаче питания на станок скорость вращения равна нулю. Если перевести педаль в среднее положение более чем на 2 секунды станок включится на номинальную скорость. Если педаль отпустить и опять перевести в среднее положение более чем на 2 секунды, то включится повышенная скорость, если педаль отпустить и опять перевести в среднее положение более чем на 2 се-

кунды, то включится номинальная скорость – таким образом можно последовательно переводить скорость станка из номинальной скорости на повышенную и наоборот. Но если педаль нажать полностью – станок остановится.

Если станок работает на повышенной скорости, то это отображается с помощью сигнальной лампы.

Используемые компоненты:

X1 – датчик «Педаль в среднем положении»;

X2 – датчик «Педаль полностью нажата»;

Y1 – 1-я скорость двигателя;

Y2 – 2-я скорость двигателя;

Y3 – лампа, сигнализирующая о работе на повышенной скорости.

Программа для приведенного случая представлена на рис. 7.9.

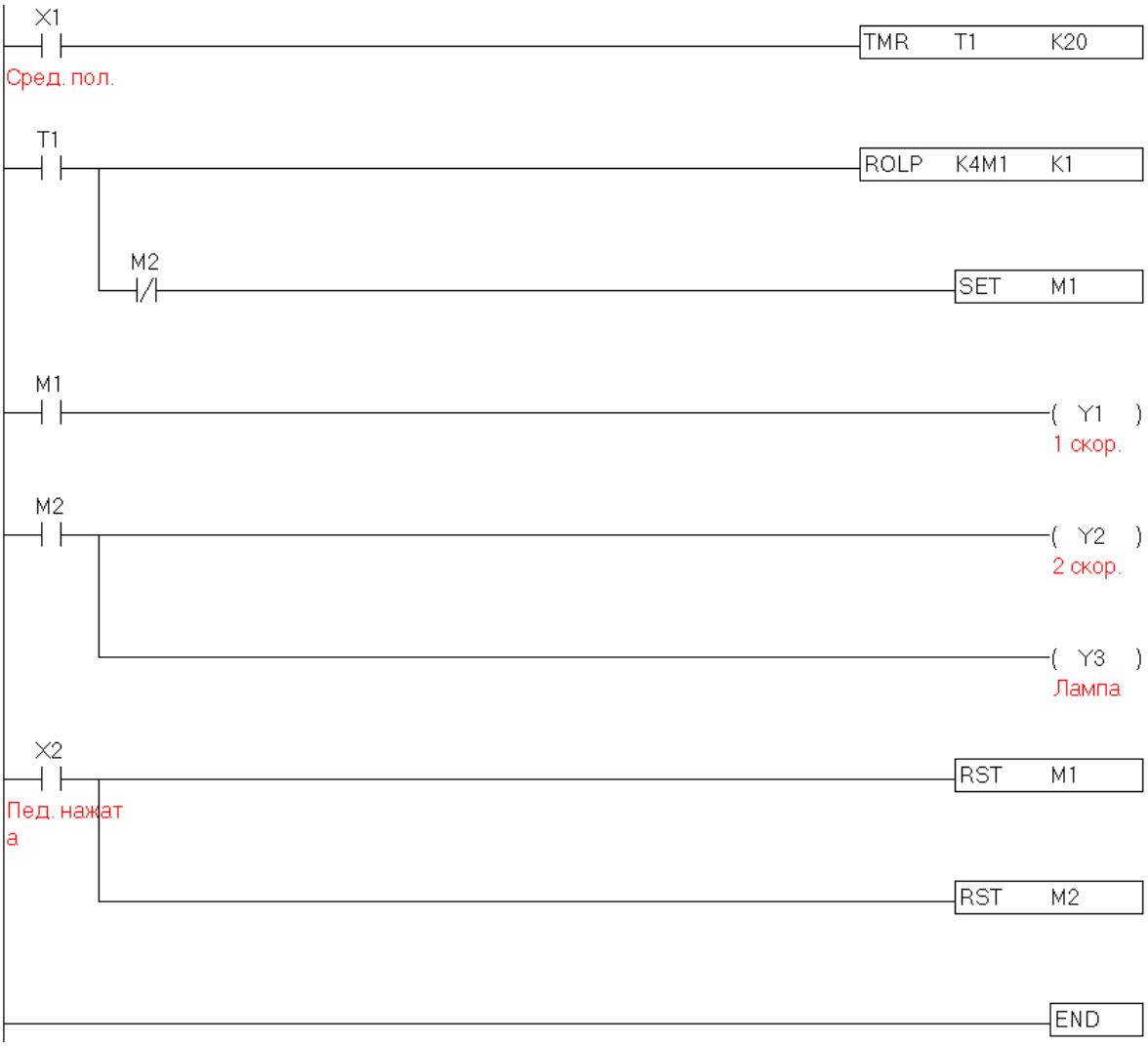


Рис. 7.9. Программа управления выбором скоростей станка

7.10. Цикловое управление приводом продольного перемещения суппорта токарного станка

Преобразователь частоты со встроенным контроллером управляет приводом продольного перемещения суппорта токарного станка, который совершает возвратно-поступательное движение.

Перемещение суппорта на скорости 15 Гц начинается по кратковременному нажатию кнопки «Пуск/Стоп», останов – по срабатыванию концевого выключателя КВ1. Затем привод стоит в течение 2 секунд и быстро возвращает инструмент (реверс на частоте 30 Гц), останов – по срабатыванию концевого выключателя КВ2. Прервать цикл работы возможно в любой момент времени кратковременным нажатием кнопки «Пуск/Стоп».

Используемые компоненты:

X0 – кнопка «Пуск/Стоп»;

X1 – концевой выключатель КВ1;

X2 – концевой выключатель КВ2.

Программа для приведенного случая представлена на рис. 7.10.

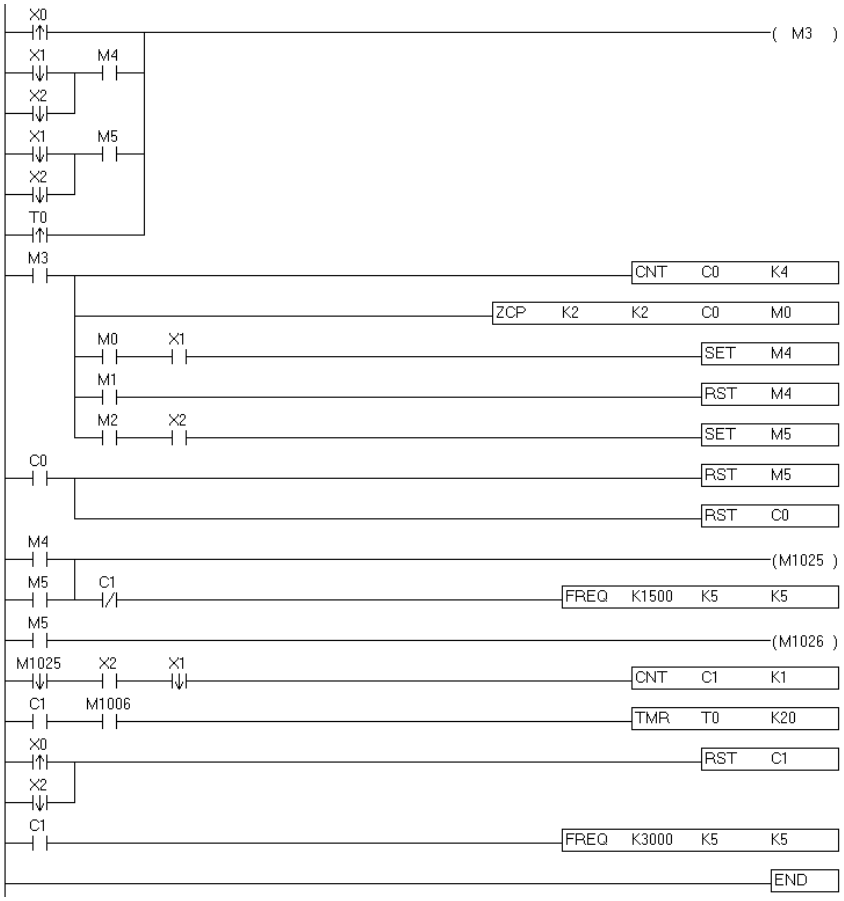


Рис. 7.10. Программа управления приводом продольного перемещения суппорта токарного станка

Процесс программирования включает также следующие действия:

1. Преподаватель выдает задание по аналогии с приведенным в соответствующем примере программирования.

2. Студент составляет программу.

3. Используя режим эмуляции среды программирования WPLSoft, следует убедиться в том, что программа составлена в соответствии с исходными данными.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Программируемые логические контроллеры SS2/SA2/SX2/ ES2/EX2/SE. Руководство по программированию // ООО «СТОИК ЛТД». – URL: http://www.deltronics.ru/images/manual/DVP_PrM_RU_ [2013].pdf (дата обращения: 23.10.2018).

2. Редактор WPL-Soft. Быстрый старт // ОАО «СКБ ИС». – URL: http://www.skbis.ru/pdf/sppu/WPLSoft_qg.pdf (дата обращения: 23.10.2018).

3. Преобразователи частоты серии VFD-E. Руководство по эксплуатации // ООО «СТОИК ЛТД». – URL: http://www.deltronics.ru/ images/manual/VFD-E_UM_RU_[092016].pdf (дата обращения: 23.10.2018).

4. Шишов, О. В. Элементы систем автоматизации. Релейные контроллеры : лабораторный практикум / О. В. Шишов. – Москва ; Берлин : Директ-Медиа, 2015. – 159 с.

5. Петров, И. В. Программируемые контроллеры. Стандартные языки и приемы прикладного проектирования / И. В. Петров. – Москва : Солон-Пресс, 2004. – 256 с.

6. ГОСТ 51841 – 2001. Программируемые контроллеры. Общие положения и функциональные характеристики.

7. Парр, Э. Программируемые контроллеры. Руководство для инженера / Э. Парр ; пер. с англ. – 3-е изд. – Москва : Бином Библиотека знаний, 2009. – 516 с.

8. Максимычев, О. И. Программирование логических контроллеров (PLC) : учеб. пособие / О. И. Максимычев, А. В. Либенко, В. А. Виноградов. – Москва : МАДИ, 2016. – 188 с.

Учебное издание

Тарасов Дмитрий Александрович,
Липов Александр Викторович,
Большаков Герман Сергеевич

МОДЕЛИРОВАНИЕ РАБОТЫ ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ КОНТРОЛЛЕРОВ Delta Electronics В РЕДАКТОРЕ WPLSoft

Редактор *Т. В. Веденеева*
Технический редактор *Ю. В. Анурова*
Компьютерная верстка *Ю. В. Ануровой*
Дизайн обложки *А. А. Стаценко*

Подписано в печать 27.06.2019.
Формат $60\times84^{1}/_{16}$. Усл. печ. л. 5,81.
Заказ № 219. Тираж 38.

Издательство ПГУ
440026, Пенза, Красная, 40.
Тел./факс: (8412) 56-47-33; e-mail: iic@pnzgu.ru

Вниманию авторов!

Издательство ПГУ выпускает учебную, научную и художественную литературу, презентационную и акцидентную продукцию, а также полноцветные юбилейные и мемориальные издания в соответствии с ГОСТ 7.60–2003.

Издательство ПГУ принимает к изданию рукописи, подготовленные с использованием текстового редактора Microsoft Word for Windows версий **2003** и выше. Формат – А4, основной шрифт – Times New Roman, 14–16 pt через одинарный интервал (минимальный размер шрифта в таблицах и сносках – 12,5 pt). Тип файла в электронном виде – doc, docx.

Работа должна содержать индекс УДК, аннотацию.

Аннотация (ГОСТ 7.86–2003, ГОСТ 7.9–1995) включает характеристику основной темы, проблемы объекта, цели работы и ее результаты. В аннотации указывают, что нового несет в себе данный документ в сравнении с другими, родственными по тематике и целевому назначению. Аннотация может включать сведения о достоинствах произведения. Текст аннотации начинают фразой, в которой сформулирована главная тема документа. Заканчивается аннотация читательским адресом.

Рисунки и таблицы должны быть размещены в тексте после ссылки на них (растровые рисунки представляются в виде отдельных файлов в формате jpg, BMP с разрешением 300 dpi, векторные рисунки – в формате Corel Draw с минимальной толщиной линии 0,75 pt. (Рисунки должны быть доступны для правки!). Рисунки должны сопровождаться подрисуночными подписями, на все рисунки и таблицы в тексте должны быть ссылки.

Формулы в тексте выполняются только в редакторе формул **MathType версия 5.0** и выше. Символы греческого и русского алфавита должны быть набраны прямо, нежирно; латинского – курсивом, нежирно; обозначения векторов и матриц – прямо, жирно; цифры – прямо, нежирно. Наименования химических элементов набираются прямо, нежирно. Эти же требования необходимо соблюдать и в рисунках.

В списке литературы **нумерация источников** должна соответствовать очередности ссылок на них в тексте ([1], [2], …). Номер источника указывается в квадратных скобках. Требования к оформлению списка литературы на русские и иностранные источники (ГОСТ 7.1–2003): для книг – фамилия и инициалы автора, название, город, издательство, год издания, том, количество страниц; для журнальных статей, сборников трудов – фамилия и инициалы автора, название статьи, полное название журнала или сборника, серия, год, том, номер, страницы; для материалов конференций – фамилия и инициалы автора, название статьи, название конференции, город, издательство, год, страницы.

К материалам **должна** прилагаться следующая информация: фамилия, имя, отчество, контактные телефоны.

Контакты Издательства ПГУ: (8412) 56-47-33, 36-84-91. E-mail: iic@pnzgu.ru