

СПИСОК ПРИНЯТЫХ СОКРАЩЕНИЙ

ПК	персональный компьютер
ПЛК	программируемый логический контроллер
АИО	автоматический исследуемый объект
ИИС	измерительная информационная система
СИ	средство измерения
ПО	программное обеспечение
АУО	регулируемый преобразователь напряжения (РПН)
МЭК	международная электротехническая комиссия
ЦЭО	цифровой электронный осциллограф

ОБЩИЕ СВЕДЕНИЯ

В различных отраслях промышленности для управления технологическими процессами используются программируемые логические контроллеры (ПЛК). Контроллер – устройство управления в электронике и вычислительной технике. На основе контроллеров возможна автоматизация технологических процессов. Автоматизация – совокупность методических, технических и программируемых средств, обеспечивающих проведение измерений и управление без участия человека.

Целями автоматизации являются: повышение качества продукции за счет повторяемости операций, увеличения числа измерений и получения более точных данных о свойствах изделия или процессов, а также увеличение надежности изделий за счет более полных данных о процессах. Экономическая эффективность при автоматизации обеспечивается за счет экономии трудовых ресурсов, путем замены труда человека трудом машины. Достигается сокращение затрат в промышленности за счет уменьшения трудоемкости работ, а также повышается производительность труда.


Лабораторный практикум состоит из предисловия, общих теоретических сведений о системах программирования и лабораторных работ, выполняемых с использованием ПЛК-150, а также библиографического списка. Лабораторный практикум направлен на изучение студентами применения ПЛК-150 для управления технологическими операциями.

Лабораторный практикум может быть использован для подготовки студентов магистратуры по направлению 27.04.01 «Стандартизация и метрология» магистерской программы «Метрологическое и нормативно-техническое обеспечение процессов и производств», дисциплины «Автоматизация контроля технологических процессов», а также при обучении студентов магистратуры по магистерской программе «Метрологическое обеспечение контроля качества свойств и состава веществ, материала и изделий», дисциплины «Автоматизированные системы и измерительные комплексы».

КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Система программирования CoDeSys V3.x

Основные сведения. Система программирования CoDeSys была изготовлена компанией 3S-Smart Software Solutions GmbH. CoDeSys – это аппаратно-независимая система программирования ПЛК. Она полностью отвечает требованиям МЭК 61131-3, позволяет использовать процедуры на С и поддерживает объектно-ориентированное программирование. В сочетании с системой исполнения CoDeSys SP возможна работа с несколькими устройствами и приложениями. Благодаря структурному разбиению на компоненты, пользователь сам может конфигурировать и расширять пользовательский интерфейс.

Установка и пуск. Для установки программы запустите файл *Setup_CoDeSysV<Version>.exe*. Далее следуйте указаниям мастера установки. После установки CoDeSys, на рабочем столе появится иконка . Запуск CoDeSys осуществляется из меню *Пуск*; путь по умолчанию Programs – 3S CoDeSys – CoDeSys – CoDeSys V<version>.

Компоненты организации программ (POU). Компоненты создают под прикладное программное обеспечение ПЛК. Компоненты организации программ POU (Program Organization Unit) содержат функции, функциональные блоки и программы. Компонент выступает как «черный ящик», внутреннее устройство и содержание которого знать не нужно. В графическом изображении он представлен прямоугольником с входами (слева) и выходами (справа).

Выбор нужного POU производится в окне объявлений (рис. 1) в строках «Программа», «Функциональный блок» или «Функция». Для LD будем использовать только «Программа», так как нам потребуются только стандартные компоненты (контакты, катушки реле, FB).

Запуск CoDeSys. Произведем первый запуск среды CoDeSys. В окне Target Settings напротив строки Configuration выбираем тип логического контроллера PLC 100.R-L, поскольку именно он используется в лабораторной работе, и нажимаем ОК. В появившемся окне «Новый программный компонент (POU)» (рис. 1) выбираем тип POU «Программа и язык», на котором будет осуществляться написание программы LD. Имя программы оставляем без изменения. Подтвер-

ждаем выбор нажатием на кнопку ОК. После выполнения всех вышеописанных действий откроется главное окно (рис. 2) среды CoDeSys.

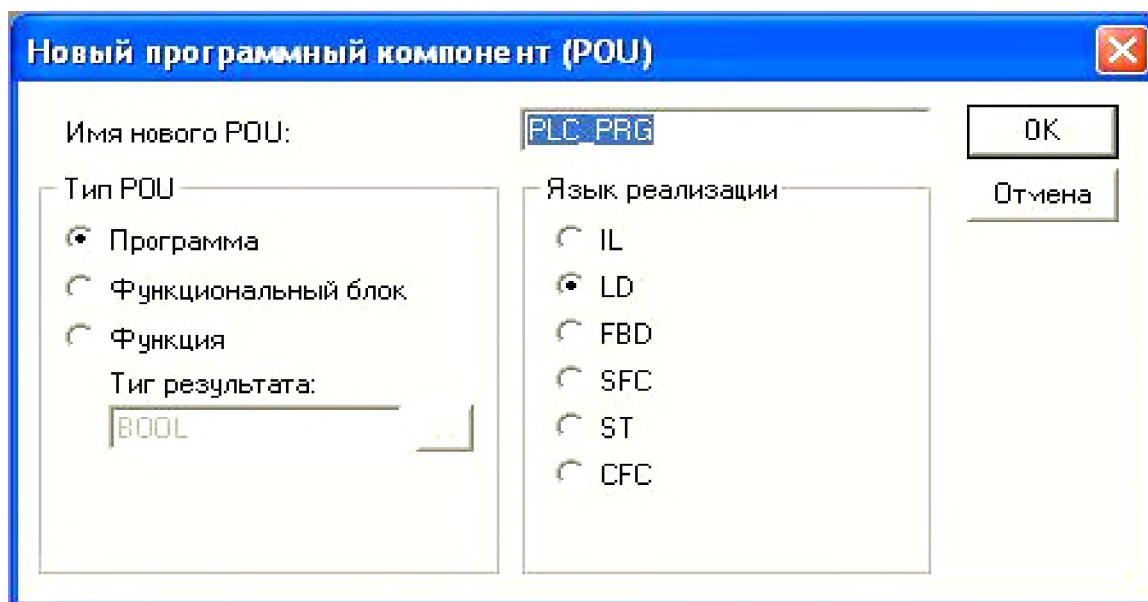


Рис. 1. Выбор языка программирования и задание имени программы

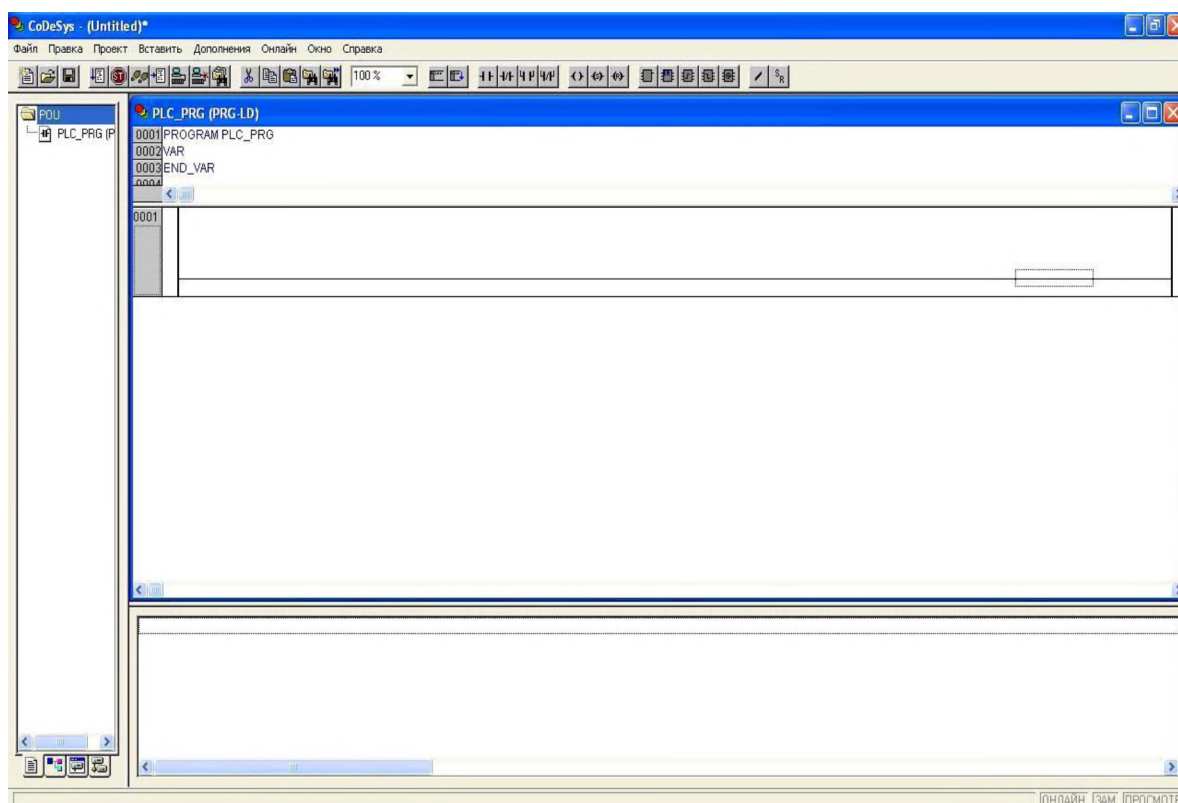


Рис. 2. Главное меню CoDeSys

Его можно разделить на различные области (в окне они расположены сверху вниз):

- меню (рис. 3);
- панель инструментов, которая содержит кнопки для быстрого вызова команд меню (рис. 4);
- организатор объектов, имеющий вкладки «POU», «Типы данных», «Визуализации» и «Ресурсы»;
- разделитель организатора объектов и рабочей области CoDeSys;
- рабочая область, в которой находится редактор;
- окно сообщений;
- строка статуса, содержащая информацию о текущем состоянии проекта.

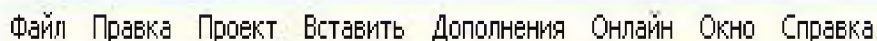


Рис. 3. Меню среды

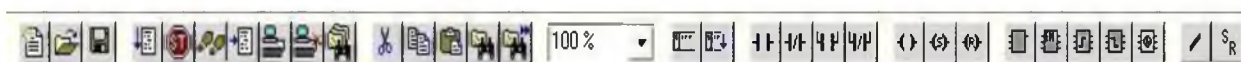


Рис. 4. Панель инструментов

Меню находится в верхней части главного окна. Оно содержит все команды CoDeSys.

Кнопки на панели инструментов обеспечивают более быстрый доступ к командам меню. Вызванная с помощью кнопки на панели инструментов команда автоматически выполняется в активном окне. Команда выполнится, как только нажатая на панели инструментов кнопка будет отпущена. Если вы поместите указатель мышки на кнопку панели инструментов, то через небольшой промежуток времени увидите название этой кнопки в подсказке. Кнопки на панели инструментов различны для разных редакторов CoDeSys. Получить информацию относительно назначения этих кнопок можно в описании редакторов. Кнопки на панели инструментов очень важны, их использование упрощает составление программы на языке LD. В других языках эта панель выглядит иначе. Назначение кнопок на панели инструментов приведено в табл. 1.

Таблица 1

Кнопки на панели инструментов

Графическое изображение кнопки	Название	Назначение	Графическое изображение кнопки	Название	Назначение
	Создать	Создает новый проект		Цепь (вперед)	Вставляет цепь перед текущей
	Открыть	Открывает проект		Цепь (назад)	Вставляет цепь после текущей
	Сохранить	Сохраняет содержимое измененного объекта		Контакт	Вставляет последовательный замыкающий контакт
	Старт	Запускает ПЛК		Инверсный контакт	Вставляет последовательный инверсный (размыкающий) контакт
	Стоп	Останавливает ПЛК		Параллельный контакт	Вставляет параллельный замыкающий контакт
	Шаг по верху	Перешагивает через текущую инструкцию, даже если это вызов подпрограммы		Параллельный инверсный контакт	Вставляет параллельный инверсный (размыкающий) контакт
	Переключить точку останова	Устанавливает/убирает точку останова в текущей позиции		Обмотка	Вставляет обмотку (катушку) «Реле»
	Подключение	Устанавливает связь с ПЛК и включает режим On-Line		Set-обмотка	Вставляет Set-обмотку (катушку)

Графическое изображение кнопки	Название	Назначение	Графическое изображение кнопки	Название	Назначение
	Отключение	Отключает режим On-Line		Reset-обмотка	Вставляет Reset-обмотку (катушку)
	Глобальный поиск	Ищет заданную строку по всему проекту		Функциональный блок	Вставляет функциональный блок
	Вырезать	Перемещает выделенную область в буфер обмена		Элемент с «EN»	Вставляет элемент со входом разрешения
	Копировать	Копирует выделенную область в буфер обмена		Детектор переднего фронта	Вставляет детектор переднего фронта
	Вставить	Вставляет содержимое буфера обмена в текущую позицию		Детектор заднего фронта	Вставляет детектор заднего фронта
	Найти	Ищет заданную строку в текущем окне		Таймер «TON»	Вставляет таймер «TON»
	Найти далее	Повторяет последний поиск		Инверсия	Инвертирует выбранный выход или вход
	Масштаб	Увеличение/уменьшение масштаба		Установка/сброс	Преобразует выход в Set/Reset-выход

При желании панель инструментов можно отключить (в меню - «Проект», в выпадающем списке – «Опции...», категория «Рабочий стол», убрать галочку «Панель инструментов»).

Организатор объектов (рис. 5) всегда находится в левой части главного окна CoDeSys. В нижней части организатора объектов находятся вкладки «POU», «Типы данных», «Визуализации» и «Ресурсы». Переключаться между соответствующими объектами можно с помощью мышки, нажимая на нужную вкладку.

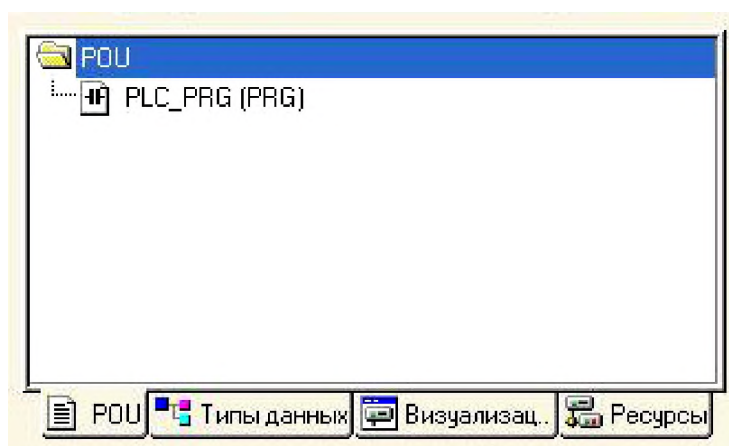


Рис. 5. Организатор объектов

Разделитель экрана – это граница между двумя непересекающимися окнами. В CoDeSys есть следующие разделители: между организатором объектов и рабочей областью, между разделом объявлений и разделом кода POU, между рабочей областью и окном сообщений. Вы можете перемещать разделители с помощью мышки, нажав ее левую кнопку. Разделитель сохраняет свое положение даже при изменении размеров окна. Если вы больше не видите разделителя на экране, значит стоит изменить размеры окна.

Рабочая область (см. рис. 2) находится в правой части главного окна CoDeSys. Все редакторы, а также менеджер библиотек открываются именно в этой области. Имя открытого объекта находится в заголовке окна.

Окно сообщений отделено от рабочей области разделителем. Именно в этом окне появляются сообщения компилятора, результаты поиска и список перекрестных ссылок.

При двойном щелчке левой клавишей мыши или при нажатии клавиши Enter на сообщение будет открыт объект, к которому относится данное сообщение.

Далее сокращенно операции с кнопками мыши будем записывать так: 1ЛКМ, если одно нажатие на левую клавишу мышки, 2ЛКМ – два нажатия; 1ПКМ – один щелчок правой кнопкой.

Статусная строка. Статусная строка находится в нижней части главного окна CoDeSys и предоставляет информацию о проекте и командах меню. Если вы поместили указатель на пункт меню, то в строке статуса появляется его краткое описание.

Если работаете в режиме «Онлайн», то надпись «Онлайн» в строке статуса выделяется черным цветом. В ином случае надпись серая. С помощью статусной строки в режиме «Онлайн» можно определить, в каком состоянии находится программа. Например, «Эмул.» – в режиме эмуляции. Статусную строку можно убрать либо включить (в меню «Проект», «Опции...», «Рабочий стол»).

Контекстное меню. Альтернативой использования главного меню для вызова команд может стать контекстное меню (рис. 6). Это меню, вызываемое ПКМ на рабочей области, содержит наиболее часто используемые команды.

Вырезать	Ctrl+X
Копировать	Ctrl+C
Вставить	Ctrl+V
Удалить	Del
Цепь (перед)	
Цепь (после)	Ctrl+T
Контакт	Ctrl+K
Инверсный контакт	Ctrl+G
Параллельный контакт	Ctrl+R
Параллельный контакт (инверсный)	Ctrl+D
Функциональный блок...	Ctrl+B
Детектор переднего фронта	
Детектор заднего фронта	
Таймер (TON)	
Обмотка	Ctrl+L
'Set' обмотка	Ctrl+I
'Reset' обмотка	
Элемент с EN	
Вставка в блоки	►
Переход	
Возврат	
Комментарий	
Инверсия	Ctrl+N
Set/Reset	
Масштаб	Alt+Enter
Открыть экземпляр	

Рис. 6. Контекстное меню программы CoDeSys

Лабораторная работа 1
**Разработка программы для управления
ОВЕН ПЛК 150 на языке LD**

Цель работы: Научиться создавать первый проект в среде разработки CoDeSys.

Перечень оборудования и приборов

1. Программируемый логический контроллер ПЛК-150 – 1 шт.
2. Персональный компьютер – 1 шт.

Порядок выполнения лабораторной работы

Язык лестничных диаграмм LD (Ladder Diagram) или релейно-контактных схем (РКС) – графический язык, реализующий структуры электрических цепей*. В начале 1970-х гг. XX в. устройства управления, построенные на релейных элементах, начали постепенно вытесняться программируемыми контроллерами. Некоторое время те и другие работали одновременно и обслуживались одним и тем же персоналом.

Так появилась задача прозрачного переноса релейных электрических схем в ПЛК. Различные варианты программной реализации релейных схем создавались практически всеми ведущими производителями ПЛК. Благодаря простоте представления, язык LD обрел заслуженную популярность, что и стало основной причиной включения его в стандарт МЭК.

Графически LD-диаграмма (программа на языке РКС) представлена в виде двух вертикальных шин питания. Между ними расположены цепи, образованные соединением контактов. Нагрузкой каждой цепи обычно служит реле (обмотка). Каждое реле имеет контакты, которые можно использовать в других цепях. Количество контактов в цепи произвольно, реле одно. Если последовательно соединенные контакты замкнуты, ток идет по цепи и реле включается. При необходимости можно включить параллельно несколько реле, последовательное включение не допускается.

* Минаев И. Г., Самойленко В. В. Программируемые логические контроллеры : практ. руководство для начинающего инженера. Ставрополь : АРГУС, 2009. 100 с. ; Петров И. В. Программируемые контроллеры. Стандартные языки и инструменты. М. : СОЛОН-Пресс, 2003. 256 с.

Важно понимать, что контакты, обмотки реле, цепи, шины питания и прочее в LD-диаграмме – это не физические устройства, а элементы, которые реализованы программно.


В LD каждому контакту ставится в соответствие логическая переменная, определяющая его состояние. Если контакт замкнут, то переменная имеет значение ИСТИНА (логическая 1). Если разомкнут – ЛОЖЬ (логический 0). Имя переменной пишется над контактом и фактически служит его названием.

Последовательное соединение контактов или цепей равноценно логической операции И. Параллельное соединение образует монтажное ИЛИ. Цепь может быть либо замкнутой (ON), либо разомкнутой (OFF). Это как раз и отражается на обмотке реле и соответственно на значении логической переменной обмотки (ИСТИНА/ЛОЖЬ или TRUE/FALSE).

Контакт может быть инверсным – нормально замкнутым. Такой контакт обозначается с помощью символа \neg / \neg и замыкается, если значение переменной ЛОЖЬ. Инверсный контакт равнозначен логической операции НЕ. Обмотки реле также могут быть инверсными, что обозначается символом \neg (/) \neg . Если обмотка инверсная, то в соответствующую логическую переменную копируется инверсное значение состояния цепи.

Логически последовательное (И), параллельное (ИЛИ) соединение контактов и инверсия (НЕ) образуют базис алгебры Буля. В результате LD идеально подходит для программной реализации комбинационных логических систем управления. Благодаря возможности включения в LD-программы функций и функциональных блоков, выполненных на других языках, сфера применения языка практически не ограничена.

Основные элементы программы. После открытия главного окна CoDeSys появляется на мониторе рабочая область (см. рис. 2), в которой и будем изображать многоступенчатую схему (лестничную диаграмму).

Эта схема представляет собой набор горизонтальных цепей, напоминающих ступеньки лестницы, соединяющих вертикальные шины питания. Если необходимо увеличить размер рабочей области, то подводим курсор к кнопке  в верхнем правом углу этой области и нажимаем ЛКМ. Первая цепь появляется в рабочей области сразу (см. рис. 2). Слева на сером фоне автоматически возникнет её номер: 0001. Наличие пунктирного прямоугольника в правой части цепи сви-

детельствует о том, что она активирована, т. е. готова принимать вносимые в неё компоненты: контакты, функциональные блоки FB, катушки реле.

Будем считать, что сама система логико-программного управления дискретным процессом в обычном, т. е. релейно-контактном исполнении, уже имеется, и наша задача перенести её в LD-диаграмму. В качестве такой системы управления возьмем схему, представленную на рис. 7.

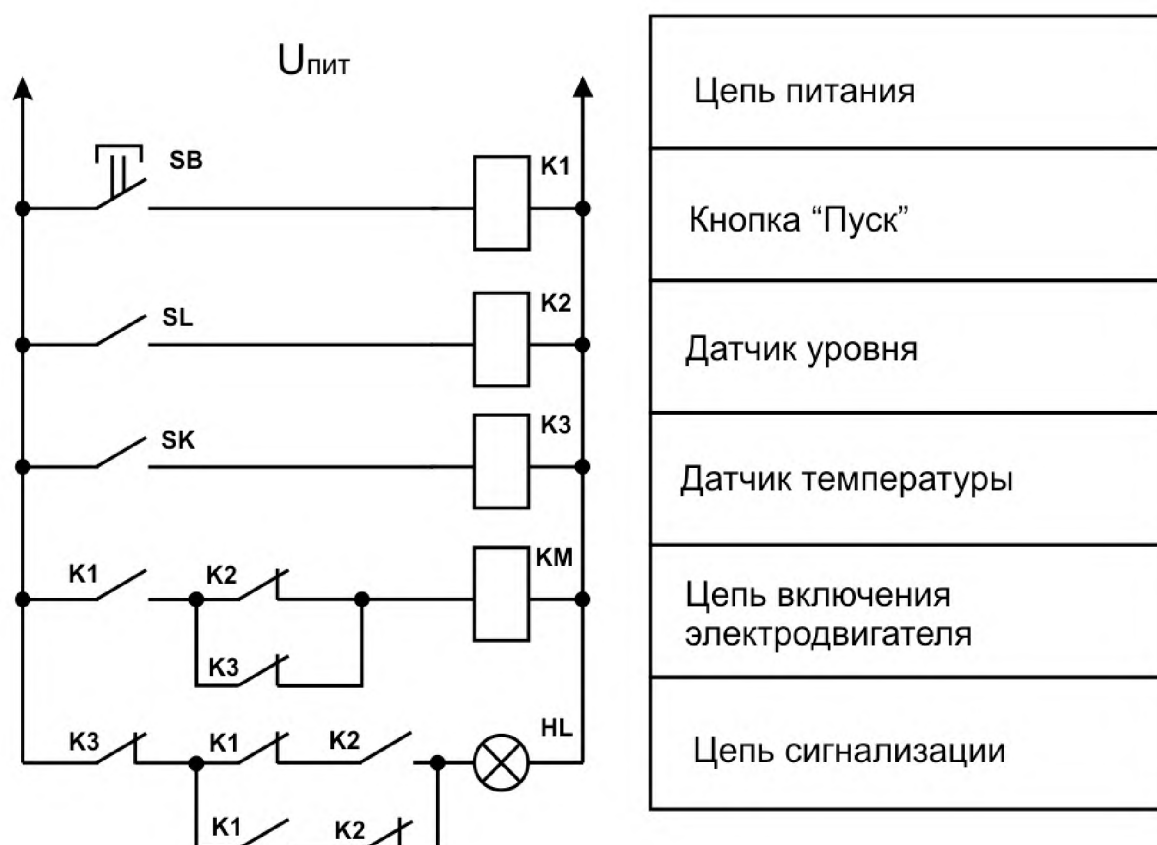


Рис. 7. Релейно-контактная схема системы логико-программного управления дискретным процессом

В состав схемы входят:

- 1) три источника входных дискретных сигнала (кнопка управления «Пуск» SB, два электроконтактных дискретных (не аналоговых) датчика – датчик температуры SK и уровня SL);
- 2) три промежуточных реле K1, K2, K3;
- 3) два выходных устройства – обмотка магнитного пускателя KM, включающего электродвигатель, и сигнальная лампа HL.

Методика программирования на языке LD в среде CoDeSys

По аналогии с исходным вариантом (см. рис. 7) в многоступенчатой LD-программе, скорее всего, также потребуется пять цепей. Словосочетание «скорее всего» использовано потому, что несмотря на кажущуюся схожесть РКС и LD-диаграмм, есть принципиальное различие в последовательности срабатывания цепей.

В первую цепь необходимо внести контакт SB и катушку реле K1.

1. Наводим курсор на кнопку в панели инструментов с изображением замыкающего контакта, нажимаем 1ЛКМ, и этот контакт появляется в цепи с тремя вопросительными знаками красного цвета (рис. 8). Эти ??? запрашивают имя или идентификатор внесенного в цепь компонента.

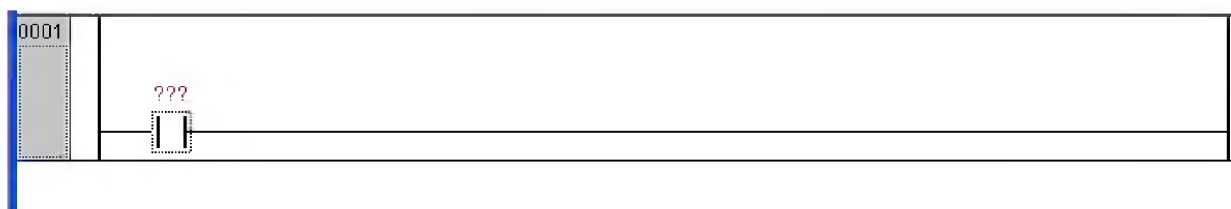


Рис. 8. Первая цепь с введенным контактом

2. Наводим курсор на ???, щелкаем 1ЛКМ. Вопросы становятся белыми на фоне синего прямоугольника. С помощью клавиатуры на английском языке задаем имя. В нашем примере «SB». Буквы русского алфавита использовать нельзя!

3. Нажимаем «Enter». Открывается окно Объявление переменной, запрашивающее, к какому классу переменных будет отнесен наш элемент (рис. 9).

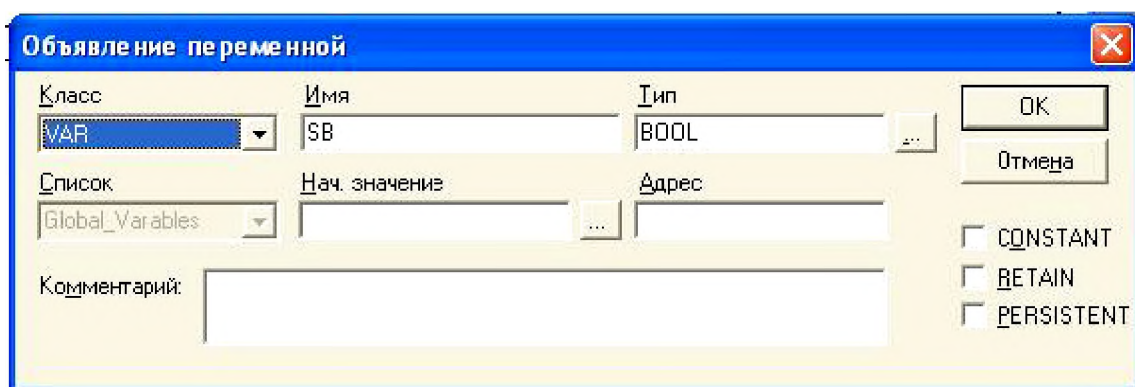


Рис. 9. Окно объявления переменной

Если проектируемая схема предназначена для учебных целей и будет работать только в режиме эмуляции, то можно сразу нажать 1ЛКМ на ОК, и имя появится над элементом (рис. 10).

Нужно обратить внимание на пунктирный квадратик, охватывающий контакт SB, и на исчезновение пунктирного прямоугольника в конце самой цепи. Это свидетельствует о том, что активирован сам контакт SB, т. е. можно (если была бы такая необходимость) последовательно и/или параллельно этому контакту (как будет показано при создании четвертой и пятой цепей) подключать к нему другие контакты.

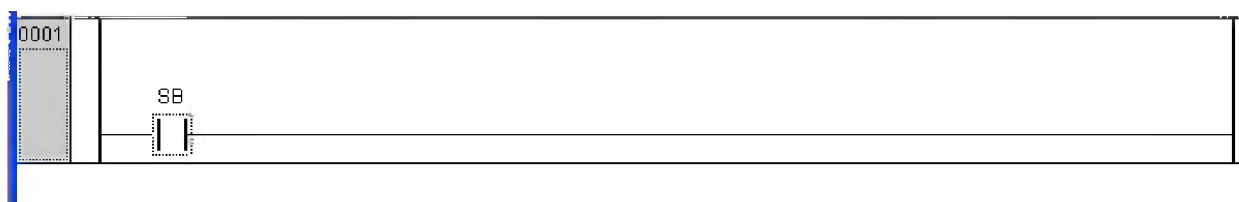




Рис. 10. Объявленный замыкающий контакт

Для завершения первой цепи необходимо ввести в нее катушку реле K1.

4. Наводим курсор на , щелкаем 1ЛКМ и ... ничего не получилось. Надо сначала активировать цепь. Для этого наводим курсор на линию цепи, щелкаем 1ЛКМ, т. е. активируем её (появился пунктирный прямоугольник!), переводим курсор на , щелкаем 1ЛКМ и в цепи появляется катушка с тремя белыми ??? в синем прямоугольнике.

Катушки и, как будет показано ниже, функциональные блоки FV появляются всегда с такими ???. Если имя сразу не присвоить и перейти к другим действиям, то ??? станут красными, как и в случае включения контактов. Потом же потребуется лишняя операция при идентификации катушки: опять навести курсор на красные ???, щелкнуть 1ЛКМ, вопросы становятся белыми на синем фоне и т. д. Но это не принципиально.

5. Присваиваем катушке имя K1. Нажимаем Enter, открывается окно Объявление переменной, нажимаем 1ЛКМ на ОК.

Можно сначала включить в цепь катушку, затем контакт. Можно включить эти элементы в любой последовательности и лишь потом присвоить им имена. Результат будет тем же.

Первая цепь завершена (рис. 11). Как видно на мониторе, зона цепи ограничена двумя горизонтальными линиями. При активации

цепи необязательно наводить курсор точно на цепь. Достаточно попасть в эту зону за исключением полосы, примыкающей к верхней линии и являющейся как бы продолжением строки, начинающейся с номера цепи. В нашем случае с 0001. Эта полоса может быть использована для задания на английском языке адресов меток переходов.

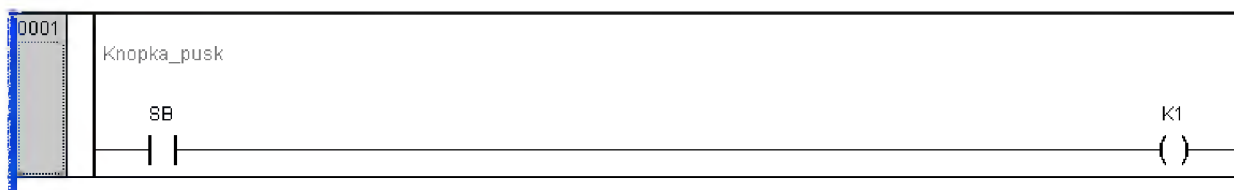


Рис. 11. Завершенная первая цепь

Для обозначения функционального назначения цепи в LD-диаграммах часто задают комментарии на любом языке, понятном, естественно, обслуживающему персоналу.

С этой целью наводим указатель на полосу, расположенную ниже полосы меток и щелкаем 1ЛКМ, появляется мигающий курсор, с клавиатуры задаем, например, «Knopka_puska» и нажимаем Enter.

Комментарии можно выполнить и на русском языке. С этой целью наводим курсор на поле цепи, щелкаем 1ПКМ. Открывается контекстное меню (см. рис. 6). Щелкаем 1ЛКМ на строке «Комментарий». Слева над цепью появится слово Comment, в строке с которым можно написать на любом языке необходимые пояснения. Слово Comment потом можно удалить.

Перед тем как взяться за следующую цепь, обсудим ещё некоторые важные моменты. Если реальное реле имеет ограниченное количество замыкающих, размыкающих и переключающих контактов, то в LD таких ограничений нет, и виртуальные контакты могут применяться в любой цепи в любом количестве. Во всех цепях одной схемы имя логической переменной контактов одного и того же реле должно сохраняться. Имя может быть однобуквенным (X, Y, Z и т. д.), иметь цифровые индексы (X1, X2 и т. д.), вписываемые без пробела. Если есть необходимость в таком пробеле, например, при написании двух или более слов, то в пробел ставится символ подчеркивания. Этот символ является значимым, т. е. имена X_1, X1, _X1 и _X_1 воспринимаются программой как самостоятельные. Но цифру на первое место ставить нельзя: 1X, 2X – неправильно!

Нельзя применять в качестве имен операторы других языков. Например, операторы языка инструкций IL: LD, ST, S, R, AND, MUL,



JMP и др. С индексами, символами подчеркивания или другими буквами можно. Например S1, RU, AND_, MULTI и т. д. Даже не зная весь список операторов, легко установить ошибку. После нажатия на клавишу Enter в процессе присваивания имени переменной запрещенный идентификатор высветится синим цветом. Его необходимо удалить и вписать другое имя.


Регистр букв не влияет на работу ПЛК. Так, имена «SET» и «Set» воспринимаются одинаково. В сложных схемах трудно запомнить назначение того или иного элемента при упрощенной (однобуквенной) системе идентификации. Поэтому имя переменной (т. е. идентификатор) можно записать в развернутом виде, не используя буквы русского языка.

Например, если есть трудности с английским языком, можно присвоить имена на русском языке «Dvigatel», «pusk», «BLOKIROVKA» и т. д.

Каждая цепь заканчивается обмоткой (катушкой) реле. Последовательно соединять катушки нельзя. Параллельно – можно.

Создаем вторую цепь.

Есть три способа: с помощью меню «Вставка» контекстного меню (см. рис. 6) и кнопками  и . Воспользуемся третьим приемом как наиболее простым.

1. Наводим курсор на , щелкаем 1ЛКМ. Сразу появляется вторая цепь (рис. 12).

2. По вышеописанной методике вносим в цепь замыкающий контакт датчика SL и катушку K2. Если есть необходимость, то можно вписать комментарий: например, «Datchik_urovnia».

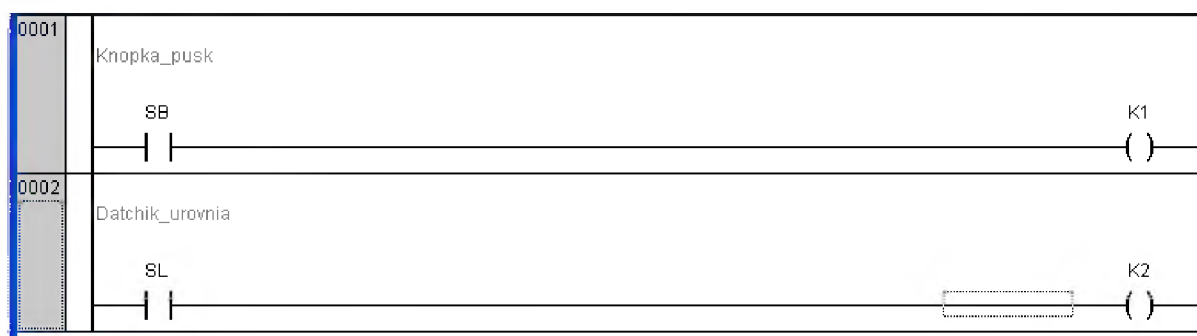






Рис. 12. Вторая цепь

Если по ошибке или преднамеренно нажали не , а , т. е. «Цепь» (вперед), то новая цепь станет впереди всех созданных и ав-

томатически присвоит себе номер 0001, а последующим цепям нумерацию увеличит на единицу. Если же проектировщик решил вставить новую цепь между двумя уже созданными, то можно это выполнить разными приёмами:

- щелкнуть 1ЛКМ по цепи 0001, затем по , т. е. «Цепь» (после). Новая цепь станет после цепи 0001, получит номер 0002, а вторая, ранее созданная цепь, станет под номером 0003;


- щелкнуть 1ЛКМ по цепи 0002, затем по  и новая цепь появится впереди цепи 0002, присвоит себе её номер, а ранее созданной цепи передаст номер 0003.

Создаем третью цепь, включая в её состав замыкающий контакт датчика температуры SK и обмотку K3.

Несколько сложнее будет программирование четвертой и пятой цепей.

По известной методике создаем четвертую цепь, последовательно внося в цепь замыкающий контакт K1, размыкающий K2. Теперь необходимо создать параллельную цепочку из размыкающего контакта K2 и размыкающего K3. Для этого:

- наводим курсор на контакт K1, щелкаем 1ЛКМ. Контакт активирован, о чем свидетельствует пунктирный квадратик, охватывающий контакт;

- наводим курсор на , щелкаем 1ЛКМ и соединение выполнено. Присваиваем имя K3.

Если активировать саму цепь, а не контакт K2, то замыкающий контакт K3 охватил бы всю цепочку из контактов K1 и K2, как показано на рис. 13.




Рис. 13. Фрагмент программы после неудачного введения контакта K3


Вносим в цепь катушку KM. Четвертая цепь завершена. Программируем пятую цепь LD-программы.

По известной методике вносим в цепь размыкающие контакты K3 и K1, замыкающий K2 и обмотку (катушку) HL. Необходимости

объявлять переменные K1, K2 и K3 нет, так как они уже внесены в список булевых переменных. Осталось создать параллельную цепочку из замыкающего контакта K1 и размыкающего K2. Для этого выполним следующее:

- активируем размыкающий контакт K1, нажимая и не отпуская «Shift» активируем замыкающий контакт K2, затем отпускаем «Shift», оба контакта оказались охваченными пунктирным прямоугольником;

- наводим курсор на , щелкаем 1ЛКМ и параллельно с этими контактами появится —| |—, которому присвоим имя K1. Фрагмент полученной схемы в пятой цепи показан на рис. 14;

- активируем новый контакт, наводим курсор на , щелкаем 1ЛКМ и последовательно появляется —| / |—. Присваиваем имя K2 новому элементу.

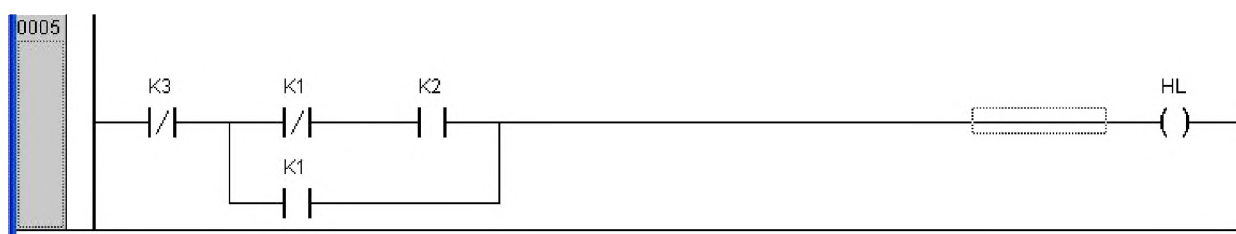



Рис. 14. Пятая цепь после введения замыкающего контакта K1 и его активации

Параллельная цепочка выполнена, и завершена вся многоступенчатая схема в виде LD-программы (рис. 15).

Дополнительные приемы при разработке LD-программ. Если по ошибке введен вместо замыкающего контакта размыкающий или наоборот, то можно навести на него курсор, щелкнуть 1ПКМ и в открывшемся контекстном меню (см. рис. 6) щелкнуть 1ЛКМ на четвертой снизу строке «Инверсия» (Negate). Произойдет инверсия этого элемента. Имя, если оно уже было присвоено, остается прежним.

Эту операцию можно выполнить проще, если есть кнопка . (В некоторых версиях CoDeSys она может отсутствовать). Тогда наведем курсор на подлежащий изменению контакт, щелкаем 1ЛКМ и нажимаем на указанную кнопку.

Если необходимо удалить какой-то компонент цепи или всю цепь со всеми элементами, то наводим курсор на этот компонент (контакт, катушку, FB) или на саму цепь, щелкаем 1ПКМ и в открывшемся контекстном меню нажимаем 1ЛКМ на самой верхней строке «Вырезать» (Cut). Исчезнет элемент или вся цепь соответственно.

Если нужно изменить идентификатор какого-то компонента цепи, наводим курсор на это имя, щелкаем ЛКМ, нажатием на клавишу Backspace удаляем имя, задаем новое, нажимаем Enter.

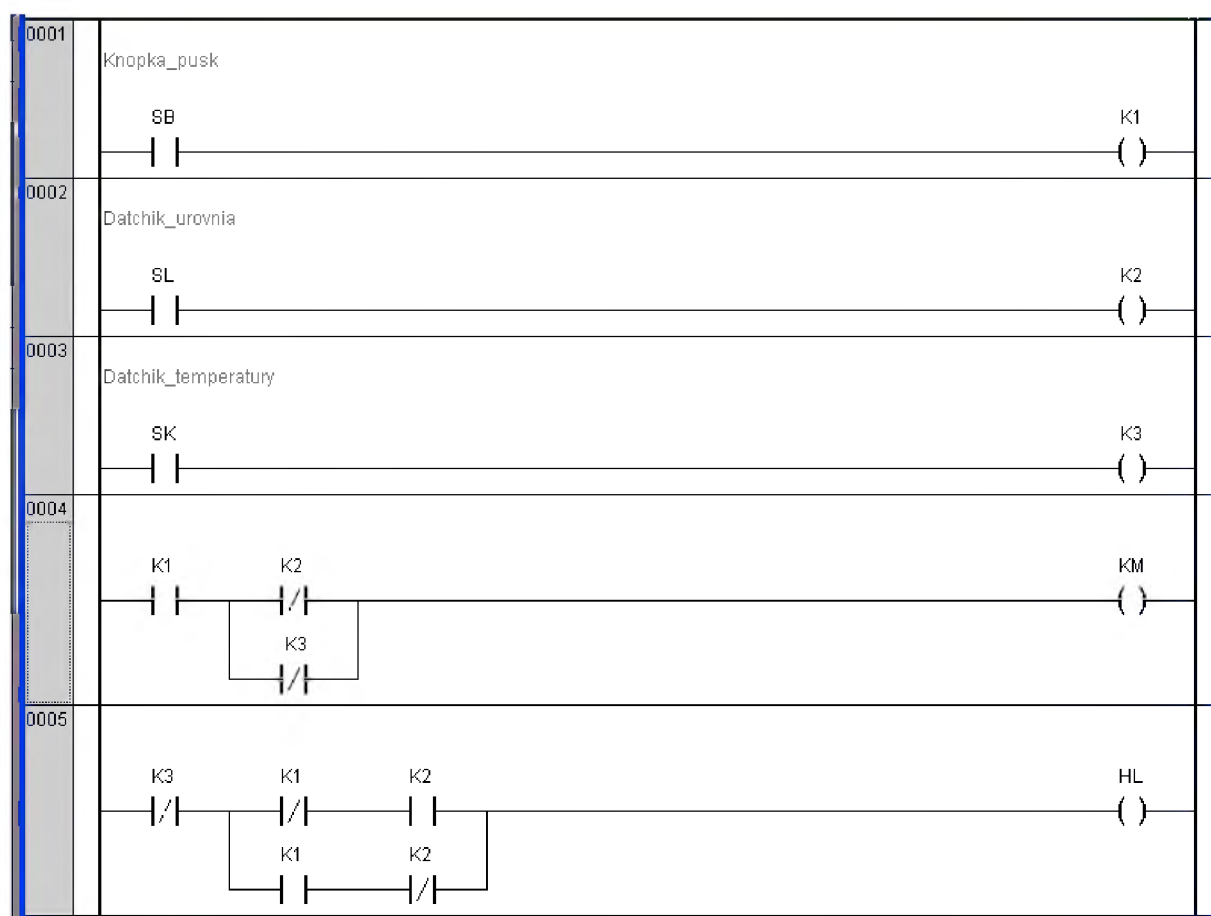


Рис. 15. Система логико-программного управления дискретным процессом в виде программы на языке LD

Если возникла необходимость переместить какой-то компонент схемы по одной цепи или даже перекинуть его в другую цепь, то захватываем этот элемент курсором и при нажатой ЛКМ переставляем его в новое место. В момент перемещения на входе и выходе каждого элемента высвечиваются небольшие прямоугольники. Как только курсор подойдет к одному из них, прямоугольник становится зеленым. Отпускаем ЛКМ и перемещаемый элемент займет новое место.

Если по мере увеличения количества элементов в цепи им уже не хватает места в пределах границ рабочей области, то пользуясь горизонтальной «прокруткой» смещаем цепь влево. Если по мере добавления цепей им уже не где размещаться, то с помощью вертикальной «прокрутки» поднимаем всю многоступенчатую схему или уменьшаем масштаб.

Задание в программе выходных дискретных сигналов. Помимо «обычных» реле $-()$ можно применять «инверсный» аналог, обозначаемый в программе $-(/)$. Это реле может иметь сколько угодно замыкающих и размыкающих контактов, но логика их действия противоположна поведению контактов обычного реле: при отсутствии тока в $-(/)$ замыкающий контакт $-| |$ замкнут, размыкающий контакт $-| / |$ разомкнут. При подаче питания в катушку $-(/)$ состояние его контактов меняется на противоположное.

В наборе программных компонентов имеются также специальные обмотки SET и RESET, обозначенные в линейке кнопок как $-(S)$ и $-(R)$. С их помощью можно фиксировать условия управления исполнительным механизмом. Если обмотка S «срабатывает», т. е. примет значение ИСТИНА (TRUE), то изменить это состояние на противоположное, т. е. ЛОЖЬ (FALSE), можно лишь с помощью обмотки R (рис. 16).

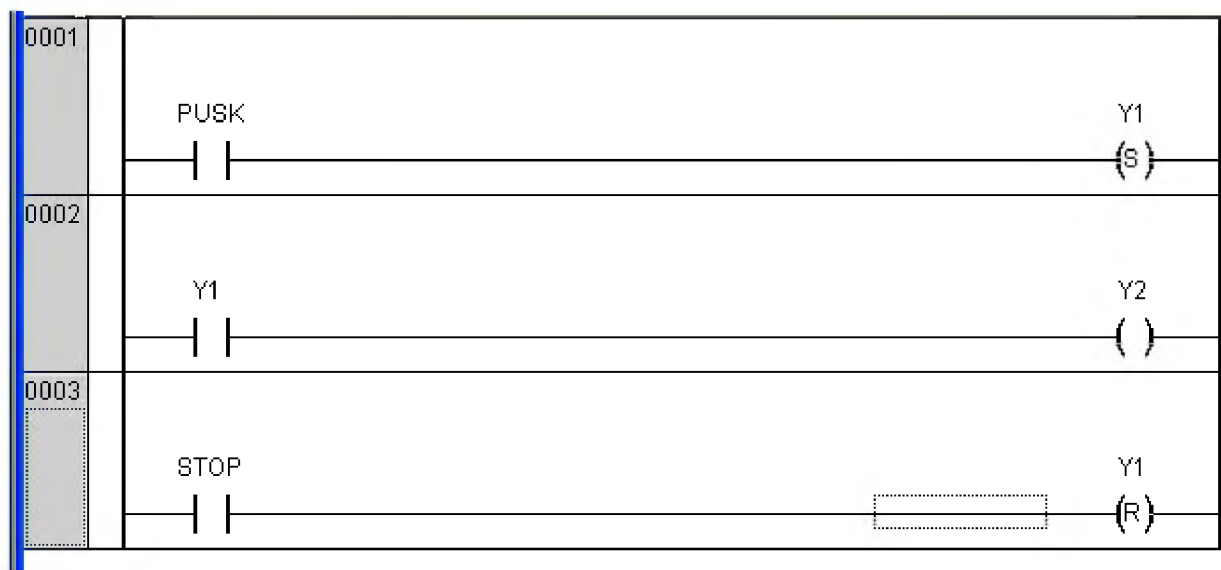


Рис. 16. LD-диаграмма фиксации катушки реле Y2 с помощью обмоток S и R

Эта схема работает как классический RS-триггер: при кратковременном нажатии кнопки PUSK срабатывает катушка S, которой присвоено имя Y1, и своим контактом Y1 включает нагрузку – катушку реле Y2. Выключить реле Y2 можно только нажатием кнопки STOP. Одновременное нажатие на PUSK и STOP, как и в классическом RS-триггере, недопустимо.

Следует заметить, что катушкам R и S присвоено одно и то же имя! В нашем примере Y1.

Эту же задачу самофиксации можно выполнить и на обычном реле (рис. 17).

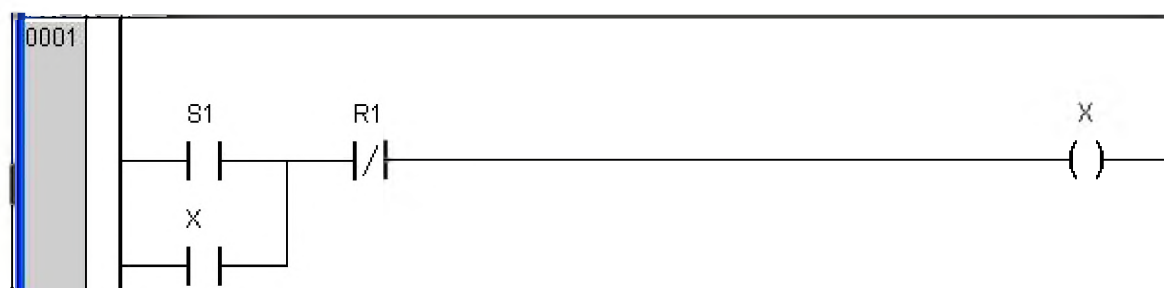


Рис. 17. Схема управления катушкой X самофиксацией состояния

При кратковременном нажатии на кнопку S1 происходит срабатывание реле X, которое своим контактом X фиксирует это состояние. Отключение реле X возможно только нажатием на кнопку R1.

Указанный прием самофиксации (самопитания) широко используется на практике при организации логико-программного управления дискретными исполнительными устройствами.

Исследование LD-программы в режиме эмуляции. Итак, исходную РКС (см. рис. 7) запрограммировали в CoDeSys на языке LD и представили её в виде многоступенчатой схемы (см. рис. 15). Теперь необходимо проверить выполнение запланированных условий срабатывания и отсутствие ложных включений исполнительных элементов.

Все это можно выполнить в режиме эмуляции, не используя реальные аппаратные средства и сам ПЛК. Для этого выполняем следующие действия:

1) наводим курсор на «Онлайн» (Online) (см. рис. 3), щелкаем ЛКМ. Открывается меню (рис. 18);

2) смещаем курсор к строке «Режим эмуляции» (Simulation Mode), щелкаем ЛКМ. Перед этой строкой появится «галочка» ✓, что свидетельствует о готовности программы к «диалогу». И в дальнейшем при работе с этой программой даже в случае внесения в неё изменений, дополнений повторять эту операцию не надо;

3) снова открываем это меню. Щелкаем ЛКМ по строке «Подключение» (Login). Левая вертикальная «шина питания», размыкающие контакты реле, участки цепей от этой шины до какого-то «непро-

водящего» в этом положении элемента цепи (например, замыкающего контакта) окрашиваются в синий цвет.

Подключение	Alt+F8
Отключение	Ctrl+F8
Загрузка	
Старт	F5
Стоп	Shift+F8
Сброс	
Сброс (холодный)	
Сброс (заводской)	
Переключить точку останова	F9
Диалог точек останова	
Шаг поверху	F10
Шаг детальный	F8
Один цикл	Ctrl+F5
Записать значения	Ctrl+F7
Фиксировать значения	F7
Освободить фиксацию	Shift+F7
Диалог Запись/Фиксация	Ctrl+Shift+F7
Показать стек вызовов...	
Отображать поток выполнения	
✓ Режим эмуляции	
Параметры связи...	
Загрузка исходных текстов	
Создание загрузочного проекта	
Записать файл в ПЛК	
Читать файл из ПЛК	

Рис. 18. Меню Онлайн

Возможна ситуация, когда после щелчка 1ЛКМ по «Подключение» будет выдано извещение о допущенной ошибке в программе. Возможно, забыли присвоить имя какому-то компоненту или снять ??? на входе того или иного функционального блока FB. В любом случае, не внеся требуемые коррективы, двигаться дальше нельзя. Будем считать, что в нашей схеме нет ошибок или мы их уже устранили;

4) опять открываем «Контекстное меню» и щелкаем 1ЛКМ по строке «Старт» (Run). Программа готова к приему входных сигналов. Можно приступить к её исследованию;

5) начнем с комбинации входных сигналов, равных логическому нулю $SB = 0$, $SL = 0$, $SK = 0$, т. е. SB, SL и SK не сработали. Открываем меню «Онлайн» и щелкаем 1ЛКМ по строке «Записать значения»

(Write Values). Убеждаемся, что K1, K2, K3, KM и HL не сработали, о чем свидетельствует их неизменившийся вид;

б) переходим ко второй комбинации. Должен замкнуться контакт кнопки управления SB ($SB = 1$). Наводим курсор на контакт SB, щелкаем 2ЛКМ. В полости этого контакта появится синий квадратик. Открываем меню «Онлайн» и щелкаем 1ЛКМ по строке «Записать значения». Контакт SB, катушка K1 и вся первая цепь залятся синим цветом. Контакты K1 в четвертой и пятой цепях замыкаются и также окрашиваются в синий цвет. Размыкающие контакты K3 и K4 в этих цепях остаются замкнутыми, следовательно, срабатывают KM и HL.

Аналогично необходимо исследовать работу программы при всех других комбинациях входных сигналов с целью проверки правильности реализации алгоритма управления и выявления возможности ложных срабатываний исполнительных элементов.

Например, необходимо «разомкнуть» контакт SB и «замкнуть» SK. Для этого наводим курсор на SB, щелкаем 1ЛКМ. В полости контакта появится зеленый квадратик и останется часть синей заливки. Наводим курсор на SK, щелкаем 1ЛКМ. Контакт заливается синим цветом. Открываем меню «Онлайн», щелкаем 1ЛКМ по строке «Записать значения». Убеждаемся, что выключилась катушка KM. И так проходим последовательно по всем строкам таблицы состояний входных и выходных устройств.

В случае исследования такой простой системы, как в нашем примере, можно сразу проверить только условия срабатывания исполнительных элементов, создавая «замыкание» и «размыкание» соответствующих входных элементов SB, SL и SK по вышеописанной методике.

Если, приступая к исследованию реакции LD-программы на следующую комбинацию состояний входных сигналов, необходимо вернуть схему в исходное состояние ($SB = 0$, $SL = 0$ и $SK = 0$), то для этого достаточно щелкнуть 1ЛКМ по строке «Отключение» (Logout), т. е. отключить режим «Онлайн». Затем по освоенной уже методике запустить режим эмуляции, создать требуемую комбинацию состояний входных элементов и, нажав 1ЛКМ по строке «Записать значения», зафиксировать поведение исполнительных элементов.

При большом количестве цепей многоступенчатой схемы нет возможности видеть ее на мониторе целиком даже при самом малом масштабе. Поэтому перед нажатием 1ЛКМ на строку «Записать значения» можно, пользуясь прокрутками, переместить интересующий нас участок схемы в поле зрения и лишь потом запустить ее.