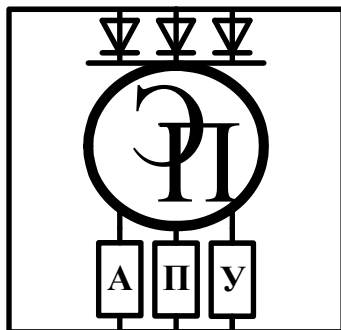


ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и АПУ»

СИСТЕМЫ ПРОГРАММНОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИХ КОМПЛЕКСОВ

*Методические рекомендации к лабораторным работам
для студентов специальности 1-53 01 05
«Автоматизированные электроприводы»
дневной и заочной форм обучения*



Могилёв 2017

УДК 004.4
ББК 32.295
С 40

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Электропривод и АПУ» «6» февраля 2017 г.,
протокол № 7

Составитель ст. преподаватель А. В. Янкович

Рецензент канд. техн. наук С. В. Болотов

Методические рекомендации предназначены к лабораторным работам для студентов специальности 1-53 01 05 «Автоматизированные электроприводы».

Учебно-методическое издание

СИСТЕМЫ ПРОГРАММНОГО УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИХ КОМПЛЕКСОВ

Ответственный за выпуск

Г. С. Леневский

Технический редактор

А. А. Подошевка

Компьютерная вёрстка

Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 115 экз. Заказ №

Издатель и полиграфическое исполнение:

Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изделий

№ 1/156 от 24.01.2014.

Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2017



Содержание

1 Лабораторная работа № 1. Изучение технических характеристик программируемых логических контроллеров.....	4
1.1 Краткие теоретические сведения.....	4
2 Лабораторная работа № 2. Изучение системы программирования логических контроллеров CoDeSys.....	15
2.1 Общие сведения о системе программирования CoDeSys.....	16
2.2 Структура проекта CoDeSys.....	17
2.3 Создание нового проекта.....	20
2.4 Переменные. Объявление переменных.....	22
2.5 Компоненты организации программ.....	24
2.6 Подключение библиотек.....	26
2.7 Стандартные операции.....	27
2.8 Установка target-файлов.....	28
2.9 Настройка связи компьютера с контроллером.....	30
2.10 Конфигурация памяти ввода/вывода контроллера.....	32
2.11 Привязка переменных к входам и выходам ПЛК	34
3 Лабораторной работе № 3. Программирование логических контроллеров на языке LD в системе CoDeSys.....	37
3.1 Общие сведения.....	37
3.2 Пример программирования ПЛК на LD.....	38
Список литературы.....	42

1 Лабораторная работа № 1. Изучение технических характеристик программируемых логических контроллеров

Цель работы: изучение технических характеристик, принципа работы и структурных схем программируемых логических контроллеров.

1.1 Краткие теоретические сведения

Структура и устройство ПЛК.

Первые программируемые логические контроллеры (ПЛК) появились в 70-х гг. XX в. и предназначались для использования в сфере промышленной автоматики для решения технологических задач, которые описывались преимущественно логическими уравнениями. Это позволило ПЛК заменять в системах управления блоки релейной автоматики и устройства жесткой логики на интегральных микросхемах малой и средней степени интеграции. Название «программируемый логический контроллер» происходит от английского Programmable Logic Controller (PLC).

Сегодня ПЛК – это специализированное микропроцессорное устройство с широкими функциональными возможностями. Специальное проблемноориентированное программное обеспечение ПЛК позволяет реализовывать не только алгоритмы программно-логического управления, но и осуществлять регулирование в замкнутых системах автоматического управления, производить сбор, обработку, хранение и передачу информации. Конструкция ПЛК позволяет эксплуатировать устройство в производственных условиях.

Кроме этого, ПЛК отличается универсальностью структуры, программируемостью и возможностью решения определенного класса задач при управлении технологическим объектом в режиме реального времени.

ПЛК имеет конечное количество входов и выходов, к которым подключены датчики и исполнительные устройства, обеспечивающие связь с технологическим объектом управления. На рисунке 1.1 представлена структурная схема ПЛК.

Принцип работы ПЛК несколько отличается от «обычных» микропроцессорных устройств. Программное обеспечение универсальных контроллеров состоит из двух частей. Первая часть – это системное программное обеспечение. Проводя аналогию с компьютером, можно сказать, что это операционная система, которая управляет работой узлов контроллера, взаимосвязями составляющих частей, внутренней диагностикой. Системное программное обеспечение ПЛК расположено в постоянной памяти центрального процессора и всегда готово к работе. По включении питания ПЛК готов взять на себя управление системой уже через несколько миллисекунд



Рисунок 1.1 – Структурная схема ПЛК

Вторая часть программного обеспечения – это программа управления, которая разрабатывается при проектировании системы управления технологическим объектом и хранится в перепрограммируемой памяти. Управляющая программа может изменяться в процессе эксплуатации оборудования, например при модернизации технологической установки.

ПЛК работают циклически по методу периодического опроса входных данных.

Рабочий цикл ПЛК включает четыре фазы:

- 1) опрос входов;
- 2) выполнение пользовательской программы;
- 3) установку значений выходов;
- 4) некоторые вспомогательные операции (диагностика, подготовка данных для отладчика, визуализации и т. д.).

Выполнение первой фазы обеспечивается системным программным обеспечением, после чего управление передается прикладной программе, которая записана в память. По этой программе контроллер делает то, что необходимо, а по ее завершении управление опять передается системному уровню. За счет этого обеспечивается максимальная простота построения прикладной программы – ее создатель не должен знать, как производится управление аппаратными ресурсами. Необходимо знать, с какого входа приходит сигнал и как на него реагировать на выходах.

Очевидно, что время реакции на событие будет зависеть от времени выполнения одного цикла прикладной программы. Определение времени реакции – времени от момента события до момента выдачи соответствующего управляющего сигнала – поясняется на рисунке 1.2.

Обладая памятью, ПЛК в зависимости от предыстории событий способен реагировать по-разному на текущие события. Возможности перепрограммирования, управления по времени, развитые вычислительные способности, включая цифровую обработку сигналов, поднимают ПЛК на более высокий уровень в отличие от простых комбинационных автоматов.

Рассмотрим входы и выходы ПЛК. Существует три вида входов: дискретные, аналоговые и специальные.

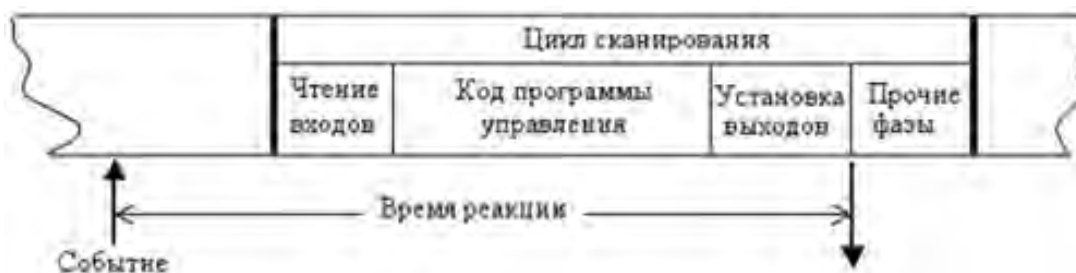


Рисунок 1.2 – Рабочий цикл ПЛК

Один дискретный вход ПЛК способен принимать один бинарный электрический сигнал, описываемый двумя состояниями – включен или выключен. Все дискретные входы (общего исполнения) контроллеров обычно рассчитаны на прием стандартных сигналов с уровнем 24 В постоянного тока. Типовое значение тока одного дискретного входа (при входном напряжении 24 В) составляет около 10 мА.

Аналоговый электрический сигнал отражает уровень напряжения или тока, соответствующий некоторой физической величине, в каждый момент времени. Это может быть температура, давление, вес, положение, скорость, частота и т. д.

Поскольку ПЛК является цифровой вычислительной машиной, аналоговые входные сигналы обязательно подвергаются аналого-цифровому преобразованию (АЦП). В результате образуется дискретная переменная определенной разрядности. Как правило, в ПЛК применяются 8–12-разрядные преобразователи, что в большинстве случаев, исходя из современных требований по точности управления технологическими процессами, является достаточным. Кроме этого, АЦП более высокой разрядности не оправдывают себя в первую очередь из-за высокого уровня промышленных помех, характерных для условий работы контроллеров.

Стандартные дискретные и аналоговые входы ПЛК способны удовлетворить большинство потребностей систем промышленной автоматизации. Необходимость применения специализированных входов возникает в случаях, когда непосредственная обработка некоторого сигнала программно затруднена, например, требует много времени.

Наиболее часто ПЛК оснащаются специализированными счетными входами для измерения длительности, фиксации фронтов и подсчета импульсов.

Например, при измерении положения и скорости вращения вала очень распространены устройства, формирующие определенное количество импульсов за один оборот – поворотные шифраторы. Частота следования импульсов может достигать нескольких мегагерц. Даже если процессор ПЛК обладает достаточным быстродействием, непосредственный подсчет импульсов в пользовательской программе будет весьма расточительным по времени. Здесь желательно иметь специализированный

аппаратный входной блок, способный провести первичную обработку и сформировать необходимые для прикладной задачи величины.

Вторым распространенным типом специализированных входов являются входы, способные очень быстро запускать заданные пользовательские задачи с прерыванием выполнения основной программы, – входы прерываний.

Дискретный выход также имеет два состояния – включен и выключен. Они нужны для управления электромагнитных клапанов, катушек, пускателей, световых сигнализаторов и т. д. В общем сфера их применения огромна и охватывает почти всю промышленную автоматику.

Конструктивно ПЛК подразделяются на моноблочные, модульные и распределенные. Моноблочные имеют фиксированный набор входов-выходов (рисунок 1.3).



Рисунок 1.3 – Моноблочный ПЛК

В модульных контроллерах модули входов-выходов устанавливаются в разном составе и количестве в зависимости от предстоящей задачи (рисунок 1.4).



Рисунок 1.4 – Модульный ПЛК

В распределенных системах модули или даже отдельные входы-выходы, образующие единую систему управления, могут быть разнесены на значительные расстояния.

Обзор контроллеров компании «ОВЕН».

Компания «ОВЕН» выпускает программируемые логические контроллеры серий:

- контроллеры с НМІ для локальных систем автоматизации ОВЕН ПЛК63/ПЛК73;
- контроллеры для малых систем автоматизации ОВЕН ПЛК100 / ПЛК150 / ПЛК154;
- моноблочные контроллеры с дискретными и аналоговыми входами/выходами для средних систем автоматизации ПЛК110 [M02] / ПЛК110 / ПЛК160.

Программируемый логический контроллер ОВЕН ПЛК63.

ОВЕН ПЛК63 (рисунок 1.5) – контроллер с НМІ для построения локальных систем автоматизации. Сегодня главные области применения этих контроллеров – ЖКХ, ЦТП, ИТП, котельные и различные небольшие установки.



Рисунок 1.5 – Контроллер ОВЕН ПЛК63

Устройство ОВЕН ПЛК63 построено на базе 32-разрядного процессора RISC с частотой 50 МГц на ядре ARM7. Имеет 10 Кбайт оперативной памяти, 280 Кбайт для программ. Объем памяти ввода-вывода составляет 600 байт для ПЛК63-М и 360 байт для ПЛК63-L. Энергонезависимая флеш-память – 448 Кбайт. Часы реального времени могут независимо работать в течение 3-х месяцев без внешнего питания.

Устройство крепится на DIN-рейку, имеет корпус класса защиты IP20. Для питания контроллера подходит как постоянное, так и переменное

напряжение – от 150 до 300 В постоянного или от 90 до 264 В переменного напряжения. Потребляемая мощность не превышает 12 Вт при питании постоянным током и не более 18 Вт при питании переменным. Имеется встроенный источник вторичного питания с выходом на 24 В и на ток не более 180 мА.

Текстовый монохромный ЖКИ-дисплей 2×16 снабжен подсветкой. Для управления используется клавиатура на шесть кнопок: «Пуск/стоп», «Ввод», «Выход», «Альт», «Вниз», «Вверх».

Интерфейсы связи: DEBUG RS-232 (RJ-11), RS-485. Протоколы: OBEN, GateWay (протокол CODESYS), Modbus RTU/ASCII.

Устройство OBEN ПЛК63 имеет восемь универсальных аналоговых выходов для подключения датчиков сигналов, таких как термодары, сигналы тока, термосопротивления, датчики напряжения, сопротивления.

Дискретных входов имеется 8 с групповой гальванической изоляцией, с возможностью подачи сигнала с максимальной частотой 50 Гц и скважностью 2.

В наличии 6 выходных элементов, один из которых электромагнитное реле 4 А 220 В, остальные пять могут отличаться в модификациях: Р – э/м реле 4 А 220 В; И – ЦАП 4–20 мА; У – ЦАП 0–10 В (активный). Количество выводов может быть расширено по внутренней шине до восьми с помощью стандартного модуля расширения MP1. Имеется возможность расширения входов-выходов путем подключения дополнительных модулей ввода/вывода. Количество дискретных выводов может быть увеличено посредством подключения стандартного модуля OBEN MP1.

Среда программирования CODESYS. Кроме стандартных библиотек CODESYS, бесплатно поставляется библиотека функциональных блоков собственной разработки OBEN: блок управления 3-позиционными задвижками, ПИД-регулятор с автонастройкой и др.

Программируемый логический контроллер OBEN ПЛК73.

OBEN ПЛК73 (рисунок 1.6) – контроллер с НМІ для создания локальных систем автоматизации, выполненный в щитовом исполнении. Внешний вид контроллера OBEN ПЛК73 представлен на рисунке 1.6. Основные области применения контроллера – ЖКХ, ЦТП, ИТП, котельные, небольшие станки и прочее.

Прибор OBEN ПЛК73 имеет некоторое сходство с OBEN ПЛК63, однако внешне выполнен в щитовом корпусе со степенью защиты IP55 и дополнен шестью светодиодными индикаторами на лицевой панели. Клавиатура уже имеет девять кнопок вместо шести, а дисплей здесь четырехстрочный 4×16 . Два интерфейса опционально: первый интерфейс – RS-485, RS-232 или отсутствует; второй интерфейс – RS-485, RS-232 или отсутствует. Интерфейсы взаимодействуют в режимах Master, Slave.



Рисунок 1.6 – Контроллер ОВЕН ПЛК73

Аналоговые входы ОВЕН ПЛК73 соответствуют ОВЕН ПЛК63, дискретные входы предусматривают возможность подключения датчиков с выходами «сухой контакт», р-п-р- и п-р-п-транзистор, при этом частота ограничена значением в 15 Гц при скважности 0,5. Питание дискретных входов осуществляется напряжением 24 В. Выходы соответствуют ОВЕН ПЛК63, четыре из них имеют возможность установки ЦАП. Среда программирования – CODESYS 2.3 (версии 2.3.8.1 и старше).

Контроллеры ОВЕН ПЛК73 широко востребованы в пищевой промышленности, в машиностроении и металлообработке, в химической отрасли, на производстве строительных материалов, в нефтегазовой промышленности, а также при автоматизации в ЖКХ, в сельском хозяйстве.

Программируемый логический контроллер ОВЕН ПЛК100.

ОВЕН ПЛК100 (рисунок 1.7) – это моноблочный контроллер с дискретными входами/выходами для организации автоматизации малых систем.

Прибор ОВЕН ПЛК100 предназначен для управления средними и малыми объектами и для построения систем диспетчеризации. Устройство имеет компактный корпус для крепления на DIN-рейку, дискретные входы/выходы с удобными креплениями, а также последовательные порты (RS-232, RS-485) и Ethernet. Любой из встроенных интерфейсов позволяет расширить количества точек ввода/вывода посредством подключения внешних модулей. Питание осуществляется либо переменным током с напряжением 220 В, либо постоянным 24 В.



Рисунок 1.7 – Контроллер ОВЕН ПЛК100

Скорость работы дискретных входов достигает 10 кГц при использовании подмодулей счетчика. Интерфейсы (три последовательных порта и USB Device для программирования) работают независимо друг от друга.

Температурный диапазон – от минус 20 до плюс 70 °С.

Внутри устройства ОВЕН ПЛК100 есть встроенный аккумулятор, который в случае пропадания питания позволит перевести выходные элементы в безопасное состояние. Присутствуют и встроенные часы.

Кроме того, по любому из портов возможна работа с нестандартными протоколами, поэтому можно подключить любые счетчики: газовые, электросчетчики или водосчетчики – либо считыватели штрих-кодов и тому подобные приспособления.

Кроме ОВЕН ПЛК100, в серии представлены также ПЛК150 и ПЛК154, отличающиеся количеством дискретных входов (8, 6 и 4 соответственно) и типом дискретных выходов (реле и сдвоенные транзисторные ключи). Всего 12 сигнальных выходов с возможностью коммутации токов до 2 А. ПЛК150 и ПЛК154 имеют также аналоговые входы (50 Ом) и выходы (до 20 мА), в ПЛК150 четыре аналоговых входа и два аналоговых выхода, а в ПЛК154 – четыре аналоговых входа и четыре аналоговых выхода.

Контроллеры данной серии широко применяются при автоматизации инженерных систем зданий, в сельском хозяйстве, на производстве строительных материалов, в машиностроении, полиграфии, ЖКХ, в химической отрасли, в электроэнергетике и т. д.

Программируемые логические контроллеры ПЛК110/160.

Это линейка моноблочных программируемых контроллеров с дискретными входами/выходами и аналоговыми входами/выходами (ПЛК160) (рисунок 1.8), предназначенных для автоматизации систем средней сложности. Устройства идеально подходят для создания распределенных систем управления. Рекомендованы для систем HVAC, в сферах ЖКХ, ИТП, ЦТП, для АСУ водоканалов, для управления насосами и другим оборудованием.

а)



б)



Рисунок 1.8 – Контроллер ОВЕН ПЛК110 (а) и ОВЕН ПЛК160 (б)

Контроллеры могут использоваться для управления станками и механизмами, в пищевой и перерабатывающей промышленности, для управления работой упаковочных аппаратов, подходят для управления торговым оборудованием, климатическим оборудованием, а также в сфере производства стройматериалов.

Отличительной особенностью линейки является значительная вычислительная мощность (RISC-процессор, 32 разряда, 180 и 400 МГц) и ус-

вершенствованные высокоскоростные входы и выходы, а также обширные возможности для программирования.

Языки программирования ПЛК.

Для составления управляющих программ контроллеров используются специализированные технологические языки, доступные инженерам и технологам и максимально упрощающие процесс программирования. За последнее десятилетие появилось множество технологических языков, которые соответствуют стандарту МЭК-61131-3, разработанному Международной Электротехнической Комиссией, которая концентрирует все передовое в области языков программирования для систем автоматизации технологических процессов.

Этот стандарт требует от различных изготовителей ПЛК предлагать системы программирования, являющиеся одинаковыми по организации программирования, по структуре программы и по действию команд.

Стандарт специфицирует пять языков программирования:

- 1) Sequential Function Chart (SFC) – язык последовательных функциональных схем;
- 2) Function Block Diagram (FBD) – язык функциональных блоковых диаграмм;
- 3) Ladder Diagrams (LD) – язык релейных диаграмм;
- 4) Statement List (ST) – язык структурированного текста, язык высокого уровня;
- 5) Instruction List (IL) – язык инструкций, типичный ассемблер с аккумулятором и переходам по метке.

Язык LD или язык релейно-контакторных схем (РКС) похож на электрические схемы релейной автоматики. Поэтому инженерам, не знающим языков программирования, несложно написать программу.

Язык FBD – графический язык описания логических и аналоговых вычислений в очень простой и выразительной форме, напоминает создание схем на логических элементах.

Язык IL дословно – список инструкций. Каждая инструкция начинается с новой строки и содержит оператор и, в зависимости от типа операции, один и более операндов, разделенных запятыми.

Язык ST – текстовый язык высокого уровня, подобен языку программирования Паскаль.

Язык SFC – графический язык, ориентированный на описание взаимосвязанных состояний и действий системы.

В каждом из этих языков есть свои минусы и плюсы. Поэтому при выборе специалисты основываются в основном на личном опыте, хотя большинство программных комплексов дают возможность перекомпилировать уже написанную программу из одного языка в другой. Некоторые

задачи изящно и просто решаются на одном языке, а на другом придется столкнуться с некоторыми трудностями.

Наибольшее распространение в настоящее время получили языки LD, ST и FBD.

Большинство фирм-изготовителей ПЛК традиционно имеют собственные фирменные наработки в области систем инструментального программного обеспечения. Например, такие как система программирования Concept фирмы Schneider Electric или Step 7 фирмы Siemens.

Программный комплекс CoDeSys.

Открытость стандарта МЭК-61131-3 привела к созданию фирм, занимающихся исключительно инструментами программирования ПЛК.

Наибольшей популярностью в мире пользуется комплекс CoDeSys, разработанный фирмой 3S. Это универсальный инструмент программирования контроллеров на языках МЭК, не привязанный к какой-либо аппаратной платформе и удовлетворяющий всем современным требованиям.

Основные особенности:

- полноценная реализация МЭК языков;
- возможность встроенным эмулятором контроллера проводить отладку проекта без аппаратных средств. Причем эмулируется не некий абстрактный контроллер, а конкретный ПЛК с учетом аппаратной платформы;
- возможность создания встроенными элементами визуализации модели объекта управления и проведения отладки, т. е. возможность создавать человекомашинный интерфейс (HMI);
- очень широкий набор сервисных функций, ускоряющий работу программиста;
- русская версия программы и русская документация.

Порядок проведения лабораторной работы

При выполнении работы необходимо использовать руководство по эксплуатации изучаемого контроллера.

- 1 Изучить назначение, технические характеристики, состав и конструкцию контроллера.
- 2 Изучить устройство контроллера, назначение органов управления.
- 3 Изучить варианты подключения контроллера к компьютеру.
- 4 Изучить меры безопасной работы, порядок установки и подготовки контроллера к работе.
- 5 Изучить схему подключения исполнительных устройств и датчиков управляемого объекта к контроллеру.
- 6 Подготовить ответы на контрольные вопросы.
- 7 Составить отчет по работе.

Содержание отчета

Отчет по лабораторной работе должен содержать следующее.

- 1 Титульный лист установленного образца.
- 2 Цель работы.
- 3 Назначение и основные технические параметры контроллера.
- 4 Схемы подключения исполнительных устройств и датчиков к контроллеру.
- 5 Ответы на контрольные вопросы.

Контрольные вопросы

- 1 Какие имеются модификации изучаемого контроллера?
- 2 Какова длительность рабочего цикла контроллера и чем она определяется?
- 3 Какие датчики и исполнительные устройства могут подключаться к каналам ввода-вывода контроллера?
- 4 Что происходит при нажатии кнопки «Сброс»?
- 5 Что означает безопасное состояние выходов?
- 6 Каким образом можно определить состояние контроллера?
- 7 Где размещается управляющая программа пользователя?
- 8 Для чего предназначена Retain-память?
- 9 Какие языки используются для программирования контроллера?
- 10 Какие способы программирования предусмотрены в контроллере?
- 11 Какие устройства контроллера можно использовать в управляющих программах?
- 12 Какие интерфейсы обмена информацией имеются в контроллере?
- 13 Какой линии СОМ-порта компьютера используются при программировании контроллера?
- 14 Как осуществить подключение нескольких контроллеров для совместной работы?
- 15 Каким образом осуществляется подключение питания к контроллеру?

2 Лабораторная работа № 2. Изучение системы программирования логических контроллеров CoDeSys

Цель работы: изучение методики создания проектов, настройки среды программирования и приобретение навыков работы в системе программирования CoDeSys.

2.1 Общие сведения о системе программирования CoDeSys

CoDeSys – универсальная система разработки программ для программируемых логических контроллеров (ПЛК) на основе языков стандарта МЭК 61131-3. Название системы расшифровывается как Controller Development System. Система выпускается немецкой фирмой 3S-Smart Software Solutions GmbH.

Первая версия CoDeSys вышла в 1994 г., в настоящее время доступна третья версия пакета, в которой поддерживается объектно-ориентированное программирование. В состав комплекса программирования ПЛК входят две обязательные части: среда разработки программ и система исполнения. Среда разработки программ CoDeSys функционирует на персональном компьютере (ПК) и не имеет конкретной аппаратной привязки, поэтому с её помощью можно создавать программы для любых контроллеров. Система исполнения (CoDeSys SP) функционирует в контроллере и устанавливается его производителем. Она обеспечивает загрузку кода программы, ее исполнение, а также отладочные функции.

Среда разработки программ CoDeSys является бесплатной, и ее можно скачать с сайта производителя. Компания 3S лицензирует только системы исполнения. Существует русифицированная версия CoDeSys. Этот программный пакет может быть установлен на компьютер, работающий под управлением операционной системы Windows. Для загрузки написанной программы в какой-либо конкретный ПЛК необходим target-файл (файл целевой платформы), обычно поставляемый производителем контроллера.

В этом файле находится информация о ресурсах данного контроллера, в том числе о входах и выходах, типах и расположении данных в памяти и т. д. Для инсталляции target-файлов служит программа Install-Target, которая устанавливается на компьютер вместе с пакетом CoDeSys.

CoDeSys применяется для программирования многими фирмами, в частности, WAGO, ABB, Beckhoff и т. д. В России CoDeSys широко применяется фирмой Овен (Москва) и Пролог (Смоленск).

Одним из преимуществ CoDeSys является наличие в нем режима эмуляции, что позволяет отлаживать программы непосредственно на компьютере, не загружая их в контроллеры и не привлекая на стадии отладки работающее технологическое оборудование. Это свойство позволяет также широко использовать CoDeSys в учебном процессе.

В настоящее время существует некоммерческая организация CoDeSys Automation Alliance (CAA) – объединение компаний-производителей ПЛК, поддерживающих CoDeSys, в которое входят более семидесяти фирм.

2.2 Структура проекта CoDeSys

При создании (открытии) проекта в CoDeSys появляется окно проекта (рисунок 2.1), в верхней части которого находится меню, а под ним – панель инструментов с учетом выбранного языка. В левой части окна находится вертикальная панель, под которой расположены четыре вкладки. Справа, последовательно сверху вниз, находятся область объявления переменных, область кода выбранного языка и область сообщений об ошибках и предупреждениях при компиляции проекта.

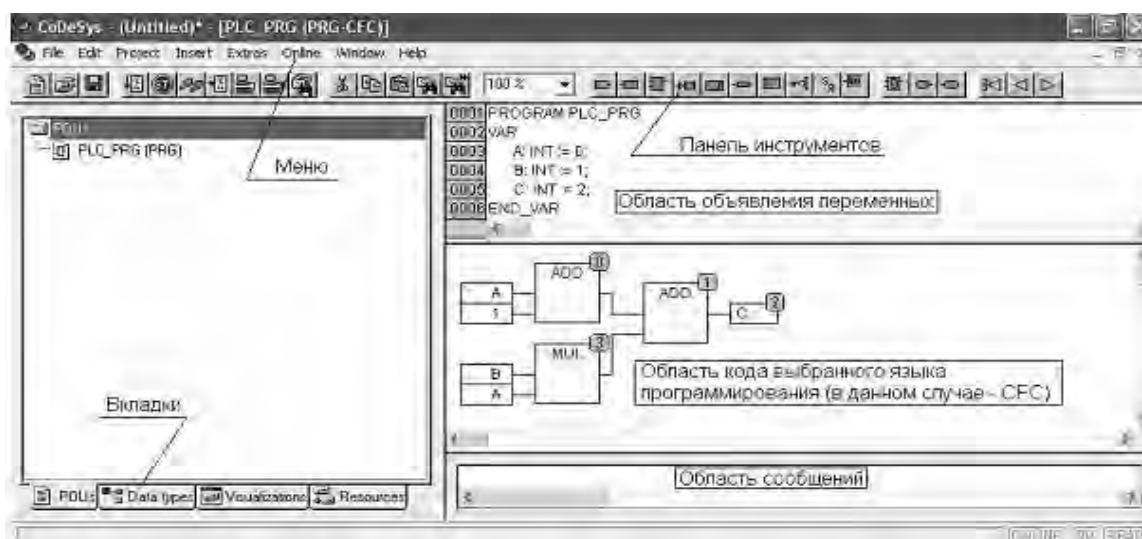


Рисунок 2. 1 – Окно проекта CoDeSys

Вкладки проекта CoDeSys отдельно показаны на рисунке 2.2.



Рисунок 2. 2 – Вкладки CoDeSys

Первая слева вкладка – POU, т. е. компоненты проекта. При входе в эту вкладку на находящейся над ней вертикальной панели (Object Organizer – список объектов) отображаются все компоненты проекта (на рисунке 2.1 компонент только один – программа (PRG) с именем PLC_PRG).

При щелчке левой кнопкой мыши по имени POU можно войти в редактор кода, соответствующий языку данного компонента. При щелчке правой кнопкой мыши по имени POU или по пустому полю вертикальной панели появляется меню, с помощью которого можно добавить новый компонент POU, переименовать или удалить уже имеющийся и т. д.

Следующая вкладка – Data types (Типы данных) – предназначена для определения пользовательских типов данных, например, перечислимых типов данных и структур.

Вкладка Visualizations (Визуализация) предназначена для создания и просмотра операторских интерфейсов (визуализаций). При входе в эту вкладку на левой панели отображаются имеющиеся визуализации. При щелчке левой кнопкой мыши по имени визуализации можно войти в нее, при щелчке правой кнопкой появляется контекстное меню.

Вкладка Resources (Ресурсы) позволяет выполнить действия, связанные с ресурсами контроллера. Через нее можно войти в «Настройку целевой платформы», выполнить конфигурирование входов и выходов ПЛК, изменить прошивку ПЛК, войти в «Менеджер библиотек» и т. д. Работа с вкладками Resources рассматривается далее.

Часть пунктов меню, находящегося в верхней части окна проектов, универсальна для всех языков, встроенных в CoDeSys, часть из них специфична для конкретного языка. Рассмотрим здесь наиболее важные пункты, общие для всех языков. В пункте меню «Проект» (Project) имеются два первых элемента «Компилировать» (Build) и «Компилировать все» (Rebuild all) (рисунок 2.3).

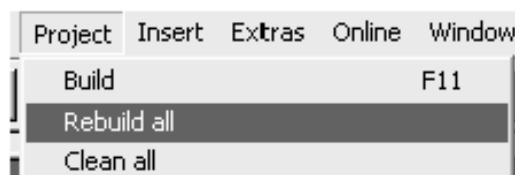


Рисунок 2. 3 – Пункт меню «Проект»

При использовании элемента «Компилировать» компилируется только новая часть проекта. При обращении к элементу «Компилировать все» компилируется весь проект. В случае успешной компиляции в области сообщений появляется запись «0 Error(s), 0 Warning(s)» – нет ошибок и нет предупреждений. Проект, в котором есть ошибки, не может быть запущен до их полного исправления. При наличии предупреждений проект может работать.

По окончании компиляции можно выполнить загрузку и пуск программы через раздел меню «Онлайн» (Online) (рисунок 2.4). Если предполагается загружать программу в ПЛК, имеющем связь с ПК, то сначала с помощью элемента «Подключение» (Login) производится их логическое соединение, затем можно загрузить программу, используя команду «Загрузить». Если программа является новой, выполняется действие «Загрузить новую программу». Если компьютер не соединен с ПЛК, то в меню «Онлайн» можно выбрать «Режим эмуляции» и выполнить

программу без загрузки в ПЛК. Программа не начинает работу до принудительного пуска.



Рисунок 2.4 – Пункт меню «Онлайн»

Пуск программы происходит при обращении к элементу «Старт» (Run) меню «Онлайн» (рисунок 2.5) или к одноименной иконке панели инструментов.



Рисунок 2.5 – Пункт меню «Онлайн», запуск программы

Останов программы происходит при обращении к элементу «Стоп» (Stop) меню «Онлайн» (рисунок 2.6) или к одноименной иконке панели инструментов. В этом случае работа программы прекращается, но компьютер не отключается от контроллера. При необходимости отключиться от компьютера используется подпункт «Отключение» (Logout) (см. рисунок 2.6). Необходимо отметить, что если отключить компьютер от контроллера без останова программы, то в контроллере программа продолжит выполняться, что является нормальным режимом для производственных условий.

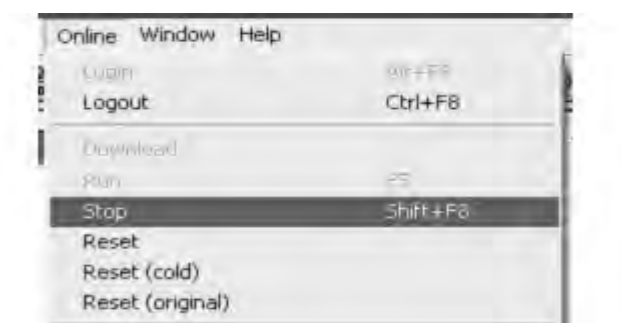


Рисунок 2.6 – Пункт меню «Онлайн», останов программы и отключение от контроллера

Пункт «Сброс» (Reset) перезапускает программу, заново инициализируя все переменные, за исключением RETAIN. Если начальные

значения переменных определены, они будут присвоены переменным, если нет – переменным присваиваются стандартные значения по умолчанию (например, 0 для целых типов). Данный сброс аналогичен выключению и включению питания ПЛК.

При выборе пункта «Сброс (холодный)» (Reset (cold)) выполняются те же действия, что и при команде «Сброс» (Reset), и дополнительно присваиваются начальные значения переменным, хранящимся в энерго-независимой области памяти RETAIN.

Пункт «Сброс (заводской)» (Reset (original)) удаляет программу пользователя, восстанавливая состояние контроллера, в котором он поступает с завода-изготовителя.

В некоторых ситуациях требуется изменить значения переменных, используемых в программе, не прекращая ее работы. Для этого необходимо открыть вкладку POU и кликнуть левой кнопкой мыши переменную в работающей программе. Появится диалоговое окно, представленное на рисунке 2.7. В поле «Новое значение» (New Value) вводится требуемое значение. Затем следует обратиться к пункту «Записать значения» меню «Онлайн» или нажать F7 (рисунок 2.8), после чего в программе будет учитываться новое значение переменной.

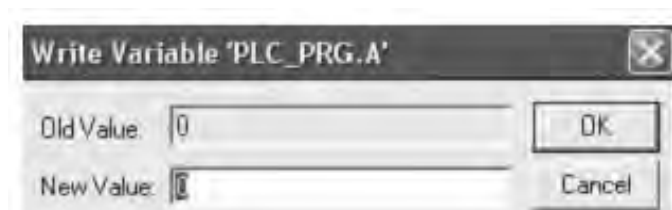


Рисунок 2.7 – Диалоговое окно для записи значений переменных

Write Values	Ctrl+F7
Force Values	F7
Release Force	Shift+F7
Write/Force-Dialog	Ctrl+Shift+F7

Рисунок 2.8 – Пункт меню «Онлайн», запись значений переменных

2.3 Создание нового проекта

Проект в CoDeSys создается в виде файла xxx.pro с помощью компонентов организации программ, обозначаемых как POU (Program Organization Unit). Чтобы создать новый проект, следует войти в пункт меню File/New (Файл/Создать). Появится диалоговое окно Target Settings (Настройка целевой платформы, рисунок 2.9), где необходимо указать тип контроллера, для которого создается проект. Если проект не предпо-

лагается загружать в контроллер, то выбирается пункт None. Чтобы тип контроллера появился в списке, необходимо предварительно установить его target-файл, как это рассмотрено далее.



Рисунок 2.9 – Настройка целевой платформы

После выбора целевой платформы на экране появится окно New POU – Новый программный компонент (POU) (рисунок 2.10), в котором требуется указать имя нового POU, выбрать его тип и язык реализации.

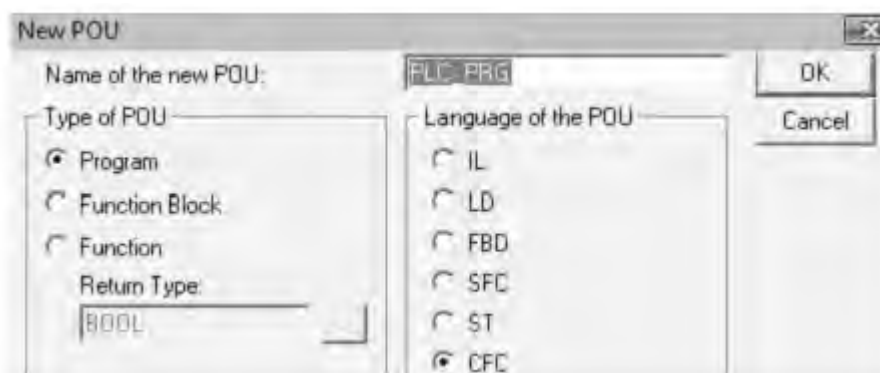


Рисунок 2.10 – Выбор вида программного компонента

В CoDeSys существует всего три типа POU: «Программа» (Program), «Функциональный блок» (Function Block) и «Функция» (Function). Если создаваемый проект предполагается загружать в контроллер, то в нем должен быть хотя бы один компонент «Программа» с именем PLC_PRG (это имя предлагается по умолчанию). Все остальные POU можно называть произвольно с учетом следующих требований: имя должно состоять только из латинских букв, цифр или знаков подчеркивания, при этом первым символом в имени должна быть буква. Имена POU в составе одного проекта не должны повторяться. Те же требования предъявляются к именам переменных и визуализаций.

Если создаваемый компонент является функцией, то нужно выбрать тип возвращаемого функцией значения и указать его в поле Return Type (тип возвращаемого значения). По умолчанию задается тип BOOL, т. е. логический. Можно нажать на кнопку слева от поля Return Type и выбрать тип из открывающегося списка. После ввода необходимой информации

следует нажать кнопку «ОК», при этом диалоговое окно New POU закроется и откроется редактор кода выбранного языка программирования.

2.4 Переменные. Объявление переменных

Согласно общепринятому определению, переменная – это поименованная или иным образом адресуемая область памяти. Имя (или адрес) этой области можно использовать для доступа к находящимся в ней данным, т. е. к значению переменной.

У любой переменной имеется имя, тип и область действия; переменные в CoDeSys могут быть локальными и глобальными. Областью действия локальной переменной является тот программный компонент, внутри которого она объявлена. Если переменная объявлена внутри функции, то область её действия – эта функция и т. д. Область действия глобальной переменной – весь проект.

В CoDeSys все переменные, связанные со входами или выходами контроллера, являются глобальными по умолчанию. В прочих случаях лучше не пользоваться глобальными переменными без необходимости, т. к. сложно проследить, как изменяется значение данной переменной при работе программы в целом. В CoDeSys все используемые переменные должны быть объявлены. Локальные переменные объявляются в области объявления переменных редактора кода (рисунок 2.11).

CoDeSys отслеживает переменные, используемые в тексте и, обнаружив новые необъявленные переменные, предлагает объявить их. При этом выводится окно объявления переменной «Declare Variable» (см. рисунок 2.11).

В этом окне необходимо ввести имя и тип переменной. По умолчанию указывается тип BOOL. Если переменная имеет другой тип, следует нажать на кнопку справа от поля Type. Откроется окно «Ассистент ввода» (Input assistant), в котором можно выбрать требуемый тип (рисунок 2.12). В поле Class можно оставить значение по умолчанию (при создании пользовательского функционального блока в этом поле необходимо указать, является ли переменная входной, выходной или внутренней, см. ниже). В поле Address ничего вводить не следует.

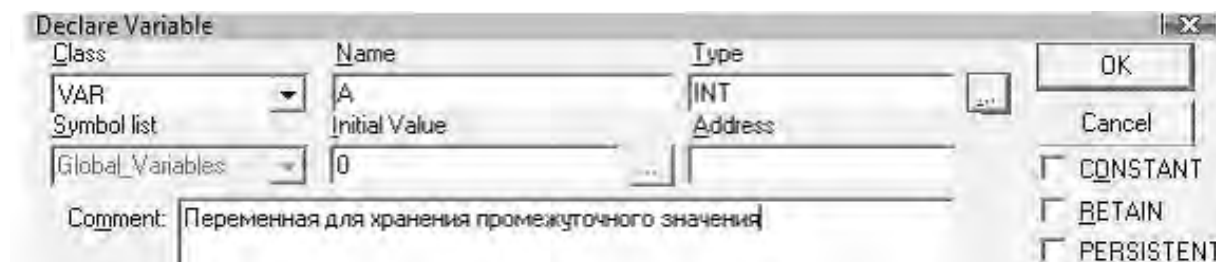


Рисунок 2.11 – Окно объявления переменной



Рисунок 2.12 – Ассистент ввода

В поле Initial value (начальное значение) вводится начальное значение переменной, его можно не изменять. В поле Comment можно ввести комментарий. После нажатия кнопки ОК введенные данные отображаются в области объявления переменных редактора кода (рисунок 2.13).



Рисунок 2.13 – Область объявления переменных редактора кода

Содержание области объявления переменных можно создать или заменить вручную.

Объявления локальных переменных для компонента PROGRAM PLC_PRG всегда пишутся между ограничителями VAR и END_VAR.

Глобальные переменные объявляют на вкладке «Ресурсы», а не в области редактора кода. Имя глобальной переменной не должно нигде повторяться, в том числе и для локальной переменной, иначе глобальная переменная будет не видна в том компоненте, где объявлена одноименная локальная, что может привести к неправильной работе программы. В таблице 2.1 приведены основные типы данных, используемые при составлении программ в среде CoDeSys, и их краткие описания.

Таблица 2.1 – Основные типы данных, применяемые при составлении программ

Имя типа	Описание типа
INT	Целые числа в диапазоне от – 32768 до 32767
REAL	Числа с плавающей точкой, 32 байта
BOOL	Логический тип данных, значения TRUE или FALSE
WORD	Целые числа в диапазоне от 0 до 65535.
TIME	Время (Например, T#5m30s0ms – 5 мин 30 с)
STRING	Строка от 1 до 255 символов

2.5 Компоненты организации программ

Проект в CoDeSys может содержать три вида компонентов организации программ (POU): программу (Program), функциональный блок (Function Block) и функцию (Function).

Программа может возвращать несколько значений, ее можно вызывать из других программ или из функциональных блоков, из функций программу вызывать нельзя.

Функции и функциональные блоки могут быть созданы пользователем самостоятельно или взяты из библиотек. Функция отображает множество значений входных параметров на один выход, а функциональный блок – на множество выходов. Чтобы создать пользовательский компонент, следует вызвать вкладку POU, щелкнуть правой кнопкой мыши по левой вертикальной области и в появившемся контекстном меню (рисунок 2.14) выбрать пункт Add object (добавить объект). Появится окно, показанное на рисунке 2.15 (Выбор нового программного компонента).



Рисунок 2.14 – Добавление нового пользовательского компонента

Если новый пользовательский компонент – это функция, то необходимо указать не только имя, тип нового компонента и язык реализации, но и тип возвращаемого значения (Return type). В примере на рисунке 2.15 тип возвращаемого значения Real. При нажатии кнопки справа от поля ввода появится окно ассистента ввода, в котором можно выбрать требуемый тип; кроме того, его можно ввести вручную. После создания нового программного компонента появляется окно редактора кода для выбранного языка. В рассматриваемом примере для функции с именем My_function и типом возвращаемого значения Real область объявления переменных показана на рисунке 2.16. Здесь в области объявления переменных для пользовательской функции есть две подобласти: VAR_INPUT...END_VAR и VAR...END_VAR. При объявлении переменных необходимо указать, к какому классу (VAR или VAR_INPUT) относится объявляемая переменная (рисунок 2.17).

Если переменная относится к классу VAR_INPUT, то эта переменная будет соответствовать одной из независимых переменных данной функции, т. е. одному из входных параметров. Если переменная относится к классу VAR, то она будет внутренней переменной данного компонента.

Выходу, т. е. собственно значению функции, соответствует переменная с именем, совпадающим с именем функции.

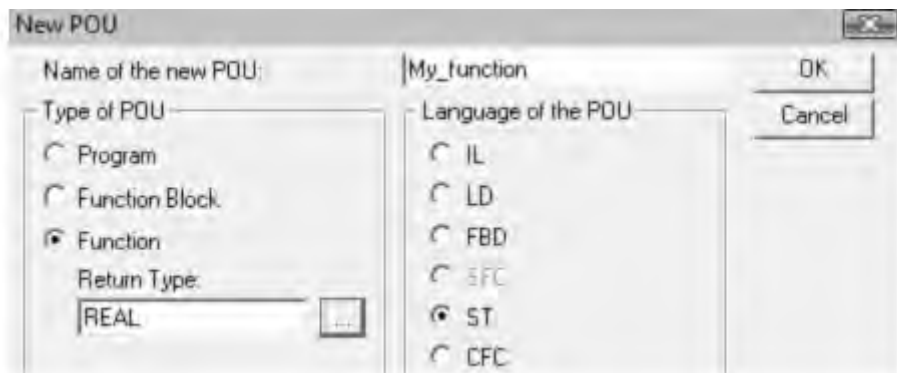


Рисунок 2.15 – Создание нового программного компонента Function

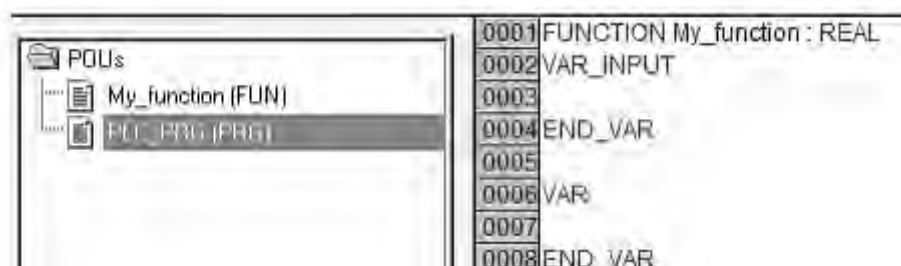


Рисунок 2.16 – Область объявления переменных для компонента Function

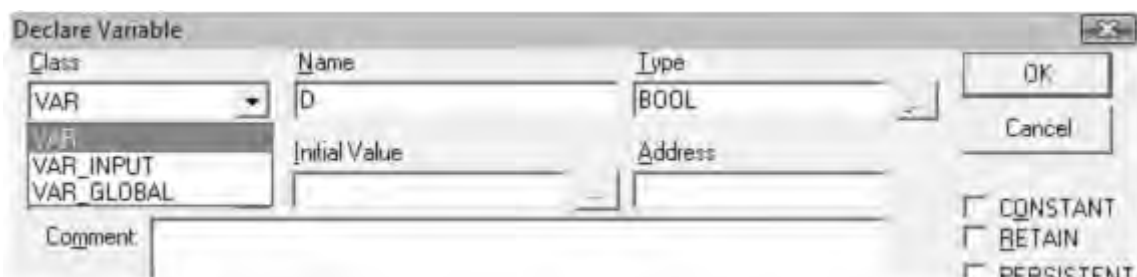


Рисунок 2.17 – Окно объявления переменных для компонента Function

Если новый компонент является функциональным блоком, то тип возвращаемого значения не указывается, т. к. у функционального блока может быть не один выход. Переменные, используемые внутри пользовательского функционального блока, могут относиться к классам VAR, VAR_INPUT и VAR_OUTPUT. Переменные из класса VAR_INPUT соответствуют входам блока, переменные из класса VAR_OUTPUT – выходам блока, а переменные из класса VAR – внутренним переменным.

Имена входов пользовательского блока будут соответствовать именам переменных класса VAR_INPUT, имена выходов – именам переменных класса VAR_OUTPUT. При этом совершенно необязательно,

чтобы имена переменных, связываемых с входами или выходами функционального блока извне, совпадали с именами его входов и выходов.

Все применяемые в программе пользовательские функциональные блоки и функциональные блоки из подключаемых библиотек должны иметь уникальные имена. К ним предъявляются такие же требования, как и к именам переменных. Имя функционального блока записывается на месте трех вопросительных знаков, находящихся над блоком.

Функциональный блок может быть написан на любом языке, встроенном в CoDeSys, функция – на любом языке, кроме SFC.

2.6 Подключение библиотек

Для подключения библиотек нужно зайти во вкладку «Ресурсы» (Resources) и левой кнопкой мыши открыть «Library Manager» (Менеджер библиотек). Далее кликом правой кнопки мыши в поле библиотек открыть контекстное меню, войти в пункт «Добавить библиотеку» (Ins) и в появившемся окне Open (Открыть) выбрать требуемый файл типа xxx.lib. Окно менеджера библиотек показано на рисунке 2.18.

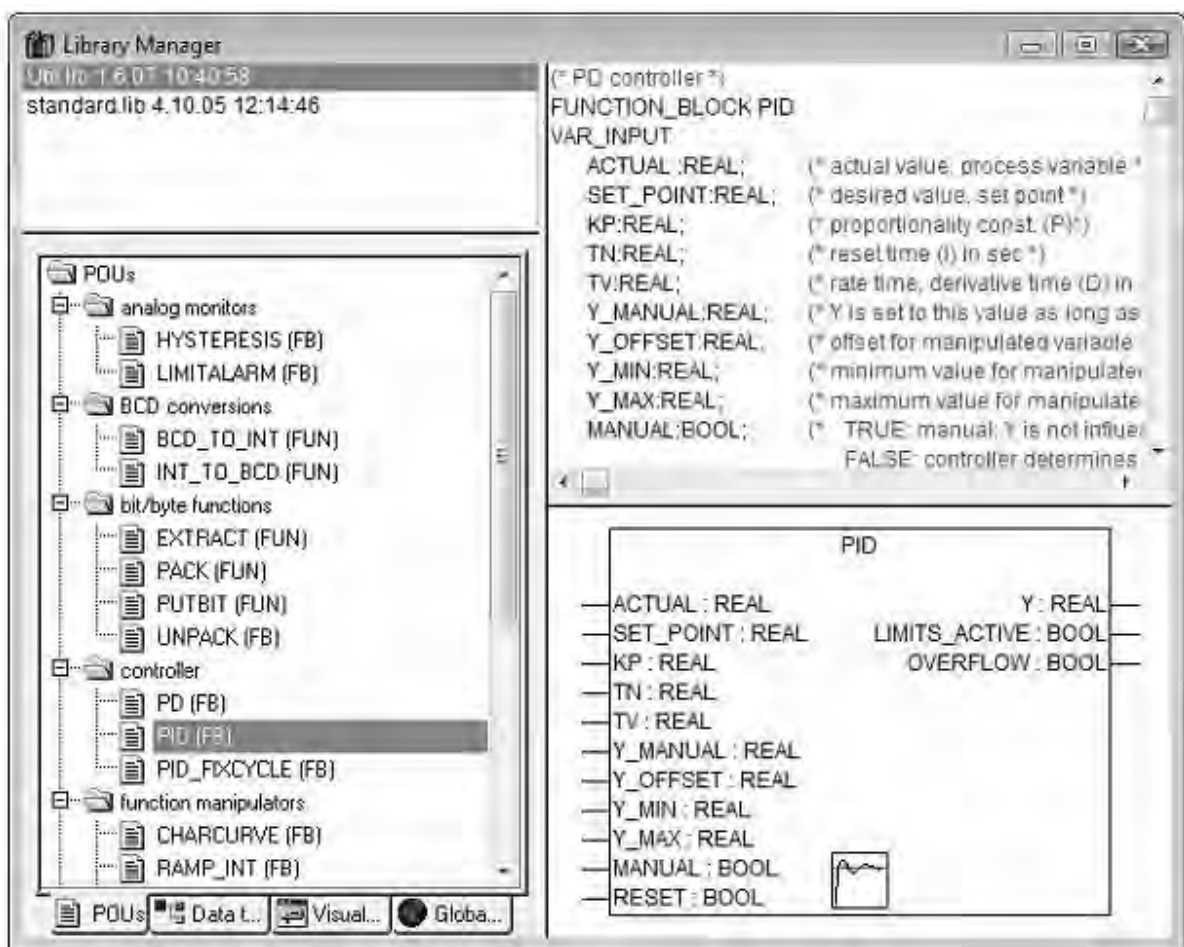


Рисунок 2.18 – Окно менеджера библиотек

Оно разделено на четыре части. В левой верхней части выводится список уже подключенных библиотек. В левой нижней части представлен список компонентов отмеченной библиотеки (на рисунке 2.18 это Util.lib). В правой верхней части выводится список локальных переменных отмеченного компонента (на рисунке 2.18 это блок PID-регулятора). В правой нижней части показан отмеченный компонент со всеми входами и выходами. Таким образом, это окно можно использовать как справочный материал по данной библиотеке.

Если щелкнуть правой кнопкой мыши под списком библиотек, то появится контекстное меню (рисунок 2.19). В нем предусмотрены следующие действия: добавить еще одну библиотеку (Ins), удалить (Del) или узнать свойства отмеченной библиотеки (Properties). В свойствах, например, указывается имя файла библиотеки, место его расположения и дата последнего изменения. Если файл библиотеки был перемещен, то с помощью пункта Properties можно указать его новый адрес.

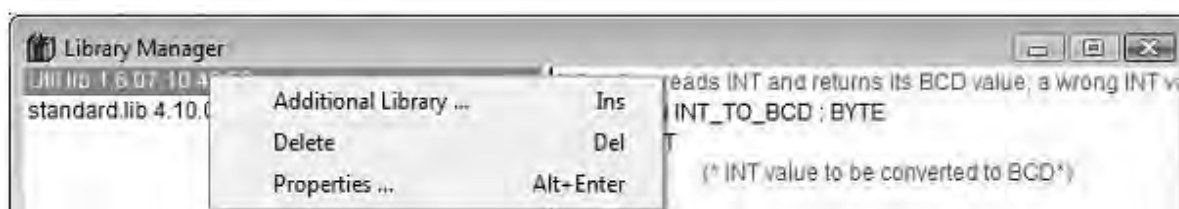


Рисунок 2.19 – Контекстное меню для менеджера библиотек

2.7 Стандартные операции

В системе CoDeSys над переменными и константами можно выполнять арифметические и логические операции. В таблице 2.2 приведены основные арифметические операции и функции; в таблице 2.3 – логические функции и функции выбора, которые не требуют подключения библиотек. Для более детального изучения используемых библиотек, функциональных блоков и функций следует обращаться к справочной системе CoDeSys.

Таблица 2.2 – Функции и арифметические операции

FBD, CFC	ST	Описание
ADD	+	Сложение
SUB	–	Вычитание
DIV	/	Деление
MUL	*	Умножение
SIN	SIN()	Синус угла

Окончание таблицы 2.2

FBD, CFC	ST	Описание
COS	COS()	Косинус угла
TAN	TAN()	Тангенс угла
ASIN	ASIN()	Арксинус числа
ACOS	ACOS()	Арккосинус числа
ATAN	ATAN()	Арктангенс от числа
EXP	EXP()	Экспонента от числа
LN	LN()	Натуральный логарифм числа
LOG	LOG()	Десятичный логарифм числа
ABS	ABS()	Модуль числа

Таблица 2.3 — Логические функции и функции выбора

FBD, CFC	ST	Описание
AND	AND()	Логическое «И» (конъюнкция)
OR	OR()	Логическое «ИЛИ» (дизъюнкция)
NOT	NOT()	Логическое «НЕ» (инверсия)
MIN	MIN()	Выбор минимального из двух чисел
MAX	MAX()	Выбор максимального из двух чисел
GT	>	Значение выхода TRUE, если $x1 > x2$
LT	<	Значение выхода TRUE, если $x1 < x2$
GE	\geq	Значение выхода TRUE, если $x1 \geq x2$
LE	\leq	Значение выхода TRUE, если $x1 \leq x2$
EQ	=	Значение выхода TRUE, если $x1 = x2$

2.8 Установка target-файлов

С помощью комплекса CoDeSys можно программировать любой контроллер, в котором его производителем установлена система исполнения CoDeSys SP. Кроме того, для данного типа контроллера должен быть target-файл от фирмы-производителя. В этом файле находится информация о ресурсах контроллера.

При необходимости написать для данного типа контроллеров проект с помощью CoDeSys следует установить соответствующий target-файл. Это делается с помощью программы InstallTarget, которую можно найти по следующему пути: Пуск/Все программы/3S Software/CoDeSys V2.3/InstallTarget (рисунок 2.20). При запуске программы InstallTarget появится диалоговое окно, показанное на рисунке 2.21. В левом поле «Possible Targets» указаны target-файлы, которые можно установить

(с жесткого или съемного диска), их можно выбрать, нажав кнопку «Open». В правом поле «Installed Targets» указаны установленные файлы.



Рисунок 2.20 – Расположение программы InstallTarget

Перед тем, как проинсталлировать файл, необходимо в поле «Installation directory» указать папку, в которую он будет установлен. Можно выбрать папку, нажав кнопку справа от поля ввода и отметив требуемую директорию в открывшемся окне «Choose Installation Directory» (выбор папки для инсталляции – рисунок 2.22).

На рисунке 2.21 для инсталляции выбран target-файл PLC150.U-L, который будет установлен в папку с адресом «C:\CoDeSys\Targets ...». Для начала установки необходимо выбрать target-файл в левом окне и нажать кнопку «Install». В результате инсталляции данный файл появится в поле «Installed Targets» (рисунок 2.23). Если требуется удалить target-файл, следует выбрать его в правом окне и нажать кнопку «Remove».

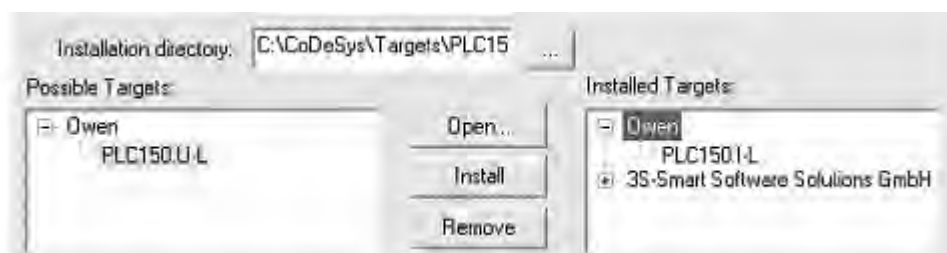


Рисунок 2. 21 – Окно программы InstallTarget

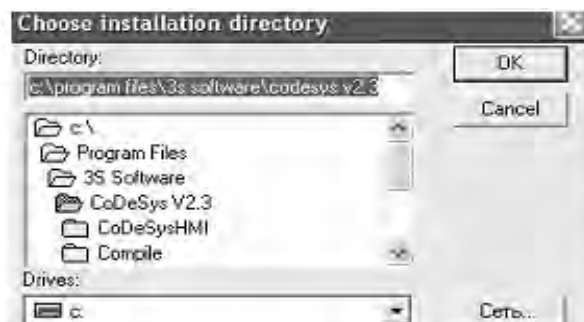


Рисунок 2. 22 – Выбор папки для инсталляции target-файл

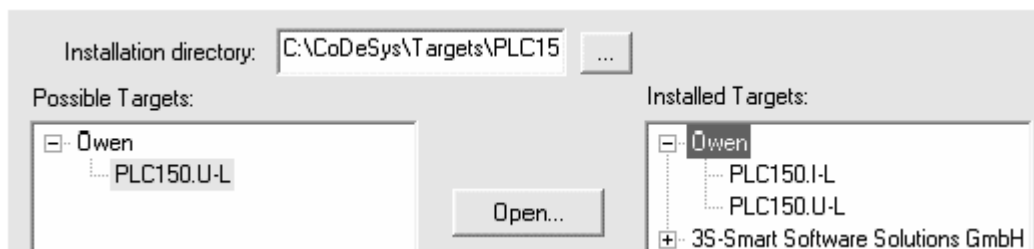


Рисунок 2.23 – Результат инсталляции target-файла

2.9 Настройка связи компьютера с контроллером

Программирование промышленных логических контроллеров ОВЕН осуществляется с помощью персонального компьютера, который соединяется с контроллером специальным кабелем посредством одного из стандартных интерфейсов.

Программируемые контроллеры подключаются к компьютеру тремя способами – с помощью COM-порта (интерфейс RS-232) кабелем KC1, входящим в комплект поставки контроллера, с помощью интерфейса локальной вычислительной сети Ethernet восьмижильным кабелем на основе витых пар, с помощью USB-интерфейса (последнее имеют не все модели) стандартным кабелем типа А-В.

Настройка подключения происходит после создания нового проекта в среде CoDeSys.

После запуска программы CoDeSys на экране появится основное окно системы, в котором можно открыть уже созданный проект или создать новый.

Создать новый проект можно, нажав на крайнюю левую кнопку панели инструментов (под главным меню) или выбрав «Файл» – «Новый». После этого появится окно выбора целевой платформы для создания проекта, где нужно выбрать, для какого вида контроллеров будет создаваться проект.

После выбора целевой платформы (например, PLC150.U-L) и подтверждения выбора кнопкой «ОК» в текущем окне на экране появится новое окно, в котором будут содержаться основные параметры и настройки выбранной платформы ПЛК (адреса сегментов памяти, тактовая частота процессора, тип процессора, количества входов и выходов, значения некоторых системных переменных). Некоторые параметры пользователь может изменять.

После подтверждения настроек контроллера и выбора языка программирования можно настраивать связь с контроллером. Для этого выбрать в главном меню команду «Онлайн» – «Параметры связи», в результате появится окно (рисунок 2.24).

На рисунке 2.24 показаны уже существующие настройки. Если

таковых не имеется и в иерархическом дереве слева есть только одна строка «'localhost' via Tcp/Ip», то нужно создать новое подключение.

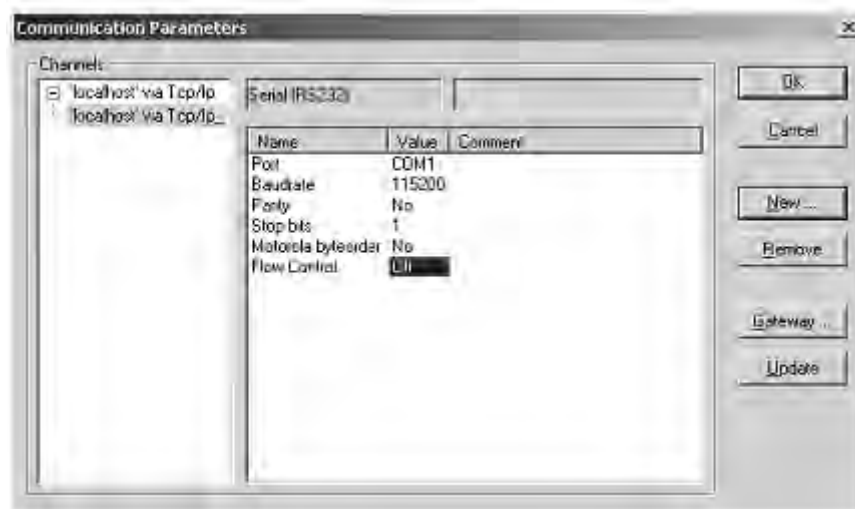


Рисунок 2. 24 – Окно настройки связи с ПЛК

Для этого надо нажать кнопку «New», в правой части окна и в появившемся окне (рисунок 2.25) выбрать вид соединения с контроллером (в данном случае – «Serial (RS232)»), затем нажать кнопку «OK».



Рисунок 2.25 – Выбор соединения компьютера с ПЛК

В средней части окна настроек связи появится список параметров для выбранного соединения (см. рисунок 2.25). Значения параметров будут установлены по умолчанию. При необходимости для выбранной целевой платформы контроллера параметры должны быть скорректированы в соответствии с руководством по эксплуатации.

Проверить связь с контроллером можно, создав простейшую программу и записав ее в контроллер.

Перед загрузкой программы в контроллер нужно проверить его подключение к компьютеру с помощью кабеля (соединение интерфейса RS-232 на лицевой панели ПЛК с COM-портом ПК), при необходимости

подключение производится с соблюдением требований, указанных в руководстве по эксплуатации.

2.10 Конфигурация памяти ввода/вывода контроллера

Конфигурация ПЛК определяется составом аппаратных средств контроллера, совокупностью модулей, каналов ввода/вывода и значением их параметров. Информация о конфигурации хранится в области памяти ввода/вывода контроллера.

Область памяти ввода/вывода ПЛК (%I и %Q) включает дискретные и аналоговые входы и выходы, модули расширения (в том числе организующие обмен информацией между ПЛК и отдельными приборами и устройствами, связанными с ПЛК по сети). Внешние устройства обмениваются данными с пользовательской программой ПЛК также через эту область памяти. Размер памяти ввода/вывода определяется типом лицензии CoDeSys контроллера ОВЕН ПЛК.

На основе описания конфигурации ПЛК CoDeSys проверяет правильность задания МЭК адресов, используемых в программах, на их соответствие фактически имеющимся аппаратным средствам.

В процессе создания и отладки проекта необходимо настроить конфигурацию входов, выходов и интерфейсов связи ПЛК с внешними устройствами.

Настройка конфигурации выполняется в окне редактора «Конфигурация ПЛК (PLC Configuration)» ПО CoDeSys.

Для входа в режим редактирования конфигурации ПЛК следует перейти на вкладку «Ресурсы». В дереве ресурсов следует выбрать пункт «Конфигурация ПЛК» (PLC Configuration). В рабочей области главного окна откроется окно редактора (рисунок 2.26).

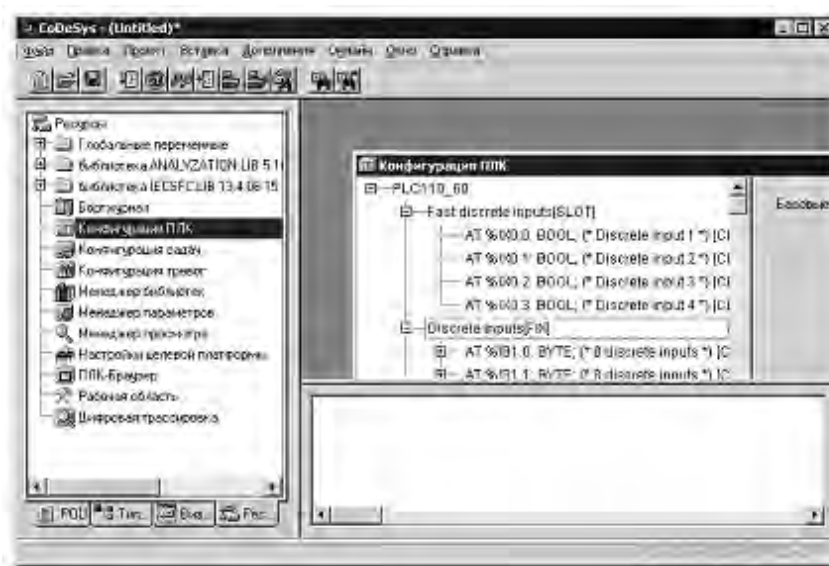


Рисунок 2.26 – Вход в режим «Конфигурация ПЛК» (PLC Configuration)

Окно редактора конфигулятора ПЛК разделено на две части. В левой части окна отображается дерево конфигурации, древовидная структура, отображающая ресурсы контроллера.

Структура и компоненты дерева определяются файлом настроек целевой платформы конфигурации, но могут быть изменены пользователем CoDeSys.

В дереве конфигурации отображаются следующие элементы:

- модуль;
- канал;
- битовый канал.

Модуль (элемент конфигурации) – независимая единица аппаратных средств. Модуль включает набор каналов ввода-вывода. Модуль (как и каждый отдельный канал) может иметь параметры. Каждый тип модуля имеет уникальный идентификатор.

Канал – это собственно данные ввода-вывода. Как правило, модуль имеет фиксированный набор каналов или подмодулей. Каждый канал имеет определенный МЭК тип и адрес. Для каждого канала автоматически выделяется определенное пространство памяти. Каждый канал имеет уникальный в пределах данной конфигурации ПЛК идентификатор.

Битовый канал – это идентификатор отдельного бита в канале.

В дереве конфигурации задается распределение адресов входов/выходов контроллера, что определяет привязку проекта к аппаратным средствам.

В правой части окна отображаются диалоги конфигурации, доступные для текущего (выделенного) элемента дерева конфигурации. Диалоги отображаются в виде одной или нескольких табличных вкладок (рисунок 2.27).

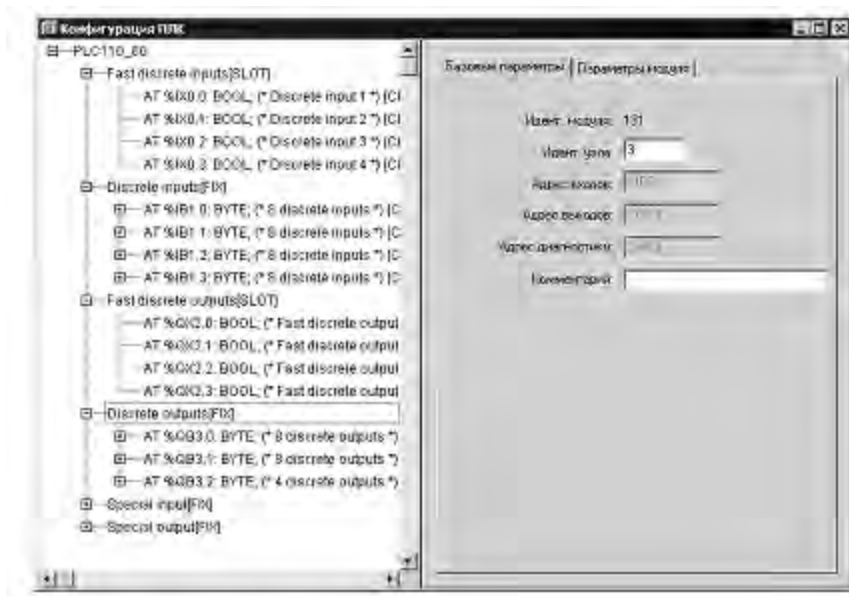


Рисунок 2.27 – Окно режима «Конфигурация ПЛК» (PLC Configuration)

В полях, расположенных на вкладках диалогов, задаются требуемые значения параметров канала или модуля. Значение параметра устанавливается до компиляции проекта.

2.11 Привязка переменных к входам и выходам ПЛК

У ПЛК могут быть аналоговые и дискретные входы и выходы, информация о которых хранится в его target-файле. Для конфигурирования входов и выходов нужно войти во вкладку «Ресурсы», а в ней – в «Конфигурацию ПЛК». Входы и выходы ПЛК описаны в таблице 2.4.

Таблица 2.4 – Входы и выходы ПЛК

Английский	Русский (примеры)
Discrete input	Дискретный вход (сигналы от датчиков-реле)
Discrete output	Дискретный выход (сигналы включения нагревателя или ИМ постоянной скорости)
Unified signal sensor	Аналоговый вход (сигналы аналоговых датчиков, например, преобразователей давления)
Analog output	Аналоговый выход (сигнал к позиционеру)

Рассмотрим для примера конфигурирование аналогового входа, к которому подключен термометр сопротивления. Для настройки следует кликнуть кнопкой мыши выбранный аналоговый вход (нумерация сверху вниз), выбрать в появившемся меню «Заменить элемент» и выбрать «RTD sensor» (рисунок 2.28).

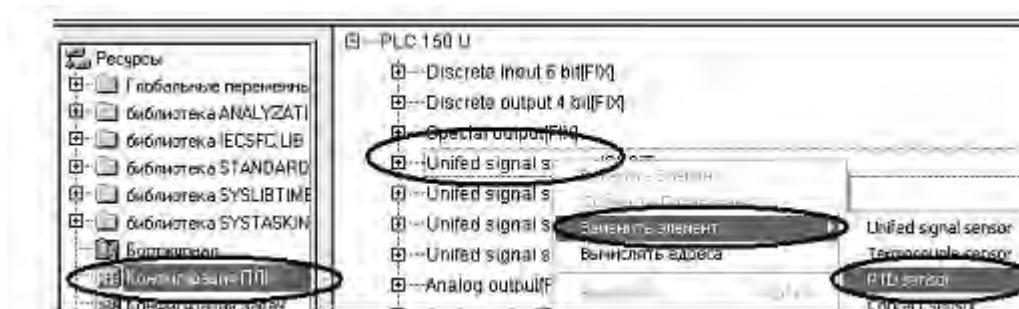


Рисунок 2.28 – Конфигурирование аналоговых входов. Выбор типа датчика

Далее следует кликнуть «АТ» в первой строке настраиваемого входа и ввести имя переменной (PV, рисунок 2.29, а). Эта переменная будет глобальной. Затем кликнуть вид датчика («RTD sensor»), открыть «Параметры модуля» (рисунок 2.29, б) и выбрать тип датчика (r428_50). Здесь можно вводить поправки в трех выбранных точках с линейной

интерполяцией. Кроме того, указывается время цикла измерения переменной по настраиваемому входу.

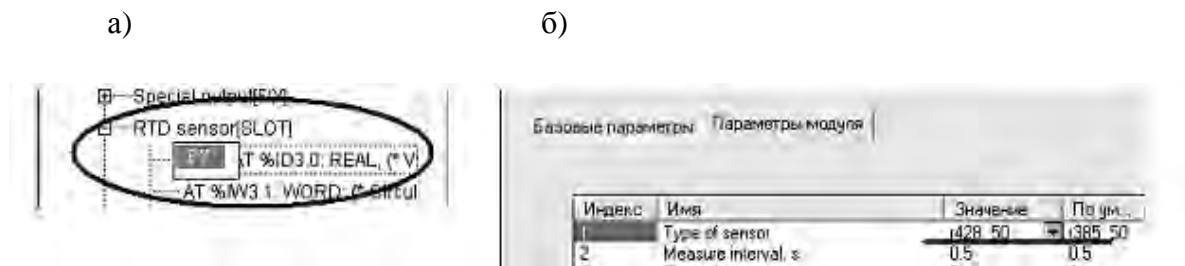


Рисунок 2. 29 – Конфигурирование аналоговых входов. Продолжение

Рассмотрим добавление и конфигурирование ШИМ. Для этого следует открыть дискретные выходы, открыть контекстное меню кликом кнопкой правой кнопки мыши и выбрать «Вставить Pulse-Width Modulator». Появится группа «Pulse-width modulator» под всеми дискретными входами. Далее нужно открыть группу ШИМ и присвоить имя переменной (на рисунке 2.30 – to_pwm), затем войти во вкладку «Параметры модуля» и указать в первой строке выход, к которому подключен ШИМ (следует иметь в виду, что нумерация в CoDeSys начинается с нуля), во второй строке – период ШИМ в сотнях микросекунд, в третьей – минимальную длительность импульса в сотнях микросекунд.



Рисунок 2.30 – Конфигурирование ШИМ

Порядок проведения лабораторной работы

При выполнении работы дополнительную информацию можно получить из «Руководства пользователя по программированию ПЛК в CoDeSys 2.3» или из справочной системы среды программирования CoDeSys.

1 Изучить состав системы программирования CoDeSys, назначение компонентов.

2 Изучить порядок установки системы и инсталляции target-файлов.

3 Изучить состав проекта, порядок создания нового проекта, создания и редактирования компонентов проекта.

- 4 Изучить способы и последовательность объявления переменных.
- 5 Изучить методы отладки проекта.
- 6 Изучить последовательность подключения контроллера к компьютеру и настройки связи.
- 7 Изучить порядок конфигурации модулей ПЛК.
- 8 Подготовить ответы на контрольные вопросы.
- 9 Запустить систему программирования CoDeSys 2.3.
- 10 Создать новый проект, подключить, если необходимо, библиотеки Standart.lib и Util.lib, ознакомиться с их содержимым.
- 11 Составить простейшую управляющую программу, запустить ее и проверить выполнение в режиме эмуляции.
- 12 Составить отчет по работе.

Содержание отчета

Отчет по лабораторной работе должен содержать следующее.

- 1 Титульный лист установленного образца.
- 2 Цель работы.
- 3 Описание структуры проекта и основных правил его создания.
- 4 Описание компонентов ROU, которые могут содержаться в проекте.
- 5 Описание операндов, используемых в системе программирования.
- 6 Описание классов переменных, используемых в компонентах проекта.
- 7 Ответы на контрольные вопросы.
- 8 Вывод по лабораторной работе.

Контрольные вопросы

- 1 Какие области содержит окно проекта в системе CoDeSys?
- 2 Какая информация отображается в статусной строке окна проекта в системе CoDeSys?
- 3 Какие файлы формируются при создании проекта и его компиляции?
- 4 Как выбрать целевую платформу и каким образом ее настроить?
- 5 Как выбрать язык программирования при создании проекта и можно ли его изменить в процессе работы над проектом?
- 6 Как подключить библиотеку к проекту?
- 7 Что такое экземпляр функционального блока и чем функция отличается от функционального блока?
- 8 Каким образом можно открыть проект и загрузить его в контроллер?
- 9 Какие методы отладки программ используются в системе CoDeSys?
- 10 Какие типы данных используются в языках программирования системы CoDeSys?
- 11 Каким способом объявляются переменные в языках программирования системы CoDeSys?

- 12 Какая информация содержится в target-файлах?
- 13 Как переменную связать с конкретным входом контроллера?
- 14 Какие параметры интерфейса RS-232 устанавливаются при настройке связи?
- 15 Какие модули могут входить в состав изучаемого контроллера?
- 16 Какие подчиненные подмодули можно подключить к модулю дискретного ввода?
- 17 Какие параметры можно редактировать при конфигурации модулей дискретного ввода-вывода?
- 18 Как осуществить подключение энкодера к изучаемому контроллеру?
- 19 Как осуществляется управление звуковым излучателем контроллера из программы?
- 20 Как изменить значение длительности рабочего цикла контроллера?

3 Лабораторная работа № 3. Программирование логических контроллеров на языке LD в системе CoDeSys

Цель работы: изучение принципов составления прикладных программ для промышленных логических контроллеров (ПЛК) на языке LD пакета CoDeSys. Приобретение навыков программирования на языке LD в системе CoDeSys.

3.1 Общие сведения

Язык лестничных диаграмм LD (Ladder Diagram) или язык релейно-контактных схем (РКС) является достаточно популярным в силу своей наглядности и позволяет решать широкий круг задач комбинационной и событийно-управляемой логики.

Язык LD – графический язык, в котором программа выглядит, как релейная схема в стандарте промышленной автоматизации. Две вертикальные линии слева и справа образуют линии питания. Между ними располагаются контактные цепи в виде горизонтальных линий по аналогии с обычными электрическими цепями релейной автоматики (по общему виду программы и дано название языка программирования). Слева по линии располагаются контакты (соответствуют входным переменным логического типа и дискретным входам). Справа – катушки реле (соответствуют выходным переменным логического типа и дискретным выходам).

Каждому контакту соответствует логическая переменная. Если переменная имеет значение ИСТИНА, то контакт считается замкнутым, если переменная имеет значение ЛОЖЬ, то контакт считается разомкнутым.

Каждой катушке также соответствует логическая переменная. Если

контактная цепь от левой линии до катушки состоит из замкнутых контактов, то реле считается включенным и соответствующей переменной присваивается значение ИСТИНА, иначе реле выключается и соответствующей переменной присваивается значение ЛОЖЬ. Если катушка инверсная (обозначается символом (/)), тогда в соответствующую логическую переменную копируется инверсное значение.

Предполагается, что контакты можно соединять в любом порядке и последовательности, определяя логику работы цепи, а катушки – только параллельно, как в релейных схемах подобных устройств.

Цепь из двух последовательно соединенных контактов соответствует логической операции «И», а цепь из двух параллельно соединенных – соответствует логической операции «ИЛИ». Операции «НЕ» соответствует нормально замкнутый контакт, который размыкает цепь (фактически дает логический ноль) при включении (подаче логической единицы), и наоборот.

Кроме последовательного и параллельного соединения нормально разомкнутых или замкнутых контактов и катушек, язык LD позволяет:

- включать фиксируемые Set / Reset-катушки;
- осуществлять переходы по цепям;
- включать в цепи функциональные блоки;
- управлять работой блоков по входам EN;
- записывать комментарии.

Программа на языке LD обрабатывается циклически слева направо, сверху вниз.

Порядок ввода, редактирования и отладки управляющих программ на языке LD в системе CoDeSys подробно описан в «Руководстве пользователя по программированию ПЛК в CoDeSys 2.3».

3.2 Пример программирования ПЛК на LD

Для того чтобы запускать электродвигатель в прямом и обратном направлениях, применяется реверсивная схема управления на магнитном пускателе (рисунок 3.1).

Подобную схему управления электродвигателем можно реализовать на базе ПЛК. Для этого к дискретным входам контроллера необходимо подключить кнопки управления, к выходам – через промежуточные реле пускатели. Принципиальная схема подключения элементов к условному контроллеру представлена на рисунке 3.2.

Входные и выходные сигналы ПЛК и соответствующие им переменные приведены в таблице 3.1.

Управляющая программа на языке LD, обеспечивающая запуск электрического двигателя в прямом и обратном направлениях, т. е. функционально соответствующая схеме на рисунке 3.1, представлена на рисунке 3.3.



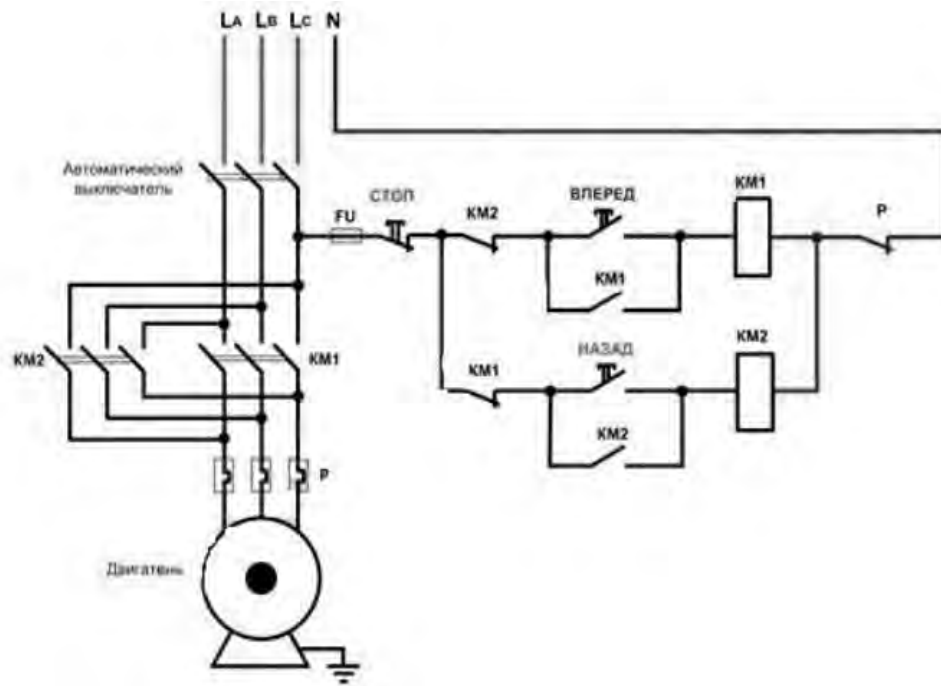


Рисунок 3.1 – Принципиальная схема работы реверсивного пускателя

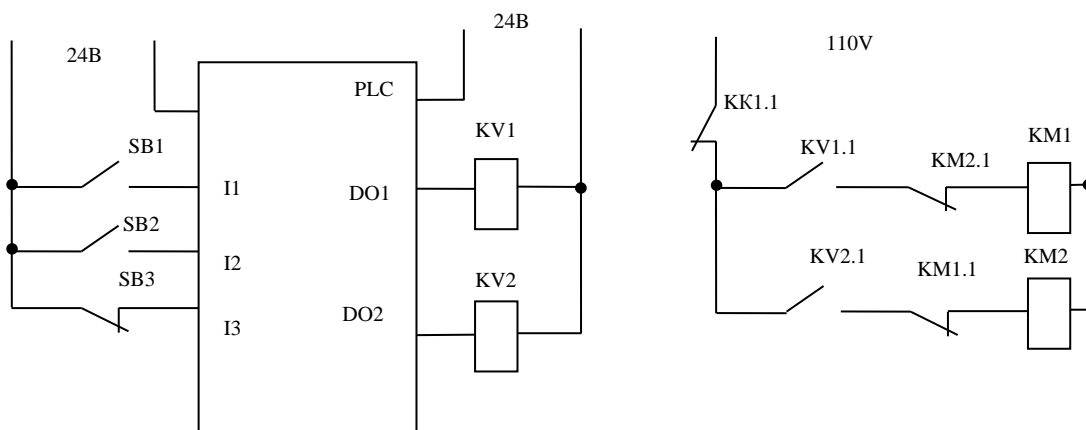


Рисунок 3.2 – Принципиальная схема подключения ПЛК

Таблица 3.1– Таблица входных и выходных сигналов ПЛК

Номер входа	Подключаемый элемент	Тип сигнала	Символьное обозначение	Адрес
DI1	SB1	BOOL	VPERED	%IX0.0
DI2	SB2	BOOL	NAZAD	%IX0.1
DI3	SB3	BOOL	STOP	%IX0.2
DO1	KM1	BOOL	FRW	%QX1.0
DO2	KM2	BOOL	REV	%QX1.1

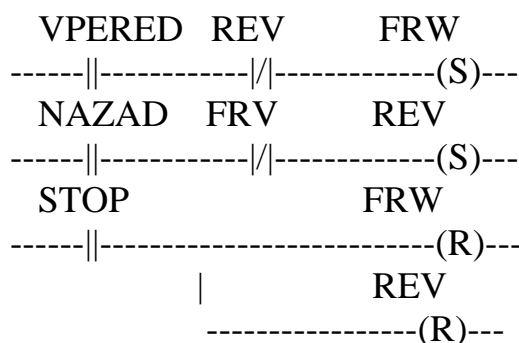


Рисунок 3.3 – Управляющая программа для ПЛК

Задание к лабораторной работе.

Используя схему управления асинхронным двигателем, составить программу на языке LD, обеспечивающую управление по заданному алгоритму. Отладить программу в режиме эмуляции. Записать в контроллер и проверить ее работу.

Варианты заданий

1 Кратковременное нажатие кнопки 1 включает двигатель ВПЕРЕД и звуковой сигнал, повторное нажатие отключает звуковой сигнал. Нажатие кнопки 2, если двигатель был ОСТАНОВЛЕН, включает вращение НАЗАД, иначе осуществляет РЕВЕРС. Нажатие кнопки 3 осуществляет ОСТАНОВ, если двигатель вращался, или включает двигатель ВПЕРЕД.

2 Кратковременное нажатие кнопки 1 включает двигатель ВПЕРЕД, последующие нажатия кнопки 1 РЕВЕРСИРУЮТ вращение. Кратковременное нажатие кнопки 2 включает двигатель НАЗАД, повторное нажатие – ОСТАНОВ. Нажатие кнопки 3 – ОСТАНОВ и выдача звукового сигнала, повторное нажатие – отключение звукового сигнала.

3 Кратковременное нажатие кнопки 1 включает двигатель ВПЕРЕД, если двигатель был ОСТАНОВЛЕН, иначе включает звуковой сигнал. Кратковременное нажатие кнопки 2, если двигатель вращается, – ОСТАНОВ, повторное нажатие – включение НАЗАД. Нажатие кнопки 3 – ОСТАНОВ и отключение звукового сигнала.

4 Кратковременное нажатие кнопки 1 включает двигатель ВПЕРЕД или ОСТАНАВЛИВАЕТ, если двигатель вращается. Нажатие кнопки 2 включает двигатель НАЗАД и включение звукового сигнала, повторное нажатие отключение звукового сигнала. Нажатие кнопки 3 – ОСТАНОВ.

5 Если двигатель ОСТАНОВЛЕН, кратковременное нажатие кнопки 1 включает двигатель ВПЕРЕД, нажатие кнопки 2 – включает двигатель НАЗАД, последующие нажатия кнопок 1 или 2 осуществляют РЕВЕРС. Нажатие кнопки 3 – ОСТАНОВ включение звукового сигнала, повторное нажатие – отключение звукового сигнала.

6 Если двигатель ОСТАНОВЛЕН, кратковременное нажатие кнопки 1 или 2 включает двигатель ВПЕРЕД. Повторное нажатие кнопки 1 – ОСТАНОВ и выдача звукового сигнала. Кратковременное нажатие кнопки 2, если двигатель вращается ВПЕРЕД, – включение двигателя НАЗАД. Нажатие кнопки 3 – ОСТАНОВ или отключение звукового сигнала.

Порядок проведения лабораторной работы

При выполнении работы необходимо дополнительно использовать «Руководство пользователя по программированию ПЛК в CoDeSys 2.3».

1 Изучить основные правила составления управляющих программ на языке LD в системе CoDeSys.

2 Изучить порядок ввода редактирования и отладки управляющих программ на языке LD в системе CoDeSys.

3 Подготовить ответы на контрольные вопросы.

4 Изучить задание к лабораторной работе.

5 Разработать принципиальную схему управления двигателем на базе изучаемого программируемого логического контроллера.

6 Составить управляющую программу, реализующую управление двигателем в соответствии с заданием.

7 Проверить работу управляющей программы в режиме эмуляции.

8 Записать программу в память контроллера и проверить ее выполнение.

9 Составить отчет по работе.

Содержание отчета

Отчет по лабораторной работе должен содержать следующее.

1 Титульный лист установленного образца.

2 Цель работы.

3 Список инструкций языка программирования LD.

4 Принципиальная схема управления двигателем на базе изучаемого программируемого контроллера.

5 Таблица входных и выходных сигналов.

6 Листинг программы.

7 Ответы на контрольные вопросы.

8 Вывод по лабораторной работе.

Контрольные вопросы

1 Какие функциональные блоки могут вставляться в цепи языка LD системы CoDeSys?

2 Каким образом организуются переходы в управляющих программах, написанных на языке LD?

- 3 Как обозначается элемент, соответствующий дискретному входу или выходу контроллера в языке программирования LD?
- 4 Как вставить комментарии в управляющей программе на языке LD?
- 5 Какие позиции может занимать курсор при составлении программы на языке программирования LD?
- 6 Как изменить размеры элементов в редакторе LD?
- 7 Как изменить наименование элемента в редакторе LD?
- 8 Как изменить последовательность элементов в цепи программы на языке LD?
- 9 Как определить состояние элемента при отладке программы на языке LD?
- 10 Как выполняется программа на языке LD в пошаговом режиме?

Список литературы

- 1 **Петров, И. В.** Программируемые контроллеры. Стандартные языки и инструменты / И. В. Петров. – Москва : СОЛОН-Пресс, 2003. – 256 с.
- 2 Руководство пользователя по программированию ПЛК в CoDeSys 2.3. – Смоленск: Пролог, 2008. – 452 с.
- 3 **Минаев, И. Г.** Программируемые логические контроллеры. Практическое руководство для начинающего инженера / И. Г. Минаев, В. В. Самойленко. – Ставрополь : АРГУС, 2009. – 100 с.
- 4 Общие сведения о CoDeSys [Электронный ресурс]. – Режим доступа: <http://www.3s-software.ru/publications>. – Дата доступа: 06.05.2017.
- 5 Каталог продукции фирмы ОВЕН [Электронный ресурс]. – Режим доступа: <http://www.owen.ru/catalog>. – Дата доступа: 16.05.2017.