ПРОГРАММИРУЕМЫЕ
КОНТРОЛЛЕРЫ
ОВЕН

Программирование
ОВЕН ПЛК63 и ПЛК73
Руководство пользователя

Программирование программируемых логических контроллеров ОВЕН ПЛК 63 и ПЛК 73

Руководство пользователя

Версия 1.4

Москва 2010

Содержание

Введение 5
1 Использование документа 5
1.1 Структура документа 5
1.2 Термины и аббревиатуры 6
1.3 Нормативно-справочная документация 6
1.4 Аппаратно программные требования к оборудованию и ПО, используемым при программировании ПЛК 7
1.4.1 Требования к оборудованию 7
1.4.2 Требования к ПО 7
1.4.3 Требования к персоналу 7
2 Предварительный этап программирования ПЛК 8
2.1 Предварительный этап 8
2.1.1 Установка операционной системы 8
2.1.2 Установка ПО (среды программирования) CoDeSys 8
2.2 Выбор контроллера: размер памяти 9
3 Этапы создания пользовательской программы («проекта») ПЛК 10
3.1 Компоненты проекта 10
3.1.1 Программные компоненты проекта (POU) 10
3.2 Установка настроек целевой платформы (target-файла) 10
3.2.1 Способ 1 11
3.2.2 Способ 2 12
3.3 Запуск ПО CoDeSys. Главное окно программы 13
3.3.1 «Организатор объектов» 14
3.3.2 Проект. Выбор контроллера и языка программирования 15
3.3.3 Проект. Программные компоненты (POU) 23
3.3.4 Проект. Типы данных 23
3.3.5 Проект. Установка связи с ПЛК 24
3.4 Конфигурирование области ввода-вывода ПЛК 26
3.4.1 Расчет потребности ПЛК в памяти ввода/вывода 26
3.5 Визуализация 27
3.6 Сохранение проекта 28
3.7 Запуск пользовательской программы 29
3.8 Сохранение программы в памяти контроллера 29
4 Написание программы 30
4.1 Программные компоненты проекта 30
4.1.1 Программы 30
4.1.2 Функции 31
4.1.3 Функциональный блок 31
4.2 Использование переменных 32
4.2.1 Типы переменных 32
4.2.2 Объявление переменных 33
4.2.3 Типы данных 36
4.2.4 Подключение дополнительных программных модулей 37
4.2.5 Библиотеки программных компонентов для работы с ЖКИ ПЛК 39
4.2.6 Создание и использование дополнительных программных модулей 40
4.3 Использование ЖКИ 41
4.4 Отладка проекта 41
4.5 Многозадачность 42
4.5.1 Отладка 43

4.5.2 Точки останова 44
4.5.3 Пошаговое выполнение 44
4.5.4 Выполнение по циклам 44
4.5.5 Эмуляция 45
4.5.6 Бортжурнал (Log) 45
5 Использование сложных структур данных 46
5.1 Пользовательские типы данных 46
5.1.1 Массивы 46
5.1.2 Перечисления 46
5.1.3 Структуры 47
5.1.4 Указатели 47
6 Визуализация проекта 48
6.1 CoDeSys HMI 49
7 Конфигурирование контроллера 50
7.1 Конфигурация памяти ввода / вывода 50
7.1.1 Приемы редактирования конфигурации ПЛК 54
7.2 Задание времени цикла ПЛК 62
7.3 Фиксированные элементы (модули) конфигурации 63
7.3.1 DInputs (модуль дискретных входов) 63
7.3.2 Outputs (модуль выходных устройств) 65
7.3.3 AInputs (модуль аналоговых входов) 67
7.3.4 MR1_Outputs (модуль выходного устройства MP1) 68
7.3.5 Additional (модуль дополнительных параметров) 69
7.3.6 Versions (модуль «версии») 72
7.3.7 RS-485 (модуль настройки RS-485) 73
7.3.8 WorkInd (модуль настройки меню) 74
7.4 Добавляемые подэлементы (подмодули) конфигурации ПЛК (пользовательские параметры) 74
7.4.1 Подэлемент «SubMenu (Ветка)» 76
7.4.2 Пользовательские параметры 76
7.5 Размещение переменных в области памяти протокола Modbus при работе в режиме Slave-устройства 80
8 Работа с ЖКИ ПЛК 81
8.1 Рабочий режим индикации 83
8.1.1 Экранный подрежим индикации 83
8.1.2 Монопольный подрежим индикации 83
8.2 Конфигурационный режим индикации 84
8.2.1 Главное меню контроллера 84
8.2.2 Редактирование значений параметров 85
8.3 Отладочный режим индикации 86
8.4 Библиотеки программных компонентов для работы с ЖКИ 86
8.4.1 Библиотека Work_Mode.lib (работа с ЖКИ в рабочем экранном режиме) 87
8.4.2 Библиотека Ind_Mode.lib (управление режимами ЖКИ и вывод символьных строк на ЖКИ в монопольном рабочем режиме) 87
9 Обновление встроенного ПО микроконтроллера и Target-файлов 89
9.1 Определение актуальной версии ПО микроконтроллера 89
9.2 Обновление ПО микроконтроллера 89
9.2.1 Обновление ПО микроконтроллера с использованием утилиты «Перепрошивка ПЛК» 89
9.3 Обновление Target-файла 91
Приложение А. Интерфейс ПО CoDeSys 92
А.1 Элементы управления ПО CoDeSys 92

A.2 Основные режимы (Редакторы) ПО CoDeSys 97
Приложение Б. Перечень параметров конфигурации ПЛК63 99
Приложение В. Список EnumP 104
Приложение Г. Перечень поддерживаемых датчиков 106
Приложение Е. Использование OPC-сервера 107
E.1 Использование OPC-сервера 3S-Software 107
E.2.1 Установка и использование OPC-драйверов фирмы ОВЕН 108
E.2.1 Использование OPC-драйверов «ОВЕН» 109
Приложение Ж. Отображаемые на ЖКИ символы и их коды 111
Приложение Н. Нормативно-справочная документация 112
Лист регистрации изменений 113

Введение

В данном руководстве изложены основы процедуры создания рабочей программы для программируемых логических контроллеров «ОВЕН ПЛК 63» и «ОВЕН ПЛК 73»

1 Использование документа

1.1 Структура документа

Первая часть документа (разделы 1 - 3) содержит краткое описание последовательности операций, выполняемых в ходе создания рабочей программы.

Вторая часть документа (разделы 4 - 7) содержит базовую часть информации, требуемой для реализации программ ПЛК, решающих задачи любой сложности.

Третья часть документа (разделы 8 – 9) содержит дополнительную информацию, требуемую при решении определенных задач программирования ПЛК.

Полностью информация, требуемая для создания программ, содержится в Руководстве пользователя программного обеспечения (ПО) CoDeSys (см. приложение Н, [2]): создание рабочей программы для программируемого логического контроллера выполняется в ПО CoDeSys.

При первоначальном ознакомлении с руководством рекомендуется ознакомиться с содержанием настоящего «Введения» и разделов 1 - 3 документа. В дальнейшем рекомендуется обращаться к разделам второй части документами, содержащим достаточный для работы объем информации по конкретным вопросам.

Информация по установке, вводу в эксплуатацию, обслуживанию и устранению ошибок работы программируемых логических контроллеров содержится в документе «Руководство по эксплуатации ПЛК63» или «Руководство по эксплуатации ПЛК73» (см. приложение Н, [6, 7]).

Процедура программирования ПЛК включает следующие этапы:

1) Предварительный этап: установка операционной системы и ПО (среды программирования) CoDeSys (**Co**ntroller **De**velopment **Sys**tem).
2) Выбор контроллера. Установка требуемого файла настроек целевой платформы (target-файла).
3) Создание и отладка проекта.
4) Установление связи с контроллером. При установке связи ПО CoDeSys автоматически компилирует проект и загружает скомпилированный код в энергонезависимую память контроллера.
5) Запуск выполнения проекта (пользовательской программы ПЛК), проверка ее работоспособности и, при необходимости, отладка.
6) В случае корректной работы проекта (пользовательской программы ПЛК) – выполнение её при включении питания ПЛК.
   В случае некорректной работы проекта – возврат на этап 5 (в процессе отладки проекта перечисленные выше операции могут выполняться многократно).

Программирование ПЛК рекомендуется выполнять до монтажа контроллера на объекте, но можно выполнить его и после монтажа.

	Внимание! Фрагменты текста, выделенные в документе аналогично данному фрагменту, содержат критически важную информацию, на которую рекомендуется обратить особое внимание.
	Внимание! Фрагменты текста, выделенные в документе аналогично данному фрагменту, содержат важную информацию, на которую рекомендуется обратить особое внимание.

1.2 Термины и аббревиатуры

Определения основных терминов и расшифровка аббревиатур, используемых в тексте данного документа, приведены в таблице 1.1.

Таблица 1.1 – Термины и аббревиатуры

Термины и аббревиатуры	Определения и расшифровки
ЖКИ	Жидкокристаллический индикатор, расположенный на лицевой поверхности ПЛК
КЗ	Короткое замыкание
ОЗУ	Оперативное запоминающее устройство
ПК	Персональный компьютер
ПЛК	Программируемый логический контроллер
ПО	Программное обеспечение
Редактор	Режим выполнения ПО CoDeSys, предназначенный для записи и редактирования текста программных компонентов. При работе с различными языками программирования используются различные редакторы.
Ресурс	Режим выполнения ПО CoDeSys, предназначенный для работы с компонентами проекта, доступными на вкладке «Ресурсы» организатора объектов ПО CoDeSys (например, «Ресурс “Конфигурация ПЛК (PLC Configuration)”»)
ПЗУ	Постоянное запоминающее устройство
Файл настроек целевой платформы (Target file)	Файл, поставляемый производителем ПЛК и описывающий аппаратные и программные особенности конкретного ПЛК. Обеспечивает корректное взаимодействие программного обеспечения CoDeSys и программируемого логического контроллера.
Проект	Пользовательская программа программируемого логического контроллера, разрабатываемая в программном обеспечении CoDeSys. После отладки и загрузки в контроллер обеспечивает правильную работу контроллера.
ПО CoDeSys	Специализированное программное обеспечение, предназначенное для подготовки пользовательских программ программируемого логического контроллера (CoDeSys)
EEPROM	EEPROM: Сокращение для электрически стираемых (программируемых) постоянных запоминающих устройств (ПЗУ). EEPROM сохраняет свое содержимое даже при отключении питания.
OPC	OLE – object linking and embedding – for Process Control, объектное связывание и встраивание для контроля процессов. OPC – открытый для использования набор спецификаций, разработанный организацией OPC Foundation на основе технологий Microsoft COM/DCOM.
OPC DA	Спецификация Data Access (DA) OPC (см. OPC), которая позволяет читать и писать данные в прибор, организовывать подписку на данные и поредавать клиенту уведомление об обновлении данных
SCADA	Диспетчерское управление и сбор данных (англ. Supervisory Control And Data Acquisition). Программное обеспечение, выполняемое на ПК с целью получения и отображения данных в удобном для пользователя виде, с возможностью управления

1.3 Нормативно-справочная документация

Перечень нормативно-справочной и эксплуатационной документации, использованной в данном документе, приведен в приложении Н.

1.4 Аппаратно программные требования к оборудованию и ПО, используемым при программировании ПЛК

1.4.1 Требования к оборудованию

Программирование ПЛК производится с использованием персональных компьютеров (ПК) с характеристиками, определяемыми тем, что программирование выполняется с использованием ПО CoDeSys (**Co**ntroller **De**velopment **Sys**tem) производства компании «3S – Smart Software Solutions GmbH» и устанавливаемыми производителем ПО:

- Pentium IV, 2 ГГц
- 512 Мб ОЗУ (рекомендуется 1024)
- 500 Мб жесткий диск
- CD ROM привод
- Последовательный порт,

Выполнение соединения ПЛК с ПК для программирования ПЛК производится с помощью кабеля КС1, входящего в комплект поставки ПЛК. Подключение описано в документах «ОВЕН ПЛК63. Руководство по эксплуатации» и «ОВЕН ПЛК73. Руководство по эксплуатации» (см. приложение Н, [6, 7]).

1.4.2 Требования к ПО

Программирование ПЛК производится с использованием следующего программного обеспечения (ПО):

- Windows XP / Vista: операционная система, установленная на персональном компьютере (ПК) и необходимая для инсталляции, запуска и выполнения ПО CoDeSys.
- CoDeSys (**Co**ntroller **De**velopment **Sys**tem): программное обеспечение (среда программирования) производства компании «3S – Smart Software Solutions GmbH», работающее на персональном компьютере (ПК) и применяемое при подготовке пользовательских программ ПЛК. Рекомендуемая версия ПО – 2.3.9.9. Бесплатные обновления версий ПО CoDeSys доступны на сайтах
- Комплект файлов библиотек дополнительных программных модулей (см. раздел 4.2.4)
- Файл настроек целевой платформы (target-файл), соответствующий используемому контроллеру.

1.4.3 Требования к персоналу

Персонал, выполняющий программирование ПЛК, должен:

- Владеть приемами работы с графическим интерфейсом операционной системы и программного обеспечения.
- Владеть методикой программирования ПЛК с использованием ПО CoDeSys в объеме, изложенном в документе «Руководство пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [2]).
- Владеть методикой эксплуатации ПЛК в объеме, изложенном в документе «Руководство по эксплуатации ПЛК63» или «Руководство по эксплуатации ПЛК73» (см. приложение Н, [5, 6]).
- Владеть методикой программирования ПЛК в объеме, изложенном в настоящем документе.

2 Предварительный этап программирования ПЛК

В данном разделе описывается предварительный этап процедуры программирования ПЛК.

2.1 Предварительный этап

В данном разделе описываются предварительные этапы работы: установка операционной системы и установка ПО CoDeSys.

2.1.1 Установка операционной системы

Для инсталляции, запуска и выполнения ПО CoDeSys на ПК необходима установка операционной системы Windows XP / Vista. Установка операционной системы производится в соответствии с инструкциями, размещенными на дистрибутивном диске ОС. В данном документе установка операционной системы не рассматривается.

2.1.2 Установка ПО (среды программирования) CoDeSys

Установка ПО (среды программирования) CoDeSys (**Co**ntroller **De**velopment **Sys**tem) производится запуском программы-инсталлятора (файл Codesys_v2399.exe на дистрибутивном диске ПЛК) и выполнением инструкций, отображаемых в окнах программы.

Примечания

1) ПО CoDeSys бесплатно и не требует лицензирования (за исключением отдельных необязательных приложений).

2) При установке ПО CoDeSys следует обратить внимание на то, что процедура выбора языка работы ПО в окнах мастера установки ПО в процессе установки выполняется дважды; при этом при первом выборе (см. рисунок 2.1) русский язык отсутствует в списке доступных языков, при втором – присутствует (см. рисунок 2.2).

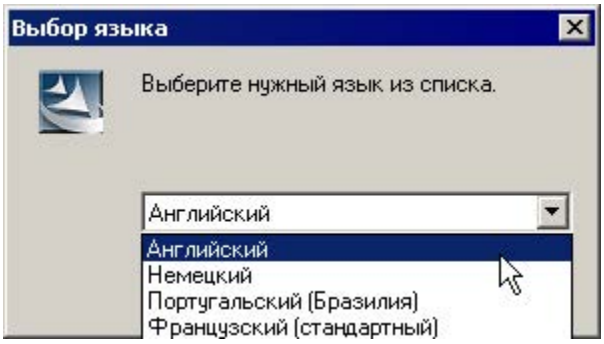


Рисунок 2.1 – Инсталляция ПО CoDeSys. окно «Выбор языка»

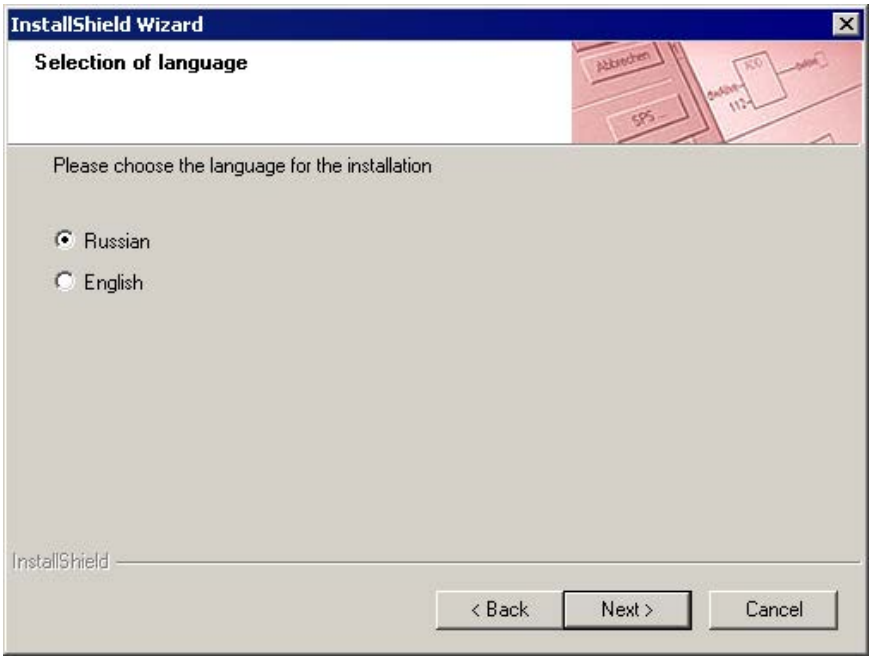


Рисунок 2.1 – Инсталляция ПО CoDeSys. окно «Selection Language»

2.2 Выбор контроллера: размер памяти

Связь ПЛК с внешними устройствами (модулями ввода-вывода и т.д.) производится по сети через специальную область памяти ПЛК: «Память ввода-вывода».

Размер памяти ввода-вывода определяется типом лицензии CoDeSys контроллера ОВЕН ПЛК. Тип лицензии указывается в маркировке конкретного ПЛК, в последнем знаке обозначения: «L» или «M».

Размеры памяти ввода-вывода приведены в таблице 2.1. Объем ОЗУ для хранения переменных программ – 10240 байт; объем энергонезависимой памяти – 448 байт

Таблица 2.1

Контроллер	%I (байт)			%Q (байта)		
	Всего	Занято	Доступно	Всего	Занято	Доступно
ПЛК63-M, ПЛК73-M	89	89	0	512	70	442
ПЛК63-L, ПЛК73-L	89	89	0	272	70	202

Внимание!
Выбор контроллера с требуемой лицензией следует делать до покупки ПЛК.

Для расчета необходимого объема памяти ввода / вывода и выбора требуемого типа лицензии можно воспользоваться методикой, изложенной в разделе 3.4.1 .

Задание конфигурации памяти ввода / вывода описано в разделе 7 настоящего документа.

3 Этапы создания пользовательской программы («проекта») ПЛК

В данном разделе содержится краткое описание последовательности операций, выполняемых при создании пользовательской программы («проекта»), описание интерфейса, основных режимов работы ПО CoDeSys и приемов работы в этих режимах.

Полностью процедуры создания и отладки пользовательской программы («проекта») ПЛК описаны в документе «Руководство пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [2]).

Описания процедур, специфичных для работы с ОВЕН ПЛК (например описания конфигурирования памяти ПЛК), приведены далее.

3.1 Компоненты проекта

Пользовательская программа («проект») ПЛК в ПО CoDeSys содержит программные компоненты (POU), типы данных, визуализации, ресурсы и библиотеки. сведения о ресурсах ПЛК и другую информацию. Проект сохраняется в одном файле (с расширением «pro» («наименование.pro»).

3.1.1 Программные компоненты проекта (POU)

Проект создается в ПО CoDeSys на любом из доступных языков программирования. Проект может состоять из одного или нескольких программных компонентов (POU, Program Organization Unit). Главная программа, выполняемая циклически, должна называться PLC_PRG.

К программным компонентам (POU) относятся функциональные блоки, функции и программы. Отдельные POU могут включать действия (подпрограммы).

Каждый программный компонент состоит из раздела объявлений и кода. Для написания всего кода POU используется только один из МЭК языков программирования (IL, ST, FBD, SFC, LD или CFC).

CoDeSys поддерживает все описанные стандартом МЭК 61131 компоненты. Для их использования достаточно включить в свой проект библиотеку дополнительных программных компонентов «standard.lib» (подробнее о библиотеках см. раздел 4.2.4).

При написании проекта следует иметь в виду, что POU проекта могут вызывать другие POU, но рекурсии (то есть обращения POU к себе самим) недопустимы.

Кроме того, в проекте могут быть явно определены несколько **задач** с различными условиями выполнения. Работа с задачами описана в разделе 6.7 «Конфигуратор задач (Task Configuration)» документа «Руководство пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [2]). См. также раздел 4.5 («Многозадачность») данного документа.

Приемы работы при написании программ и примеры программ представлены в документе «Первые шаги в CoDeSys» (см приложение Н [1]).

Выполнение программы начинается с программного компонента POU «PLC_PRG» и выполняется циклически.

Создание и отладка проекта производится в несколько этапов, перечисленных ниже.

3.2 Установка настроек целевой платформы (target-файла)

В данном разделе описывается этап выбора целевой платформы (т.е. программируемого контроллера).

Программа работы ПЛК не зависит от конкретной модели контроллера. Но на этапе установления связи с контроллером программа должна взаимодействовать с конфигурацией конкретного контроллера, содержащей определенным образом настроенные параметры его входов, выходов, интерфейсов связи, и некоторые другие характеристики.

Исходная информация о конфигурации ПЛК содержится в предварительных настройках целевой платформы (target-файле) контроллера, поставляемых производителем контроллера и размещенных в папке «Target-файлы» дистрибутивного диска. Настройки целевой платформы поставляются в виде набора файлов, основным (указываемым пользователем в процессе установки настроек) среди которых является Target-файл, имеющий расширение *.tnf, (Target Information File).

Файл содержит информацию о ресурсах конкретного ПЛК (о количестве и типах входов и выходов, интерфейсов, памяти, дополнительных устройств и т.д.), с которыми работает ПО CoDeSys.

Выбор целевой платформы (контроллера) производится установкой требуемого файла настроек целевой платформы (target-файла).

Внимание! Компания ОВЕН совершенствует производимые контроллеры и программное обеспечение и периодически предлагает пользователю обновленные версии встроенного ПО микроконтроллера и Target-файлов. Подробнее об обновлении встроенного ПО микроконтроллера и Target-файлов см раздел 9

Установка настроек целевой платформы (target-файла) может производиться двумя способами, описанными ниже.

Внимание! При выборе Target-файла в процессе установки настроек следует обратить внимание на то, что имя Target-файла не полностью совпадает с наименованием контроллера: в наименовании контроллера использована кириллица (например, ПЛК63), а в названии Target-файла – латиница (например, PLC63). Для каждой модификации ПЛК в поставку включен соответствующий Target-файл. Так, для ПЛК63 на дистрибутивном диске размещены Target-файлы PLC63-I и PLC63-M.

3.2.1 Способ 1

Установка настроек целевой платформы (target-файла) производится при помощи утилиты «InstallTarget». Утилита представляет собой компонент ПО CoDeSys и устанавливается на ПК совместно с ПО.

Порядок инсталляции Target-файлов таков:

1) Выбрать команду **Пуск | Программы | 3S Software | CoDeSys V2.3 | InstallTarget**.
2) В открывшемся окне утилиты «InstallTarget» окне (рисунок 3.1) – нажать кнопку «Open (Открыть)».
3) В открывшемся окне выбора файла – указать путь к инсталлируемому Target-файлу.
   В перечне доступных файлов следует выбрать требуемый: соответствующий модификации программируемого контроллера.
   В поле «Installation directory» отобразится выбранный путь к папке.
   В поле «Possible Taigets» отобразится список доступных Target-файлов.
   В поле «Installed Targets» отобразится перечень ранее установленных файлов (см. рисунок 3.1).
4) После того, как требуемый Taiget-файл выделен, следует нажать кнопку «Install». Taiget-файл будет инсталлирован на используемый ПК. В поле «Installed Targets» он отобразится в перечне установленных файлов (см. рисунок 3.1).
5) При необходимости (например, при ошибке в выборе файла), уста-

новленный Taiget-файл может быть деинсталлирован. Для этого его следует выделить в списке в поле «Installed Targets» и нажать кнопку «Remove». Файл будет деинсталлирован и перестанет отображаться в списке в поле «Installed Targets».

6) После завершения инсталляции требуемого Taiget-файла – нажать кнопку «Close». Окно утилиты «InstallTarget» закроется.

На этом процедура инсталляции Target-файла завершается.

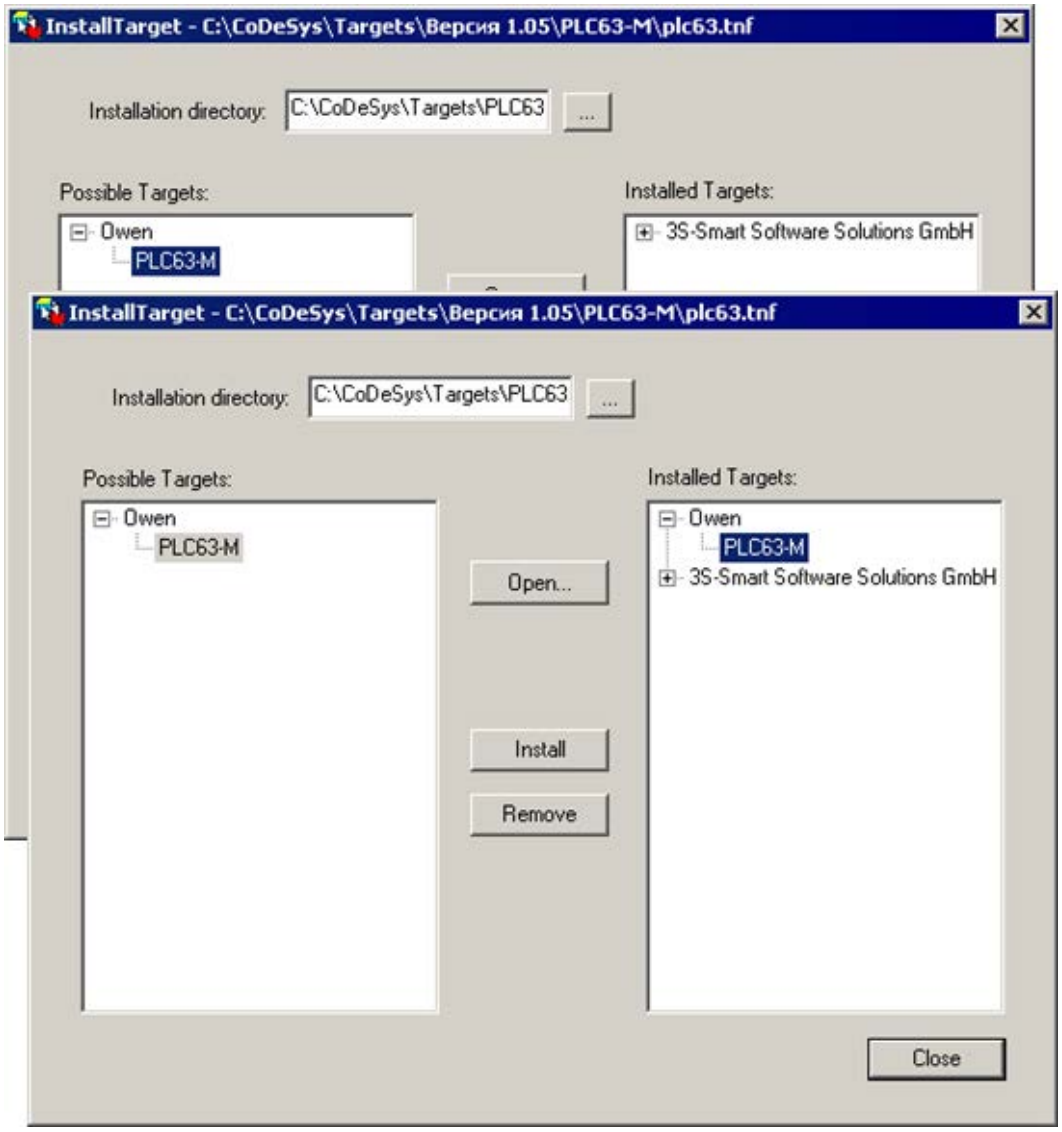


Рисунок 3.1 – Окно «InstallTarget» утилиты InstallTarget

3.2.2 Способ 2

Установка настроек целевой платформы (target-файла) производится при помощи специализированной утилиты «InstallTarget» от производителя ПЛК. Утилита представляет собой файл InstallTarget.bat, который, вместе с исполняемым файлом InstallTarget.exe, размещен в каждой папке каталога «Target-файлы» дистрибутивного диска.

Порядок инсталляции Target-файлов таков:

1) Выбрать в каталоге «Target-файлы» дистрибутивного диска папку с требуемым (соответствующим модификации ПЛК) Target-файлом;
2) Запустить на выполнение файл InstallTarget.bat.

На этом инсталляция Target-файла завершается. Никаких дополнительных операций выполнять не требуется.

После инсталляции Target-файлов следует перейти к собственно программированию ПЛК.

3.3 Запуск ПО CoDeSys. Главное окно программы

Запуск ПО осуществляется любым из способов, доступных в ОС MS Windows. Например, вызовом команды **Пуск | Программы | 3S Software | Codesys 2.3 | Codesys 2.3**.

После запуска ПО CoDeSys открывается Главное окно программы (см. рисунок 3.2).

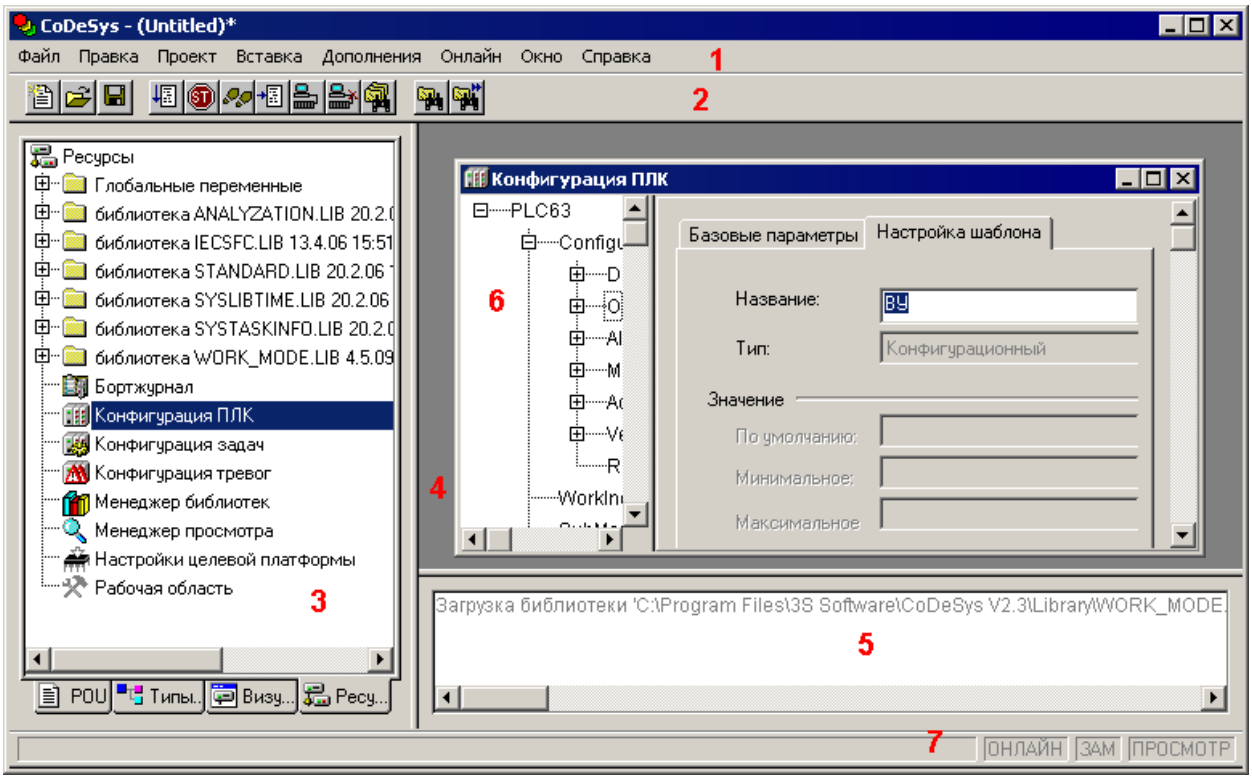


Рисунок 3.2 – Главное окно ПО CoDeSys

Главное окно программы содержит следующие элементы (помечены красными цифрами на рисунке 3.2):

- Главное меню (1), содержащее перечень доступных групп команд программы. В различных режимах работы группы команд главного меню дополняются специализированными командами.
- Панель инструментов (2), содержащая кнопки, дублирующие часто используемые команды программы. В различных режимах работы панель инструментов дополняется специализированными панелями.
- «Организатор объектов» (3) – переключатель групп режимов работы программы; включает четыре вкладки: «POU», «Типы данных (Data types)», «Визуализации (Visualizations)» и «Ресурсы (Resources)».
- Рабочая область программы (4), в которой отображаются окна (6) режимов работы программы (в правой верхней части).
- Окно (область) сообщений (5, в правой нижней части). В этом окне появляются сообщения компилятора, результаты поиска и список перекрестных ссылок.

– Строка статуса (7), содержащая информацию о текущем состоянии проекта (в нижней части). При выборе пункта меню в строке статуса отображается его описание; при работе в текстовом редакторе – указывается позиция, в которой находится курсор (например, **Line:5, Col.11**); в режиме визуализации – отображаются координаты курсора X и Y, которые отсчитываются относительно верхнего левого угла окна; если указатель мыши находится на элементе, или над элементом производятся какие-либо действия, то отображается номер этого элемента; при вставке элемента – отображается его название (например, Rectangle).
  При работе в режиме «Online» надпись **Online** в строке статуса выделяется черным цветом, в ином случае надпись серая.
  В режиме «Online» можно определить, в каком состоянии находится программа: **SIM** – в режиме эмуляции, **RUN** – программа запущена, **BP** – установлена точка останова, **FORCE** – происходит фиксация переменных.

Области Главного окна разделены линиями – разделителями, которые могут перемещаться с помощью мыши. Это позволяет подобрать оптимальное сочетание размеров областей.

Управление работой программы осуществляется выбором требуемых команд главного меню, которое отображается в верхней части главного окна программы и содержит пункты «Файл (File)», «Правка (Edit)», «Проект (Project)», «Вставка (Insert)», «Дополнения (Extras)», «Онлайн (Online)», «Окно (Window)» и «Справка (Help)».

Вызов команд главного меню дублируется кнопками панели инструментов, командами контекстных меню и горячими клавишами.

Перечень элементов управления программой с описанием запускаемых ими операций приведен в таблице А.1.

3.3.1 «Организатор объектов»

Организатор объектов ПО CoDeSys расположен в левой части главного окна программы (см. рисунок 2.2) и предназначен для вызова режимов работы ПО.

Организатор объектов включает четыре вкладки: «POU», «Типы данных (Data types)», «Визуализации (Visualizations)» и «Ресурсы (Resources)».

В пределах вкладок отображаются иерархические списки соответствующих объектов программы (например, режимов работы ПО или объектов в пределах одного и того же режима).

Для перехода на требуемую вкладку следует щелкнуть левой кнопкой мыши на наименовании требуемой вкладки (в нижней части «Организатора объектов»).

Для перехода к требуемому объекту в пределах выбранной вкладки следует щелкнуть левой кнопкой мыши на наименовании требуемого объекта или перейти на требуемую строку с помощью клавиш со стрелками.

Для открытия окна режима (или окна одного из объектов режима) следует дважды щелкнуть левой кнопкой мыши на наименовании требуемого объекта или, выбрав наименование требуемого объекта, нажать клавишу <Enter>. Окно режима (или окно одного из объектов режима) откроется в рабочей области (в правой верхней части) главного окна.

На вкладке «**POU**» отображается иерархический список программных компонентов (POU) проекта: функциональных блоков, функций и программ. Отдельные POU могут включать действия (подпрограммы). Каждый программный компонент состоит из раздела объявлений и кода. Для написания всего кода POU используется только один из МЭК языков программирования (IL, ST, FBD, SFC, LD или CFC). **CoDeSys** поддерживает все описанные стандартом МЭК компоненты. Для их использования достаточно включить в свой проект библиотеку standard.lib.

На вкладке «**Типы данных (Data types)**» отображается иерархический список типов данных, используемых в проекте. Кроме стандартных типов данных, в проекте могут быть использованы определяемые пользователем типы данных: структуры, перечисления и ссылки.

На вкладке «**Визуализации (Visualizations)**» отображается иерархический список элементов визуализации проекта – **графических представлений** объекта управления. Визуализация непосредственно связана с созданной в CoDeSys программой контроллера. **Редактор визуализации** CoDeSys предоставляет набор готовых графических элементов, которые могут быть связаны соответствующим образом с переменными проекта.

Примечание – Визуализация отображается на мониторе ПК, и не связана с информацией, отображаемой на ЖКИ ПЛК.

В Online режиме представление элементов на экране изменяется в зависимости от значений переменных.

Визуализация может исполняться в системе программирования, в отдельном приложении CoDeSys HMI.

На вкладке «**Ресурсы (Resources)**» отображается иерархический список ресурсов – объектов CoDeSys, обеспечивающих конфигурацию проекта, включая:

- Глобальные переменные, используемые во всем проекте.
- Менеджер библиотек для подключения необходимых библиотек к проекту.
- Журнал записи действий во время исполнения.
- Конфигурация ПЛК для конфигурирования области ввода / вывода контроллера.
- Настройки целевой платформы.
- Конфигурация задач для управления задачами.
- Менеджер просмотра для просмотра и заказа наборов значений переменных.

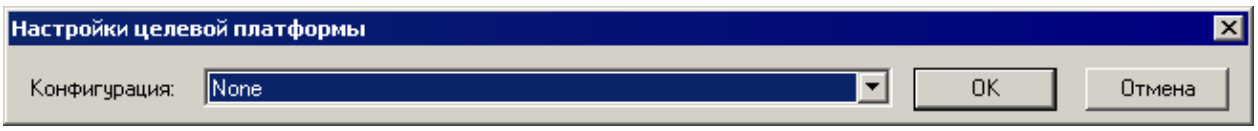
Двойной щелчок левой кнопкой мыши на требуемой записи в списке «Ресурсы» приводит к открытию в рабочей области главного окна выбранного режима («ресурса»).

3.3.2 Проект. Выбор контроллера и языка программирования

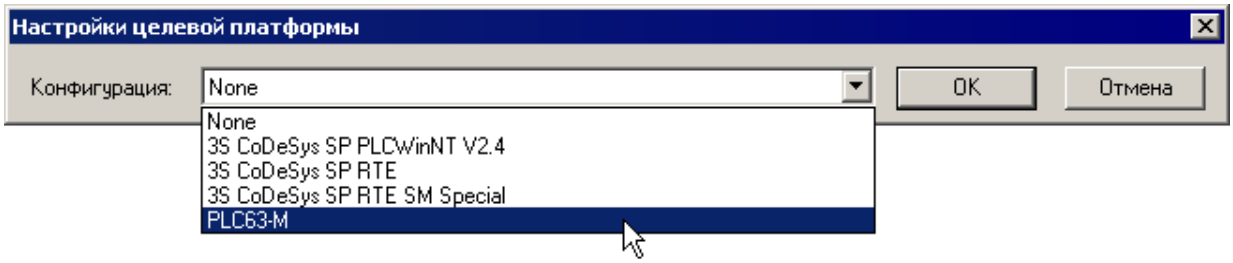
Для создания нового проекта (пользовательской программы ПЛК) следует:

1) Выбрав команду **Пуск | Программы | 3S Software | CoDeSys V.2.3 | CoDeSys V.2.3**, запустить ПО CoDeSys.
2) В открывшемся главном окне ПО CoDeSys (рисунок 3.2) – вызвать команду «Файл | Новый (File | New)**»** главного меню или нажать кнопку «Новый (New)» ( ) панели инструментов.
3) В открывшемся окне «Настройки целевой платформы (Target Settings)» (рисунок 3.3, а) – нажатием на кнопку у правого края поля «Конфигурация (Configuration)» раскрыть список предварительно установленных на ПК Target-файлов (см. п. 3.2). В списке – выделить требуемый файл (рисунок 3.3, б) и щелкнуть на его названии левой кнопкой мыши.

а)



б)



в)

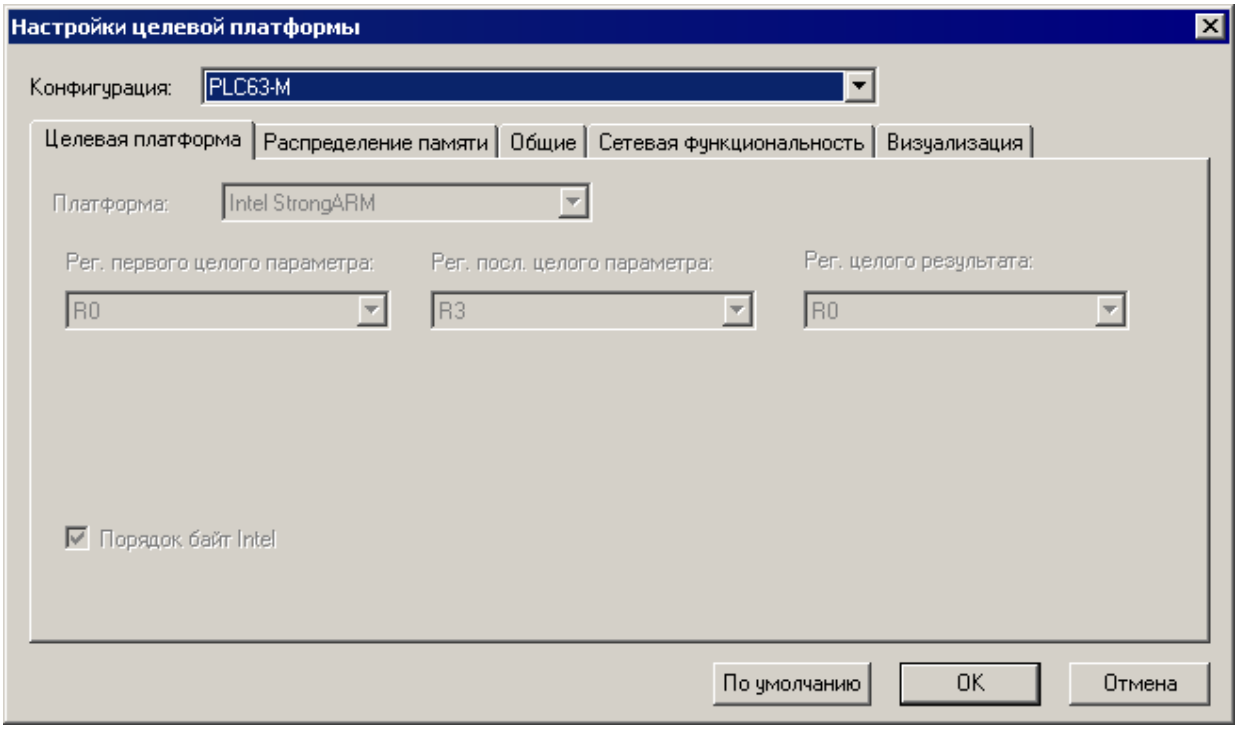


Рисунок 3.3 – Окно «Настройки целевой платформы (Target Setting)»

4) В открывшихся вкладках окна «Настройки целевой платформы (Target Setting)» отображаются установленные производителем значения параметров целевой платформы (рисунок 3.3, в).
5) Нажать кнопку «ОК» окна «Настройки целевой платформы (Target Setting)».

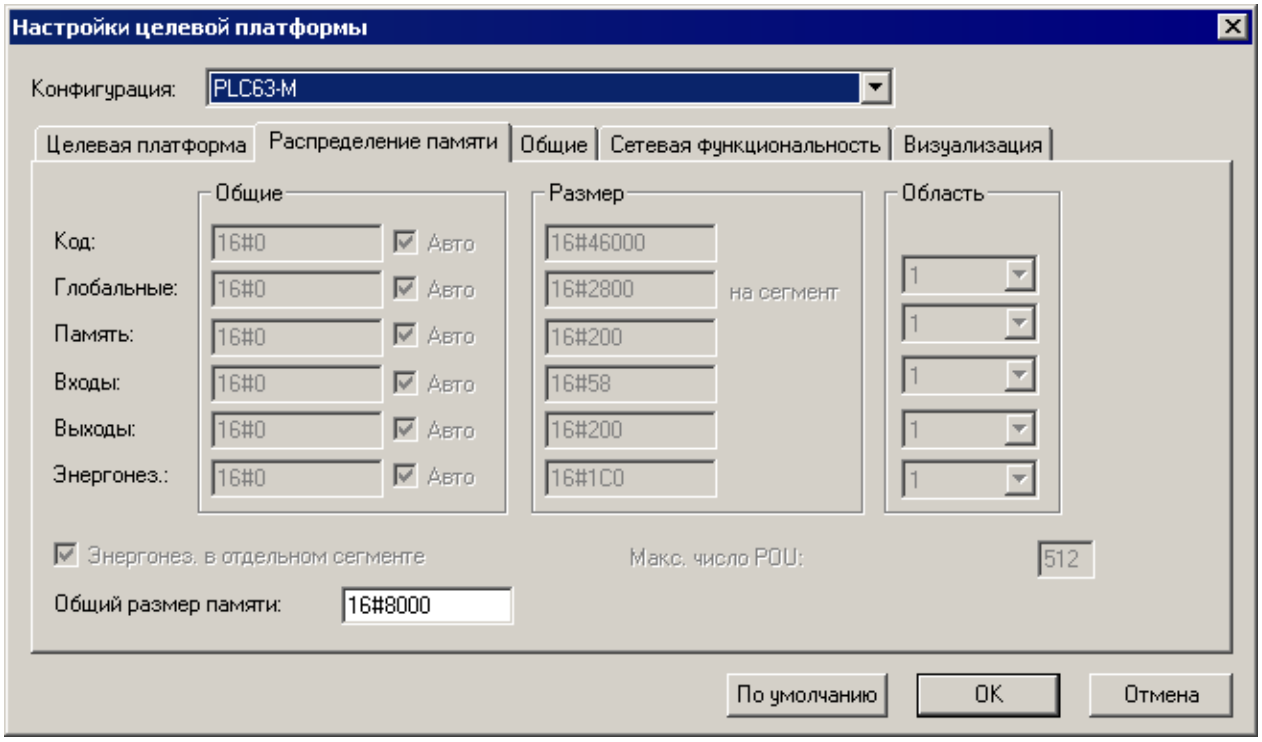


Рисунок 3.4 – Окно « Настройки целевой платформы (Target Setting)», вкладка «Распределение памяти (Memory Layout)»

6) В открывшемся окне «Новый программный компонент (New POU)» (см. рисунок 3.5), в поле «Имя нового POU (Name of new POU)» – отображается заданное по умолчанию имя новой главной программы проекта (**PLC_PRG**); его не следует изменять. В группе переключателей «Тип POU (Type of POU)» отображается заданный по умолчанию тип новой главной программы проекта (**Программа (Program)**); его также не следует изменять.

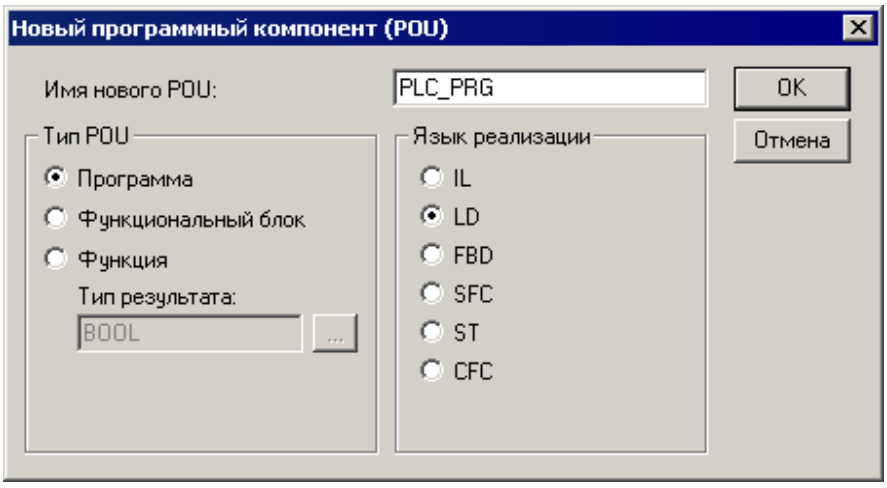


Рисунок 3.5 – Окно «Новый программный компонент (New POU)»

7) В группе переключателей «Язык реализации (Language of the POU)» следует выбрать требуемый язык программирования (о языках программирования – см. раздел 3.3.2.1).
В правой верхней области главного окна программы откроется окно редактора, в котором создается программа, исполняемая контролле-

ром. В зависимости от выбранного языка программирования это окно выглядит по-разному (на рисунке 3.6, а – пример для языка LD (Ladder Diagram – Язык релейных диаграмм).

В верхней части этого окна отображается область объявления переменных – «Редактор объявлений», в нижней – область редактора собственно программы.

Одновременно главное меню программы (команда «Вставить (Insert)») и контекстное меню области редактирования программы (см. рисунок 3.6, б) дополняются командами, специфичными для выбранного языка. Кроме того, панель инструментов дополняется локальной панелью, содержащей кнопки, соответствующие этим командам.

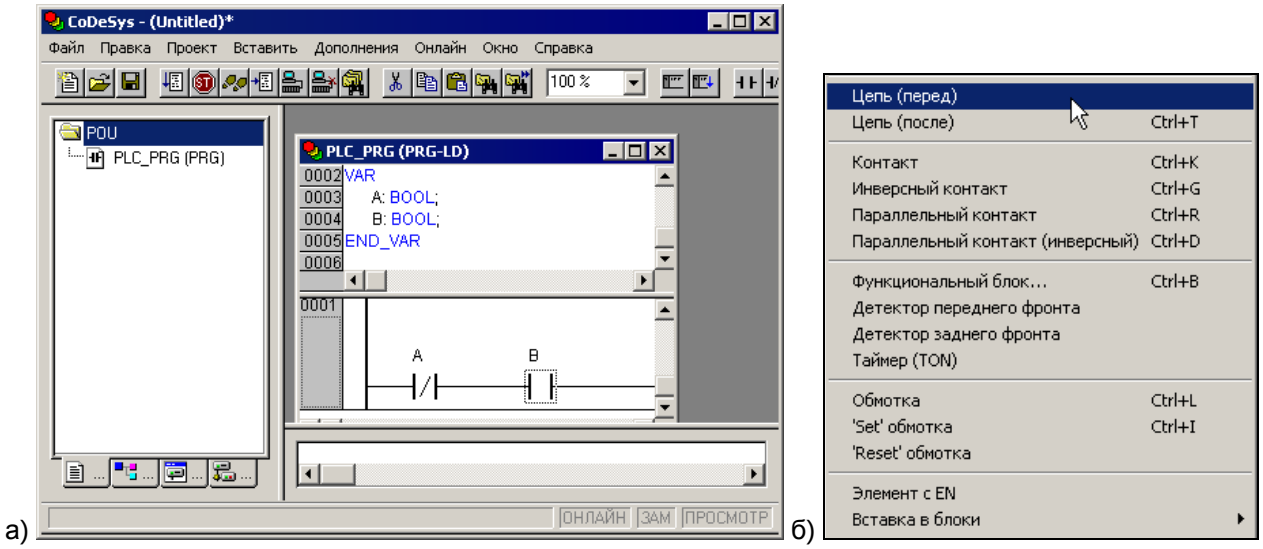


Рисунок 3.6 – Окно создания проекта (а) PLC_PRG (PRG-LD) и контекстное меню области редактирования программы (б) на языке LD (Ladder Diagram – Язык релейных диаграмм)

3.3.2.1 Языки программирования

В соответствии с требованиями стандарта МЭК 61131, ПО CoDeSys поддерживает языки программирования, перечисленные ниже.

Кроме того, ПО CoDeSys поддерживает «Язык непрерывных функциональных схем» (**CFC**), схожий с FBD, но, в отличие от последнего, блоки и соединители в этом языке располагаются свободно, разрешаются циклы и свободные соединения

Каждый из перечисленных языков обладает специфическими чертами, определяющими их применение для решения определенных задач.

Подробное описание языков программирования приведено в документе документа «Руководство пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [2]).

Краткие описания языков программирования приведены ниже.

3.3.2.1.1 Язык «IL» – список инструкций

Текстовый язык, схожий с ассемблером STEP5 фирмы SIEMENS; все операции производятся через аккумулятор; легко читается в случае небольших программ.

Каждая инструкция начинается с новой строки и содержит оператор и, в зависимости от типа операции, один и более операндов, разделенных запятыми.

Перед операндом может находиться метка, заканчивающаяся двоеточием (:). Комментарий должен быть последним элементом в строке. Между инструкциями могут находиться пустые строки. Пример IL программы приведен на рисунке 3.7

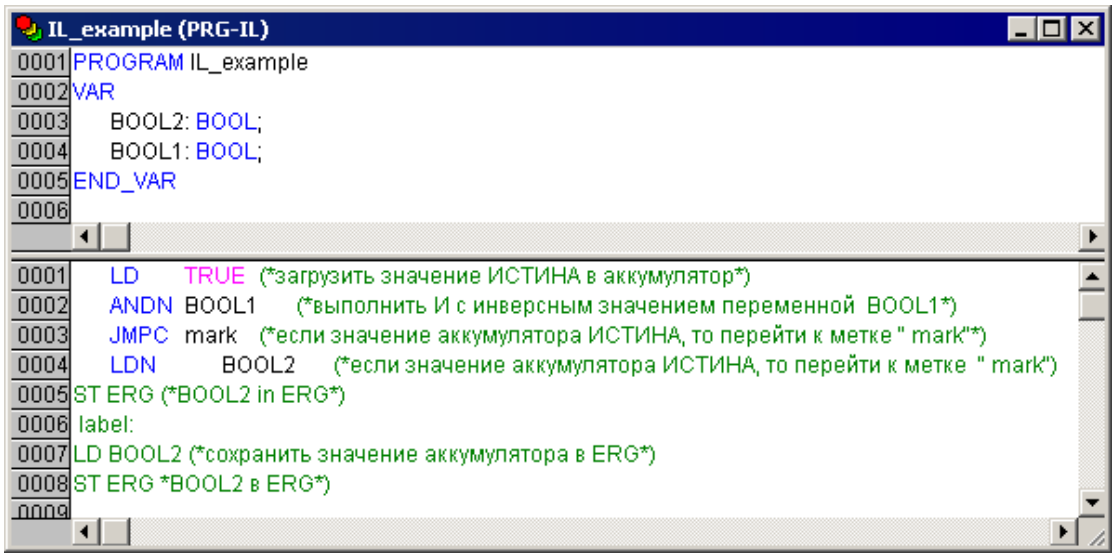


Рисунок 3.7 – Пример программы на языке IL

3.3.2.1.2 Язык «ST» – структурированный текст

Текстовый язык высокого уровня, схожий с языком «Паскаль»; оптимален для программирования циклов и условий. Представляет собой набор инструкций, которые могут использоваться в условных операторах (IF..THEN..ELSE) и в циклах (WHILE..DO).

Пример ST программы приведен на рисунке 3.8.

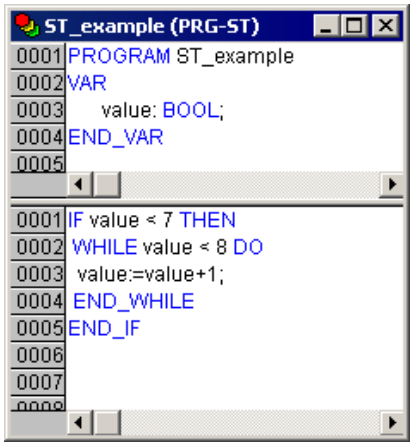


Рисунок 3.8 – Пример программы на языке ST

3.3.2.1.3 Язык «FBD» – функциональные блоковые диаграммы

Графический язык программирования. Работает со схемами, состоящими из блоков и операндов – с последовательностью цепей, каждая из которых содержит логическое или арифметическое выражение, вызов функционального блока, переход или инструкцию возврата.

Пример FBD программы приведен на рисунке 3.9.

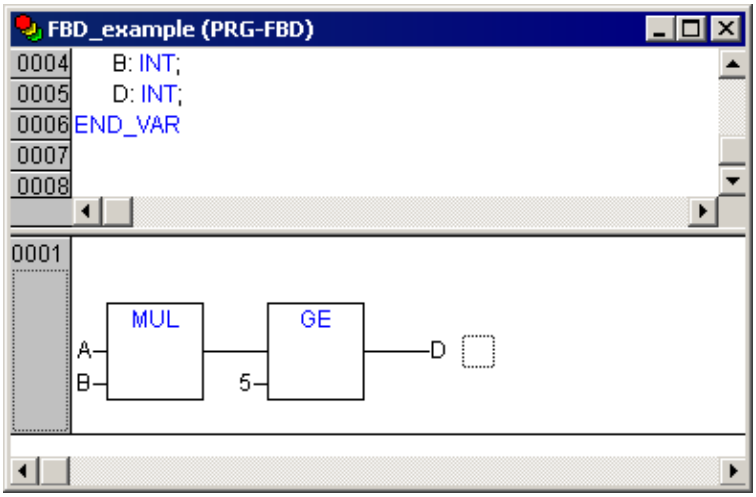


Рисунок 3.9 – Пример программы на языке FBD

3.3.2.1.4 Язык «LD» – релейные диаграммы

Графический язык, реализующий структуры электрических цепей; программа на языке LD состоит из схем с последовательностью цепей, каждая из которых содержит логическое или арифметическое выражение, вызов функционального блока, переход или инструкцию возврата. Сложен в использовании для работы с аналоговыми типами данных.

Лучше всего LD подходит для построения логических переключателей, но достаточно легко можно создавать на нем и сложные цепи – как в FBD. Кроме того, LD достаточно удобен для управления другими компонентами POU.

Используется для программирования большинства ПЛК. Допустимо переключение между языками FBD и LD.

Диаграмма LD состоит из ряда цепей. Слева и справа схема ограничена вертикальными линиями – шинами питания. Между ними расположены цепи, образованные контактами и обмотками реле, по аналогии с обычными электронными цепями. Слева любая цепь начинается набором контактов, которые посылают слева направо состояние «ON» или «OFF», соответствующие логическим значениям «ИСТИНА» или «ЛОЖЬ». Каждому контакту соответствует логическая переменная. Если переменная имеет значение «ИСТИНА», то состояние передается через контакт, если «ЛОЖЬ», то правое соединение получает значение «Выключено (OFF)».

Пример программы на языке LD приведен на рисунке 3.11.

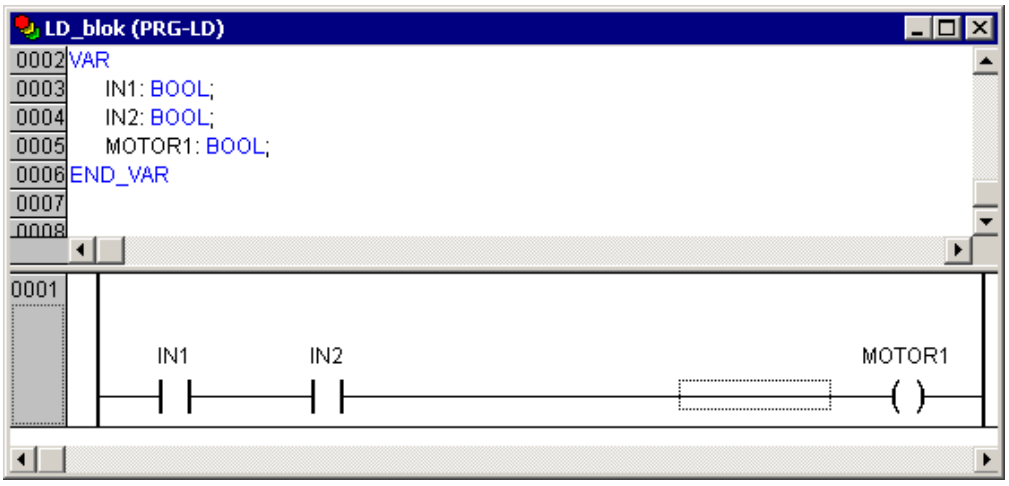


Рисунок 3.11 – Пример программы на языке LD

3.3.2.1.5 Язык «SFC» – последовательные функциональные схемы

Графический язык, используемый для структурирования приложений; состоит из шагов и переходов; действия выполняются внутри шагов. Не конвертируется в другие языки.

Пример программы на языке SFC приведен на рисунке 3.12.

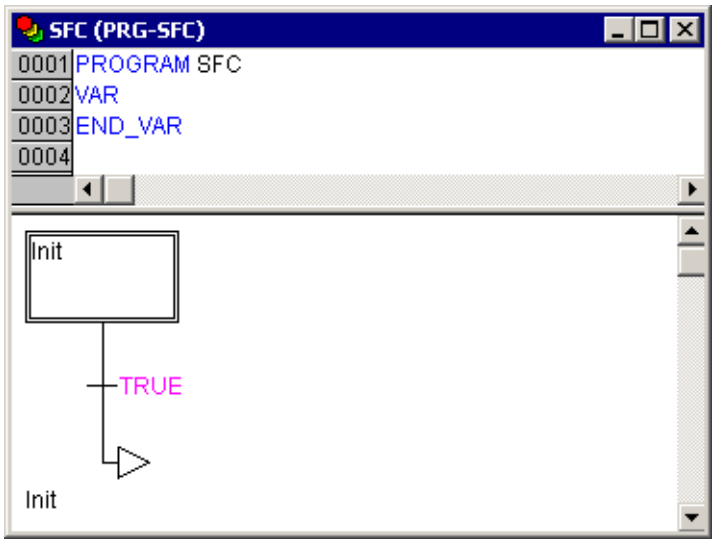


Рисунок 3.12 – Пример программы на языке SFC

3.3.2.1.6 Язык «CFC» – непрерывные функциональные схемы

Язык непрерывных функциональных схем. В отличие от FBD, не использует цепи, но дает возможность свободно размещать компоненты и соединения, что позволяет создавать, в частности, обратные связи.

Пример CFC программы приведен на рисунке 3.10.

Примечание. Свобода размещения компонентов и соединений определяет необходимость упорядочения порядка выполнения программы. Группа команд «Порядок | Показать порядок / Упорядочить топологически / В соответствии с потоком данных / Порядок: Выше, Ниже, В начало, В конец» контекстного меню позволяет отобразить порядковые номера (по очередности выполнения) элементов программы и изменить этот порядок при необходимости. Порядковые номера элементов отображаются в затемнённом квадратике у правого верхнего угла каждого элемента (см. рисунок 3.13).

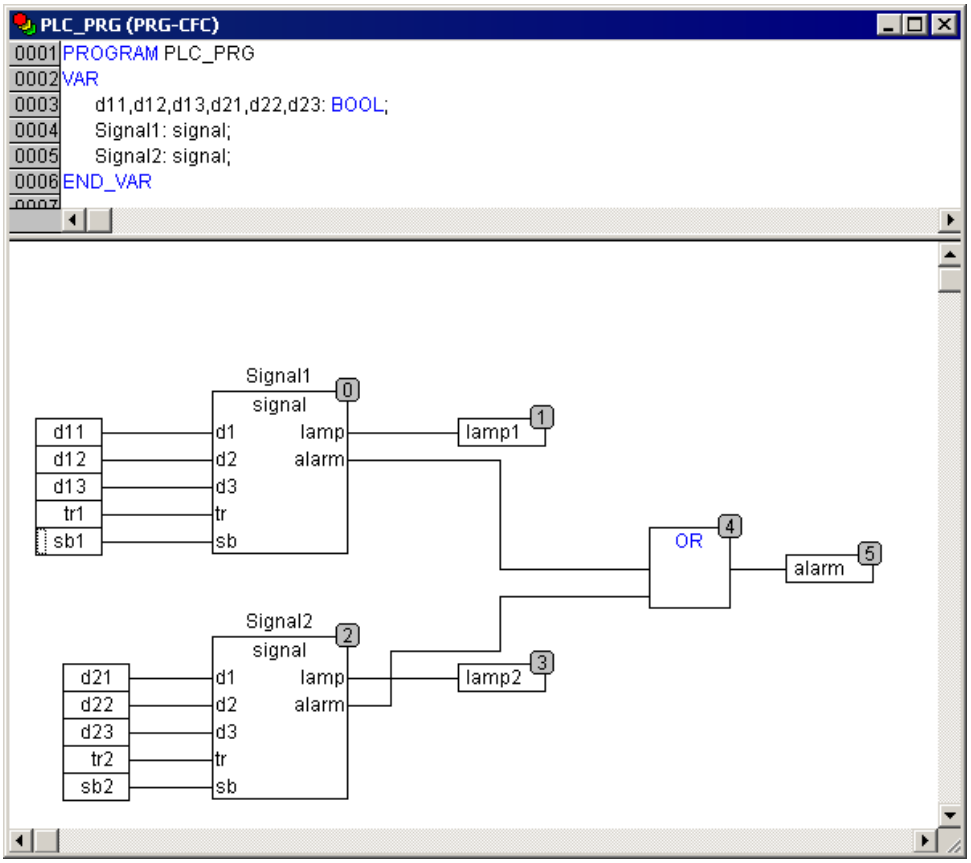


Рисунок 3.13 – Пример программы на языке CFC

3.3.2.2 Редакторы ПО CoDeSys

Все режимы редактирования (далее – «редакторы») программных компонентов POU (Program Organization Units) содержат область кода (нижняя часть окна) и раздел объявлений (верхняя часть окна), см. рисунки 3.7 – 3.13.

Область кода может включать графический или текстовый редактор; раздел объявлений – это всегда текст. Разделы кода и объявлений разделены горизонтальной границей, которую можно перетаскивать мышкой.

Окно редактирования открывается при входе в режим написания программного компонента. Для входа следует перейти на вкладку «POU» организатора объектов и выбрать в дереве программных компонентов проекта, отображаемом на вкладке, требуемый элемент. В рабочей области главного окна ПО CoDeSys откроется окно редактора. Тип окна зависит от выбранного языка программирования (см. п. 3.3.2.1). В каждом окне редактора становятся доступны команды контекстного меню, содержащие основные операции, доступные в выбранном языке. Одновременно панель инструментов главного окна ПО CoDeSys дополняется панелью, кнопки которой аналогично командам контекстного меню вызывают основные операции, доступные в выбранном языке. Окно редактора открывается также при добавлении программного компонента.

Пример главного окна ПО CoDeSys с открытым окном редактирования программного компонента на языке FBD (с дополнительной панелью инструментов и открытым контекстным меню) приведен на рисунке 3.14.

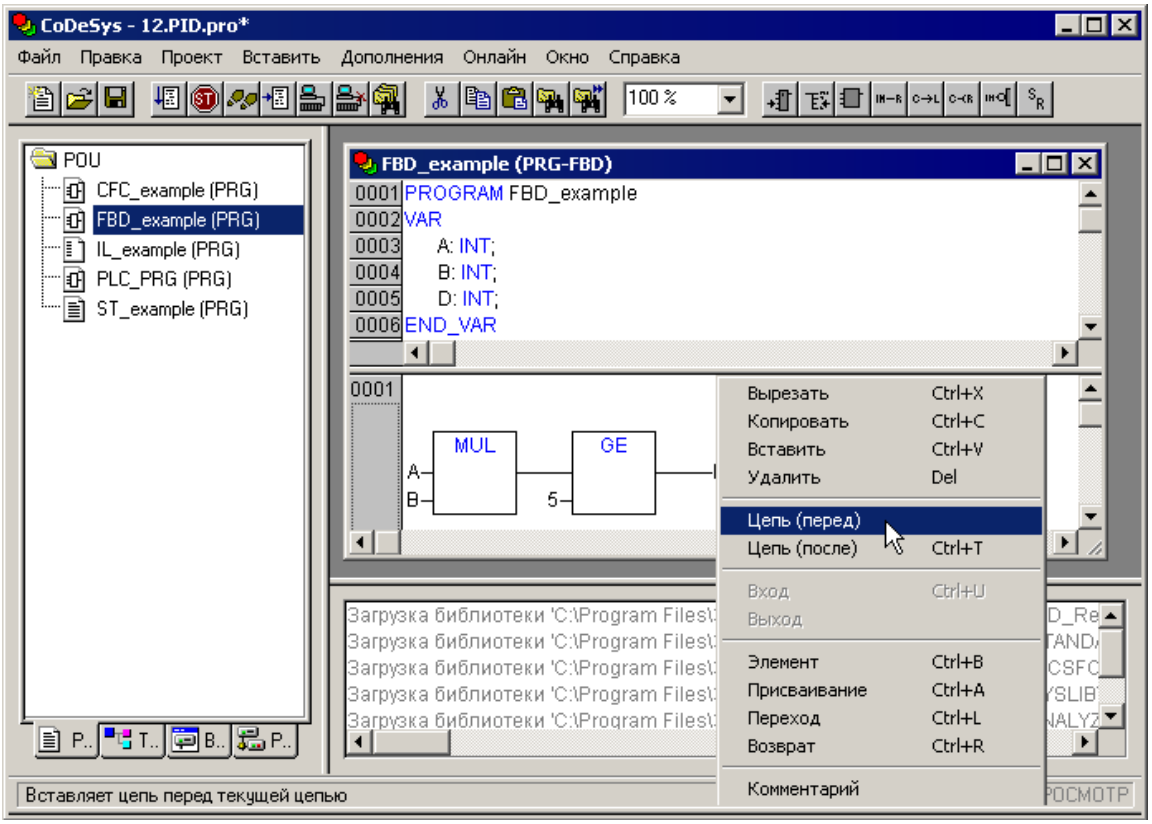


Рисунок 3.14 – Пример окна редактирования FBD программы

3.3.3 Проект. Программные компоненты (POU)

Проект создается в ПО CoDeSys на любом из доступных языков программирования. Проект может состоять из одного или нескольких программных компонентов (POU, Program Organization Unit). Главная программа, выполняемая циклически, должна называться PLC_PRG.

К программным компонентам (POU) относятся функциональные блоки, функции и программы. Отдельные POU могут включать действия (подпрограммы).

Каждый программный компонент состоит из раздела объявлений и кода. Для написания всего кода POU используется только один из МЭК языков программирования (IL, ST, FBD, SFC, LD или CFC).

CoDeSys поддерживает все описанные стандартом МЭК компоненты. Для их использования достаточно включить в свой проект библиотеку standard.lib (подробнее о библиотеках см. раздел 4.2.4).

POU могут вызывать другие POU, но рекурсии недопустимы.

Внимание! Нельзя удалять или переименовывать POU PLC_PRG: PLC_PRG является главной программой в однозадачном проекте.

Приемы работы при написании программ и примеры программ представлены в разделе 3.3 и в документе «Первые шаги в CoDeSys» (см приложение Н [1]).

3.3.4 Проект. Типы данных

Тип данных определяет род информации и методы ее обработки и хранения, количество выделяемой памяти. Программист может непосредственно использовать элементарные (базовые) типы данных (логический, целочисленные, строковые, вре-

менные; подробнее см. раздел 4.2.3) или создавать собственные (пользовательские) типы на их основе.

При работе с ПЛК63 и ПЛК73 используются пользовательские типы данных, используемые при программировании работы ЖКИ (см. раздел 8.4.2).

В разделе 4.2.3.1 описаны элементарные (базовые) типы данных.

В разделе 4.2.3.2 описаны пользовательские типы данных.

3.3.5 Проект. Установка связи с ПЛК

Перед установкой связи ПО CoDeSys с контроллером следует однократно настроить канал связи (интерфейс и настройки обмена), по которому будет осуществляться связь. В дальнейшем, при отладке программы, настройка интерфейса связи может потребоваться только при переходе на связь по другому интерфейсу.

Установка связи реализуется через кабель КС1, подключённый к последовательному COM-порту ПЛК. В ПЛК63 это – универсальный порт контроллера, который может быть использован как для связи ПЛК с CoDeSys (кабель КС1), так и для связи с ПК, другими контроллерами и т.д., по протоколам ОВЕН и Modbus; в ПЛК73 это отдельный порт для связи только с CoDeSys.

При этом CoDeSys можно запускать как на том компьютере, к которому подключён ПЛК, так и на удалённом (с использованием CoDeSys GateWay).

3.3.5.1 Настройка интерфейса связи

Для настройки интерфейса соединения ПК с контроллером следует:

1) Выбрать команду «Онлайн | Параметры связи (Online|Communication parameters)» главного меню ПО CoDeSys. Откроется окно «Communication parameters», см. рисунок 3.15, а.
2) Нажать кнопку «New» окна «Communication parameters». Откроется окно «Communication parameters: New Channel» (см. рисунок 3.15, б). В этом окне задается имя нового соединения (например, Owen) и выбирается из перечня интерфейс соединения.

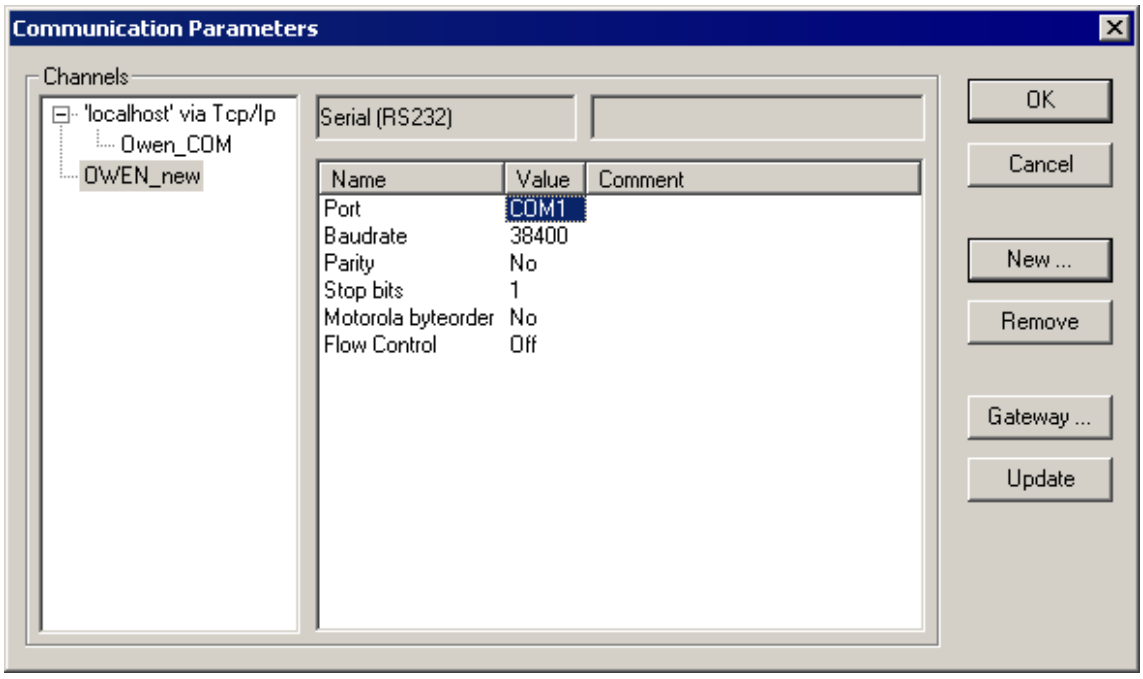


Рисунок 3.15 – Настройка интерфейса для соединения с ПЛК. Окно «Communication parameters»

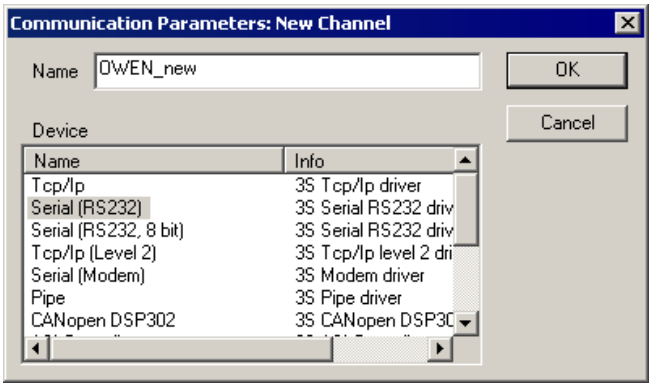
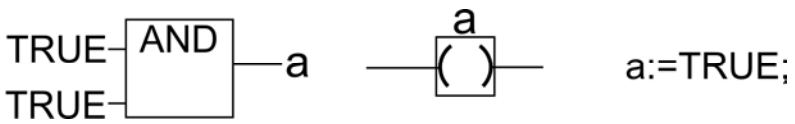


Рисунок 3.16 – Настройка интерфейса для соединения с ПЛК. Окно «Communication parameters: New Channel»

3.3.5.2 Установка связи с контроллером

Для установки связи необходимо, чтобы предварительно была создана программа пользователя, хотя бы простейшая.

Примеры программ на языках FBD, LD и ST, которые можно использовать для проверки связи с контроллером, приведены на рисунке 3.17. Простейшей программой на языке ST является символ «;» (двоеточие). Такой программы достаточно для проверки связи с контроллером.



а) б) в)

Рисунок 3.17 – Примеры программ на языках FBD (а), LD (б) и ST (в)

Для загрузки программы в контроллер следует:

1) Выбрать команду «Онлайн | Подключение (Online | Login)» главного меню, тем самым – установить связь с ПЛК. При этом должен быть снят флаг перед строкой меню «Онлайн | Режим эмуляции (Online | Simulation Mode)» (установка и снятие флага производится последовательными щелчками левой кнопкой мыши на строке).
   Перед установкой связи ПО скомпилирует проект; и в случае наличия в нем ошибок – прервет установку связи.

Сразу после установки связи среда программирования предложит загрузить (см. рисунок 3.18) или обновить код пользовательской программы в оперативной памяти контроллера.

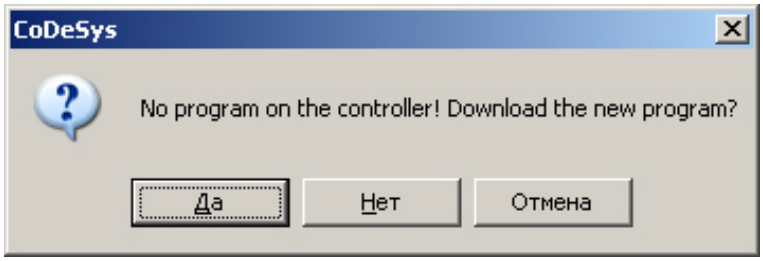


Рисунок 3.18 – Окно предложения загрузки программы

3.4 Конфигурирование области ввода-вывода ПЛК

Перед созданием программы необходимо настроить конфигурацию жидкокристаллического индикатора, входов, выходов и интерфейсов связи ПЛК с внешними устройствами (модулями ввода-вывода, устройствами индикации и т.д.), обмен данными с которыми будет производиться по сети. Перечисленные устройства обмениваются данными с пользовательской программой через специальную область памяти: Память ввода-вывода. Конфигурация ее задается в окне режима («Ресурса») «Конфигурация ПЛК (PLC Configuration)» ПО CoDeSys и подробно описана в разделе 7 настоящего документа.

Размер памяти ввода-вывода определяется типом лицензии CoDeSys контроллера ОВЕН ПЛК (см. раздел 2.2).

Подробное описание процедур конфигурирования области ввода-вывода приведено в разделе 4 .

Конфигурирование ЖКИ описано в разделе 4.3 .

3.4.1 Расчет потребности ПЛК в памяти ввода/вывода

Наиболее простым способом, позволяющим точно проверить, хватит ли доступного объема памяти ввода/вывода для выполнения проекта, является способ создания проекта. Не приобретая контроллер, но установив на компьютере ПО CoDeSys и Target-файл, можно создать проект, в котором подключить все необходимые модули. Подключение производится в режиме («Ресурсе») «Конфигурация ПЛК (PLC Configuration)», см. раздел 7 . При компиляции проекта либо компиляция пройдет успешно, либо CoDeSys сообщит об ошибке, если памяти недостаточно. Для осуществления подобной проверки собственно программу контроллера писать не требуется.

Альтернативным, более сложным, способом потребности ПЛК в памяти ввода/вывода рассчитываются по следующей схеме:

1) Для подсчета потребности ПЛК в памяти ввода / вывода, необходимой для работы с приборами ОВЕН, следует воспользоваться данными, приведенными в таблице 3.1.

Таблица 3.1 – Потребности ПЛК в памяти ввода/вывода, требуемой для работы с некоторыми приборами производства компании ОВЕН

Прибор	Протокол	
	ОВЕН	Modbus
	%Q (байт)	%Q (байт)
МВА8. Один аналоговый вход	8	2… 8*
МВУ8. Один аналоговый выход	2	2
МДВВ. Один дискретный выход в режиме ШИМ	4	2
МДВВ. Один счетчик дискретного входа	2	2
МДВВ. Битовая маска всех входов	2	2
МДВВ. Битовая маска всех выходов	-	2
ТРМ2хх. Один аналоговый вход	4	-
ТРМ151, ТРМ148, ТРМ133. Один аналоговый вход	8	-
ИП320. Одна переменная на чтение с ПЛК	-	2
ИП320. Одна переменная на запись в ПЛК	-	2

Примечания:
1) В области %q память используется в объеме, кратном 2-м байтам.
2) Область %i памяти в ПЛК63 не может быть занята.
*) – При работе с МВА8 можно считывать либо только измеренные значения (2 байта), либо дополнительно считывать время измерения и статус ошибки (до 8 байт информации).

2) При использовании приборов других производителей, работающих по протоколам Modbus, следует по руководствам на эти приборы определить, сколько байт данных содержат команды, посылаемые по сети. При работе с приборами ввода количество этих байт надо прибавить к размеру области %I, при работе с приборами вывода количество надо прибавить к размеру области %Q.
3) Для дискретных модулей ввода/вывода сторонних производителей, работающих по протоколу Modbus, как правило, значение одного входа или одного выхода кодируется одним битом. Соответственно, занимаемый размер памяти в области ввода/вывода следует считать в битах, но с учетом того, что на один модуль тратится целое число байт. Таким образом, на двенадцатиканальный модуль дискретного ввода потребуется два байта, из 16 бит которых только 12 будут значащими.
4) Для приборов и операторских панелей, работающих по протоколу Modbus, передача одного значения параметра осуществляется как минимум в двухбайтном регистре (даже если параметр – однобайтовый).
5) При использовании модулей Master сетевых протоколов (т.е. модулей, организующих обмен с внешними устройствами и модулями) дополнительно следует учесть, что эти модули содержат ряд служебных переменных, также расположенных в области памяти вывода %Q. Один модуль Master одного сетевого протокола дополнительно требует от 4 до 8 байт.
6) После подсчета необходимого размера областей памяти %I и %Q следует провести проверку достаточности объема доступной памяти каждого типа. При этом следует учитывать, что часть памяти занимается собственными входами и выходами. Сводные данные приведены в таблице 2.1.
8) Если расчет показал, что резерва памяти (в контроллере модификации L) нет, то следует приобретать контроллер модификации M.

3.5 Визуализация

ПО CoDeSys позволяет создать одно или несколько окон-визуализаций, в которых пользователь может располагать визуальные элементы, позволяющие графически отобразить данные из пользовательской программы. Данные в визуализацию передаются из ПЛК, при установленной с ним связи (подробнее см. п. 3.3.5).

В режиме «Online» представление элементов на экране изменяется в зависимости от значений переменных.

Например, если уровень заполнения емкости жидкостью доступен в программе в виде значения некоторой переменной, то в окне визуализации он может быть изображен графическим элементом в виде полосы, которая, в зависимости от значения переменной проекта, будет изменять свою длину и/или цвет. Рядом может быть размещен текст, отображающий в виде числа текущий результат измерения. Здесь же можно разместить и, например, кнопки запуска и остановки программы.

Создание окна визуализации выполняется на вкладке «Визуализации» Организатора объектов ПО CoDeSys (см. рисунок 3.19).

Внимание! Число одновременно отображаемых переменных в окнах визуализации (и отладки) зависит от их типа и от частоты их обновления. В среднем количество переменных для отладки и визуализации не должно превышать 30-50 шт.
Это ограничение не относится к общему количеству переменных, используемых в проекте.

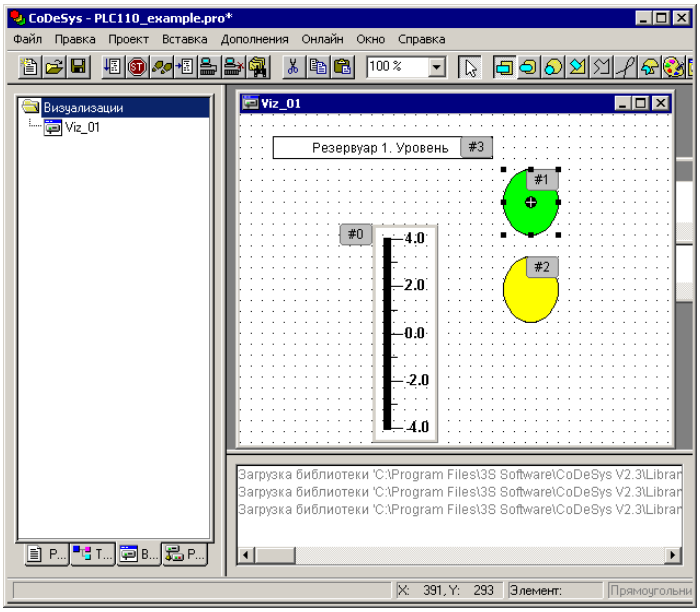


Рисунок 3.19 – Окно визуализации проекта

3.6 Сохранение проекта

Созданный проект следует сохранить в виде файла на жестком диске персонального компьютера для дальнейшей работы. Начальное сохранение проекта производится вызовом команды «Файл | Сохранить как (File | Save as)», последующие сохранения изменений – вызовом команды «Файл | Сохранить (File | Save)» или нажатием кнопки «Сохранить» ( ) панели инструментов.

Проект CoDeSys может быть сохранен совместно с конфигурацией, т.е. со структурой, описанной в Target-файле, загруженном при вызове проекта. Такой способ сохранения на несколько килобайт увеличивает сохраняемый файл проекта, но позволяет в дальнейшем не заботиться о совместимости проекта и версии Target-файла, установленного в системе на момент редактирования проекта. В таком режиме рекомендуется сохранять проект в случае, когда предполагается возможность редактирования проекта по прошествии значительного времени с момента его создания. Включение режима сохранения проекта совместно с конфигурацией производится в окне «Конфигурация ПЛК (PLC Configuration)», установкой флажка переключателя «Сохранять конф<игурационные> файлы в проекте (Save configuration file in project)», см. рисунок 3.20.

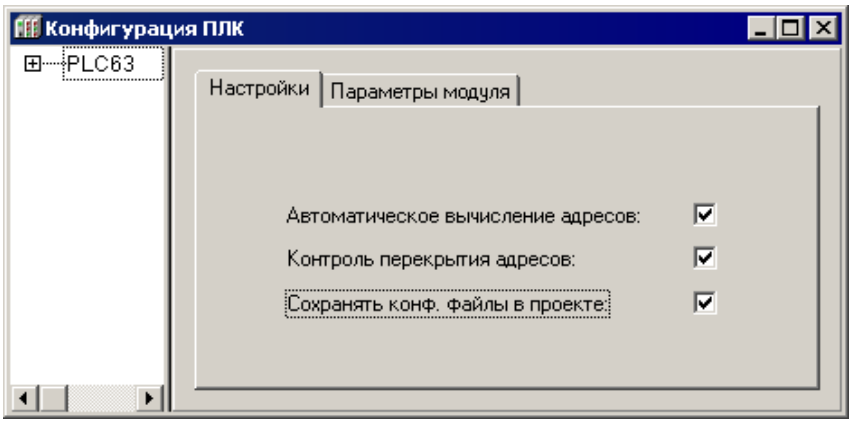


Рисунок 3.20 – Окно «Конфигурация ПЛК (PLC Configuration)». Установка опции сохранения конфигурации в файл проекта

3.7 Запуск пользовательской программы

Для запуска загруженной программы следует выбрать команду «Онлайн | Старт (Online | Run)» главного меню. Кроме того, она автоматически запускается при старте контроллера (то есть после перезагрузки контроллера).

После запуска программы следует проверить ее работоспособность, эмулировав или воссоздав необходимые сигналы на входах контроллера или на подключенных модулях ввода и удостовериться в правильности управления выходами или модулями вывода.

3.8 Сохранение программы в памяти контроллера

При загрузке программы в ПЛК она каждый раз автоматически загружается во flash память контроллера. Таким образом, выполнения команды «Online | Login» главного меню достаточно для записи программы в контроллер. Дополнительных действий не требуется.

	Внимание! Ресурс встроенной Flash-памяти контроллера ограничен (около 50 000 циклов перезаписи), поэтому не рекомендуется при отладке программы каждый раз записывать ее во Flash-память.

Если контроллер циклически перегружается из-за ошибок в программе, сохраненной во Flash-памяти, или некорректной записи программы во Flash-память, следует запустить контроллер, удерживая нажатой кнопку

на передней панели контроллера. Программа из Flash-памяти не будет автоматически запущена, что даст возможность подключиться к контроллеру через CoDeSys и загрузить в него корректно работающую программу.

4 Написание программы

В данном разделе приводится пример разработки программы (проекта).

Подготовительный этап создания пользовательской программы (проекта) – установка ОС и ПО CoDeSys описаны в разделе 2 .

Этапы создания пользовательской программы (проекта) описаны в разделе 3 данного руководства.

Установка настроек целевой платформы описана в разделе 3.2 .

Пользовательская программа («проект») ПЛК в ПО CoDeSys содержит программные компоненты (POU), типы данных, визуализации, ресурсы и библиотеки, сведения о ресурсах ПЛК и некоторую другую информацию, хранимую в одном файле («name.pro»).

4.1 Программные компоненты проекта

К программным компонентам (POU) относятся функциональные блоки, функции и программы. Отдельные POU могут включать действия (подпрограммы).

Каждый программный компонент состоит из раздела объявлений и кода. Для написания всего кода POU используется только один из МЭК языков программирования (IL, ST, FBD, SFC, LD или CFC).

CoDeSys поддерживает все описанные стандартом МЭК 61131 компоненты. Для их использования достаточно включить в свой проект библиотеку standard.lib (подробнее о библиотеках см. раздел).

POU могут вызывать другие POU, но рекурсии недопустимы.

Максимальное число POU в программах контроллеров ПЛК63 и ПЛК73 – 256.

Приемы работы при написании программ и примеры программ представлены в документе «Первые шаги в CoDeSys» (см приложение Н [1]).

4.1.1 Программы

Программа это программный компонент (POU), способный формировать произвольное число значений во время вычислений. Значения всех переменных программы сохраняются между вызовами. В отличие от функционального блока (см. ниже), экземпляров программы не существует. Программа является глобальной во всем проекте.

Нельзя вызывать программу из функции (см. ниже).

Если вызвать программу, которая изменит значения своих переменных, то при следующем вызове ее переменные будут иметь те же значения, даже если она вызвана из другого POU.

В этом заключается главное различие между программой и функциональным блоком, в котором изменяются только значения переменных данного экземпляра функционального блока.

Объявление программы начинается ключевым словом «PROGRAM».

Пример записи программы на языке IL приведен на рисунке 4.1

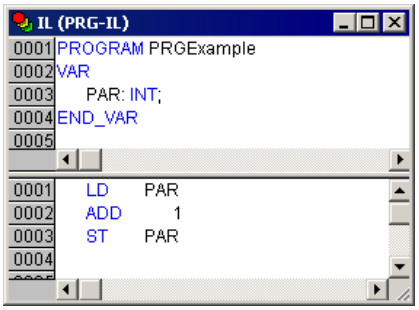


Рисунок 4.1 – Пример записи программы на языке IL

4.1.2 Функции

Функция – это программный компонент (POU), который возвращает только единственное значение (которое может состоять из нескольких элементов, если это битовое поле или структура). В текстовых языках функция вызывается как оператор и может входить в выражения.

При объявлении функции необходимо указать тип возвращаемого значения. Для этого после имени функции нужно написать двоеточие и тип (см. рекомендации по наименованию в приложении J документа «Руководство пользователя по программированию ПЛК в CoDeSys 2.3», см. приложение H, [2]). Правильно объявленная функция выглядит следующим образом: **FUNCTION Fct: INT**;

Имя функции используется как выходная переменная, которой присваивается результат вычислений.

Объявление функции должно начинаться с ключевого слова FUNCTION. Пример функции, написанной на языке IL, использующей три входных переменных (**par1** – **par3**) целочисленного типа (INT; диапазон изменения – от минус 32768 до 32767) и возвращающей результат деления произведения первых двух на третью.

Пример записи функции на языке IL приведен на рисунке 4.2.

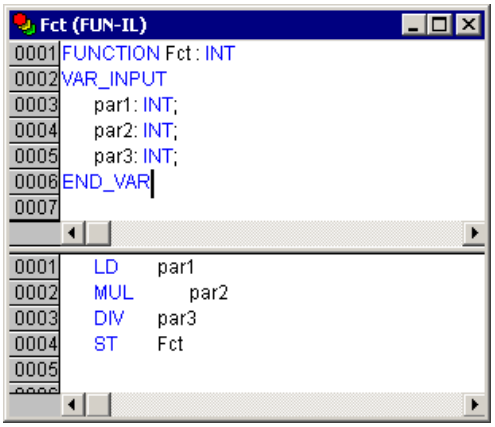


Рисунок 4.2 – Пример записи функции на языке IL

В языке ST вызов функции может присутствовать в выражениях как операнд.

В SFC функция вызывается только из шага или перехода.

Примечание. Функция не имеет внутренней памяти, но CoDeSys допускает использование в функциях глобальных переменных. Это является отклонением от требований стандарта МЭК 61131-3, в соответствии с которыми выходное значение функции должно зависеть исключительно от входных параметров. Т.е. функция с одними и теми же значениями входных параметров всегда должна возвращать одно и то же значение.

4.1.3 Функциональный блок

Функциональный блок – это программный компонент (POU), который принимает и возвращает произвольное число значений. В отличие от функции (см. ниже), функциональный блок не формирует возвращаемое значение.

Объявление функционального блока начинается с ключевого слова FUNCTION_BLOCK.

Функциональный блок может имеет один или несколько экземпляров (копий).

На рисунке 4.3 приведен пример функционального блока, написанного на IL, который имеет две входных и две выходных переменных. Значение выходной переменной MULERG равно произведению значений двух входных переменных, а значение VERGL определяется в результате сравнения значений входных переменных.

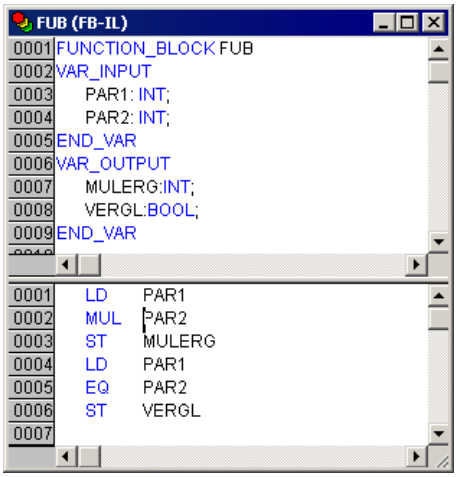


Рисунок 4.3 – Пример записи функционального блока на языке IL

4.2 Использование переменных

Программные компоненты (POU) проекта обрабатывают **переменные** – величины, значения которых могут меняться в ходе выполнения программы (в частных случаях переменные, обрабатываемые программой, могут быть и константами). Переменные могут использоваться для хранения и передачи промежуточных результатов выполнения логических операций, значений состояний входов или выходов функциональных блоков программы, значений состояний входов или выходов ПЛК и др. Каждая переменная имеет идентификатор.

4.2.1 Типы переменных

Переменные, используемые в ПО CoDeSys, могут принадлежать к нескольким типам.

Во-первых, переменные могут относитьмся к **локальным** или к **глобальным** переменным. Локальные переменные могут использоваться только в рамках текущего компонента POU, глобальные переменные – могут использоваться в рамках всего проекта (во всех программных компонентах, входящих в его состав). Локальные переменные задаются в редакторе объявлений (см. раздел 3.3.2), глобальные – в аналогичном редакторе, вызываемом выбором объекта «Глобальные переменные (Global Variables)» .

Кроме того, используемые переменные могут относиться к **входным** или **выходным**, а также к переменным, одновременно являющимся **входными и выходными**.

При необходимости сохранять значения переменных они могут быть объявлены как **ременантные переменные** – такие переменные сохраняют свои значения при определенных сбоях в системе. Они бывают сохраняемые и постоянные.

Сохраняемые переменные обозначаются при объявлении ключевым словом RETAIN. Эти переменные сохраняют свои значения, даже если произошла авария питания (выключение и включение) контроллера, что равносильно команде «Сброс» (Онлайн | Сброс (Online | Reset')). Значения RETAIN переменных сохраняются в энергонезависимой памяти контроллера.

Постоянные переменные обозначаются ключевым словом PERSISTENT. В отличие от сохраняемых переменных эти переменные сохраняют свои значения только при загрузке кода новой программы, но не при выключении питания или любом сбросе. Значения постоянных переменных размещаются вне энергонезависимого ОЗУ.

Подробнее о типах переменных см. Руководство пользователя ПО CoDeSys (см. приложение Н, [2]).

4.2.2 Объявление переменных

Для использования в POU переменная должна быть **объявлена**. Объявление переменных производится в «Редакторе объявлений» (см. раздел 3.3.2).

«Редактор объявлений» отображается в верхней части окна редактирования POU, открывающегося при выборе существующего или при добавлении нового программного компонента (в дереве программных компонентов на вкладке «POU» Организатора объектов).

Редактор объявлений используется для объявления переменных POU, глобальных переменных, описания типов данных.

В разделе объявлений зарезервированные слова, типы данных и сами переменные автоматически выделяются разными цветами (см. рисунок 4.4).

Наиболее важные команды можно найти в контекстном меню, которое появляется по щелчку правой кнопки мыши или по нажатию сочетания клавиш <Ctrl>+<F10>.

Локальные переменные POU объявляются в разделе объявлений редактора программного компонента. Такими переменными могут быть входные и выходные переменные, переменные, одновременно являющиеся входными и выходными, локальные переменные, сохраняемые переменные и константы.

Синтаксис, используемый при объявлении переменных, соответствует стандарту МЭК61131-3.

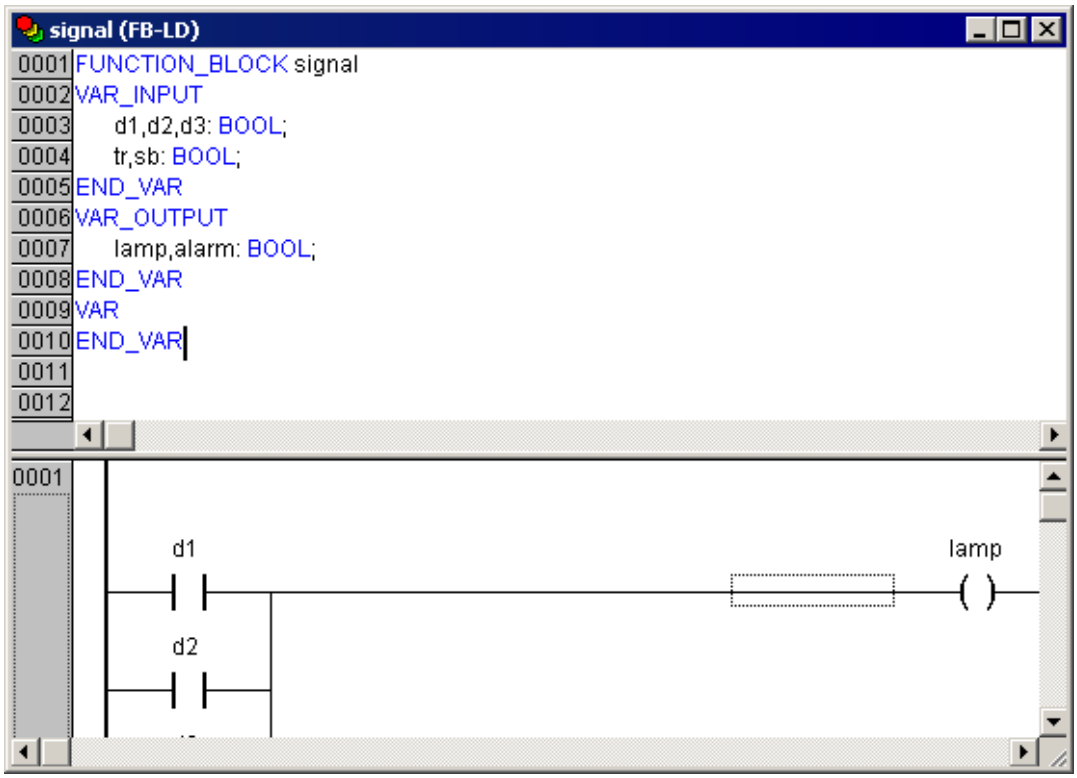


Рисунок 4.4 – Окно редактора объявлений (верхняя часть окна ркедактора POU)

Переменные объявляются следующим образом:

<Идентификатор> {AT <Адрес>}:<Тип> {:=<начальное значение>};

Части, заключенные в фигурных скобках, не обязательны.

Имена переменных не должны содержать пробелов и специальных символов, должны объявляться только один раз и не должны совпадать с зарезервированными словами. Регистр букв в имени переменной не имеет значения (т.е. переменные Var1, VAR1 и var1) не различаются. В именах переменных допустим знак подчеркивания: (переменные A_BCD и AB_CD) считаются разными. Идентификатор не должен содержать подряд более одного символа подчеркивания. Длина индентификатора не ограничена, все символы являются значимыми.

Все переменные и типы данных можно инициализировать. Для этого используется оператор «=». Переменные простейших типов инициализируются константами. По умолчанию все переменные инициализируются нулем.

Пример:

iVar1:INT:=12; (*Переменная типа INT, инициализируемая числом 12*).

Если требуется поместить переменную по определенному адресу, то следует объявить ее с ключевым словом **AT**.

4.2.2.1 Методы объявления переменных

В ПО CoDeSys применяются три метода объявления переменных: текстовый (описан выше, в разделе 4.2.2), табличный и автоматический.

4.2.2.2 Табличное объявление переменных

Табличный способ объявления переменных позволяет ускорить процедуру объявления переменных. Для вызова окна табличного объявления следует выбрать команду «Объявления в форме таблицы (Declarations as Tables)» контекстного меню окна редактора объявлений. Окно редактора объявлений примет вид, изображенный на рисунке 4.5. На вкладках окна редактора отображаются списки переменных различных типов. В ячейках таблицы списки переменных могут быть дополнены новыми переменными. Значения атрибутов переменных могут быть введены или отредактированы также в ячейках таблицы. Кроме того, требуемые переменные могут быть не только отредактированы, но и удалены из списков.

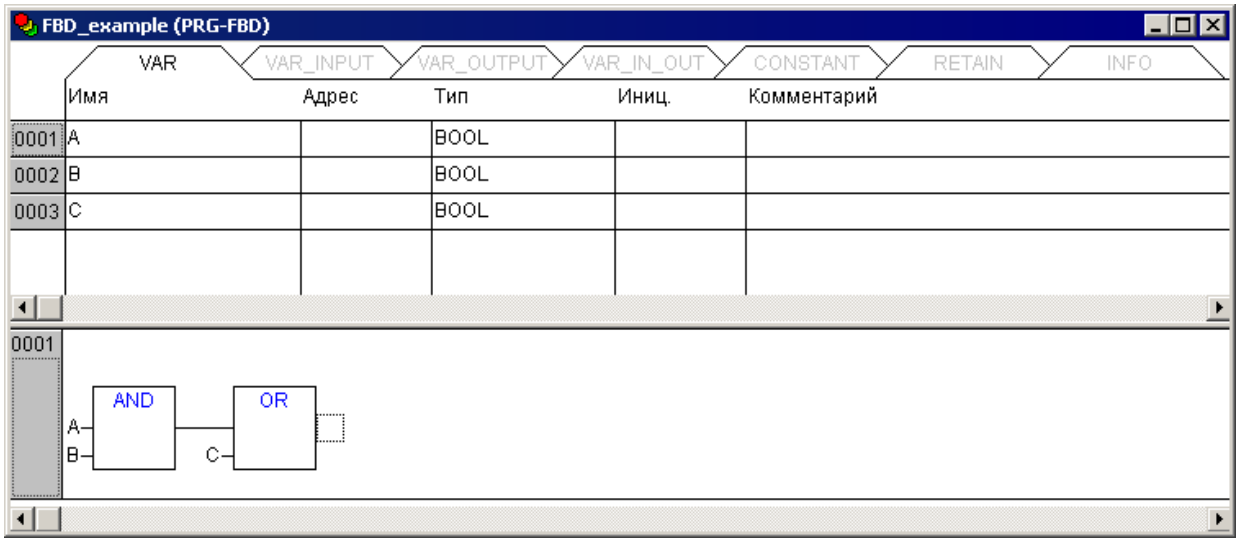


Рисунок 4.5 – Окно редактора объявлений (в табличной форме) (верхняя часть окна редактора POU)

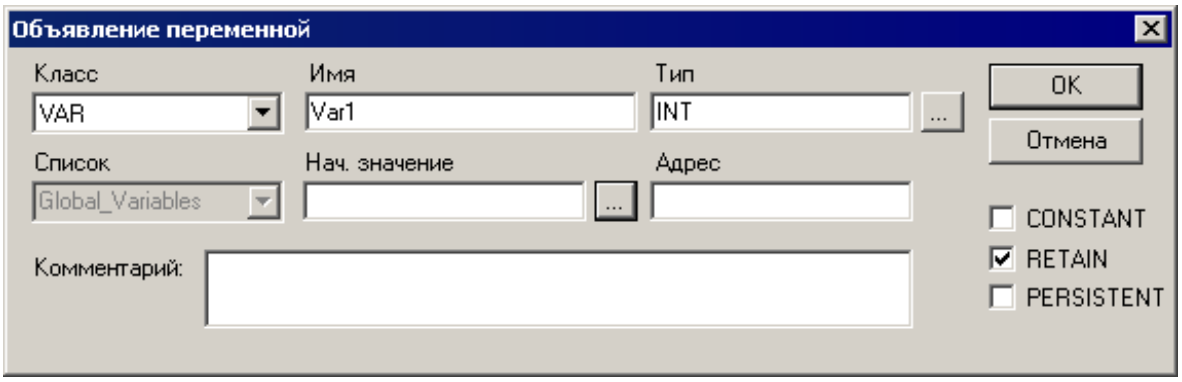
4.2.2.3 Автоматическое объявление переменных

Автоматическое объявление переменных позволяет автоматизировать ввод значений ряда атрибутов переменной, что позволяет ускорить и упростить процедуру ввода и одновременно избежать ошибок, возможных при ручном вводе.

Для вызова окна автоматического объявления переменных следует выбрать команду «Автообъявление (Auto Declare)» контекстного меню окна редактора объявлений.

В открывшемся окне (см. рисунок 4.6, а) задается имя добавляемой переменной (в поле «Имя»). В других полях окна значения задаются выбором либо из раскрывающегося списка, либо из списков, отображаемых в специальных окнах. Например, выбор требуемого типа переменной производится в окне «Ассистент ввода» (см. рисунок 4.6, б), которое открывается по нажатию кнопки с тремя точками, размещенной у правого края поля «Тип».

а)



б)

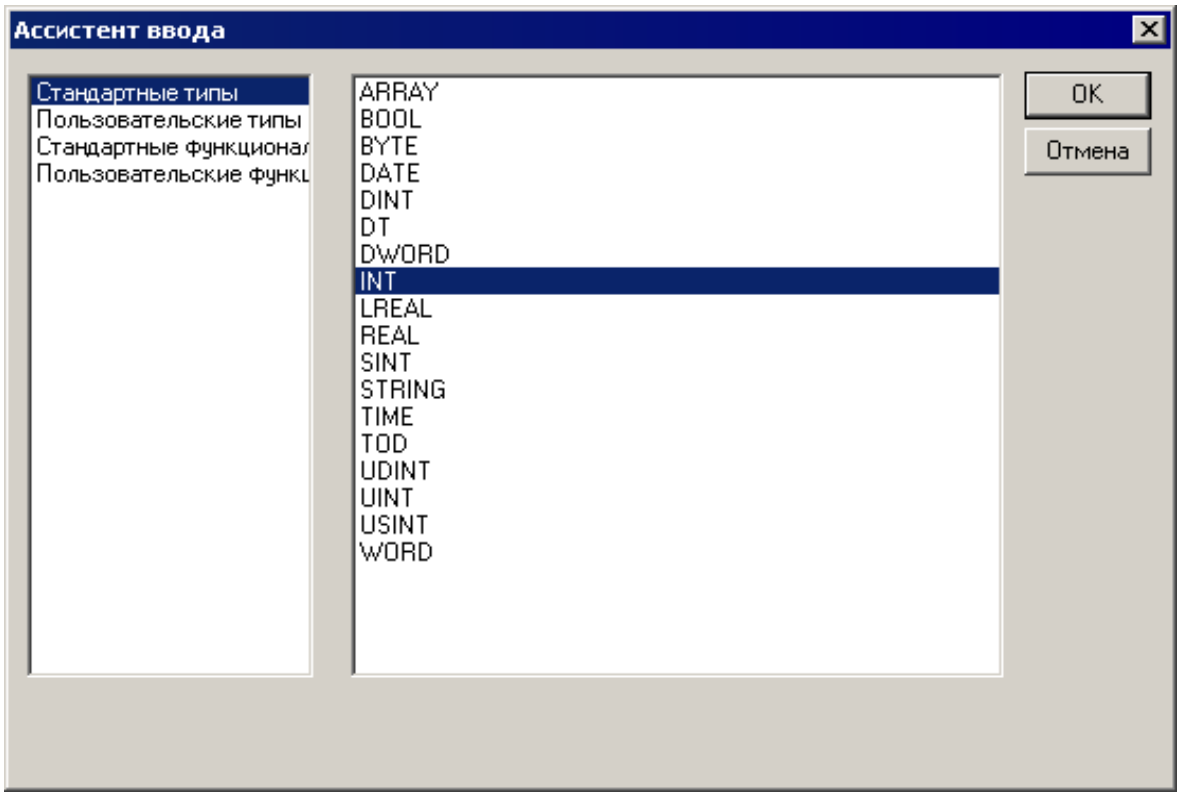


Рисунок 4.6 – Окна «Объявление переменной» (а) и «Ассистент ввода» (б)

4.2.3 Типы данных

Тип данных определяет род информации, методы ее обработки и хранения, количество выделяемой памяти. Программист может непосредственно использовать элементарные (базовые) типы данных или создавать собственные (пользовательские) типы на их основе.

4.2.3.1 Базовые типы данных

4.2.3.1.1 Логический (BOOL)

BOOL – логический тип данных. Переменная может принимать 2 значения: ИСТИНА (TRUE) или ЛОЖЬ (FALSE). Занимает 8 бит памяти (если не задан прямой битовый адрес).

4.2.3.1.2 Целочисленные

BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, и **UDINT** – целочисленные типы данных. Они отличаются диапазонами сохраняемых данных и требованиями к памяти. Подробно их характеристики приведены в таблице 4.1.

Таблица 4.1 – Характеристики целочисленных типов данных

Тип	Нижний предел	Верхний предел	Размер памяти
BYTE	0	255	8 Бит
WORD	0	65535	16 Бит
DWORD	0	4294967295	32 Бит
SINT	-128	127	8 Бит
USINT	0	255	8 Бит
INT	-32768	32767	16 Бит
UINT	0	65535	16 Бит
DINT	-2147483648	2147483647	32 Бит
UDINT	0	4294967295	32 Бит

4.2.3.1.3 Рациональные

REAL и **LREAL** – данные в формате с плавающей запятой, используются для сохранения рациональных чисел. Для типа REAL необходимо 32 бита памяти, для LREAL – 64 бита.

Диапазон значений REAL: от [1.175494351e-38] до [3.402823466e+38].

Диапазон значений LREAL: от [2.2250738585072014e-308] до [1.7976931348623158e+308].

4.2.3.1.4 Строки

Строковый тип STRING представляет строки символов. Максимальный размер строки определяет количество резервируемой памяти и указывается при объявлении переменной. Размер задается в круглых или квадратных скобках. Если размер не указан, принимается размер по умолчанию – 80 символов.

Длина строки в CoDeSys не ограничена, но строковые функции способны обращаться со строками от 1 до 255 символов.

Пример объявления строки размером до 35 символов:

str:STRING(35):='Просто строка'';

4.2.3.1.5 Время и дата

- **TIME** представляет длительность интервалов времени в миллисекундах. Максимальное значение для типа TIME: 49d17h2m47s295ms (4194967295 ms).

- **TIME**, **TIME_OF_DAY** (сокр. **TOD**) содержит время суток, начиная с 0 часов (с точностью до миллисекунд). Диапазон значений **TOD:** от 00:00:00 до 23:59:59.999.
- **DATE** содержит календарную дату, начиная с 1 января 1970 года. Диапазон значений от: 1970-00-00 до 2106-02-06.
- **DATE_AND_TIME** (сокр. DT) содержит время в секундах, начиная с 0 часов 1 января 1970 года. Диапазон значений от: 1970-00-00-00:00:00 до 2106-02-06-06:28:15.

Типы **TIME, TOD, DATE** и **DATE_AND_TIME** (сокр. **DT**) сохраняются физически как **DWORD**.

4.2.3.2 Пользовательские типы данных

Кроме стандартных типов данных (см. раздел 4.2.3.1), в проектах можно использовать определяемые пользователем сложные типы данных: массивы, перечисления, структуры и некоторые другие (см. раздел 5).

4.2.4 Подключение дополнительных программных модулей

Подключение требуемых дополнительных программных модулей (библиотек) производится в окне «Менеджер библиотек (Library Manager)» ПО CoDeSys.

Вызов окна производится выбором команды Окно | Менеджер библиотек (Window | Library Manager)» или выбором пункта «Менеджер библиотек (Library Manager)» в дереве ресурсов (на вкладке «Ресурсы» Организатора объектов).

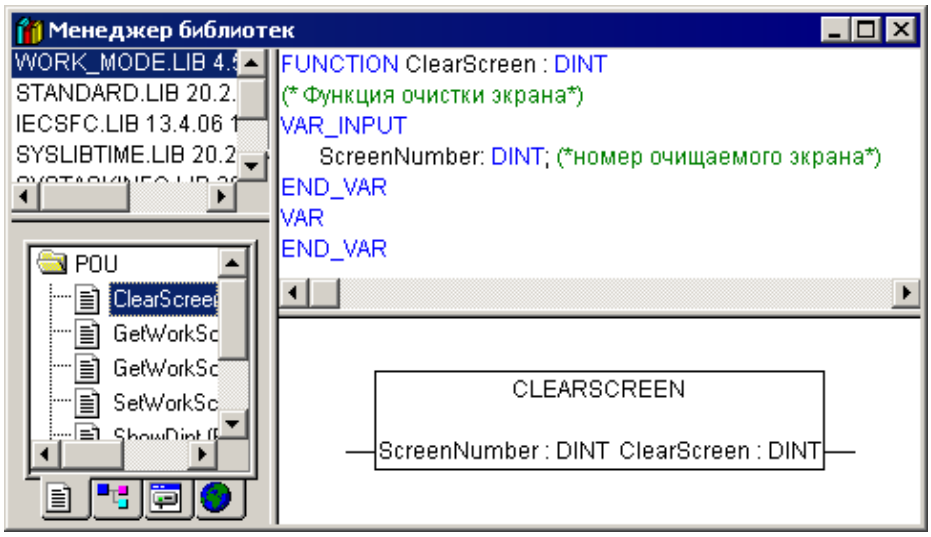


Рисунок 4.7 – Окно «Менеджер библиотек (Library Manager)»

Для подключения библиотеки следует:

1) Выбрать команду «Добавить библиотеку (Add library)» контекстного меню списка подключенных библиотек (отображаемого в верхней левой области окна режима) или команду «Вставка | Добавить библиотеку (Insert | Additional Library)» главного меню.
2) В открывшемся окне выбора файлов следует выбрать файл требуемой библиотеки и нажать кнопку «Открыть». Выбранная библиотека будет подключена к проекту. Ее наименование отобразится в списке установленных библиотек (в верхней левой области окна режима).
   Примечание. Файлы библиотек предварительно следует скопировать на жесткий диск компьютера (рекомендуется в папку c:\Program Files\3S Software\CoDeSys V2.3\Library\, где размещены файлы биб-

лиотек, используемых ПО CoDeSys).

Для удаления подключенной библиотеки следует:

1) Выделить требуемую запись в списке подключенных библиотек (отображаемого в верхней левой области окна режима).
2) Выбрать команду «Удалить (Delete)» контекстного меню списка. Выделенная библиотека будет отключена от проекта.

Для того, чтобы включить в проект дополнительный программный модуль (то есть модуль, содержащийся в подключенной к проекту библиотеке), следует выполнить следующие операции:

1) Перейти на вкладку «POU» Организатора объектов.
2) В дереве программных компонентов объекта выбрать требуемый.
3) Выбрать команду «Правка | Ассистент ввода (Edit | Input Assistant)» главного меню или команду «Ассистент ввода (Input Assistant)» контекстного меню редактора объявлений.
4) В открывшемся окне «Ассистент ввода» (см. рисунок 4.8), в левой части, где отображается перечень доступных типов добавляемых объектов, – выделить требуемый тип (в данном случае – «Стандартные функциональные блоки»).
   В правой части окна при этом отобразится перечень доступных объектов выбранного типа. При этом: если флажок переключателя «Структурно» в нижней части окна установлен, то перечень отображается в виде иерархического структурированного списка, см. рисунок 4.5; если флажок не установлен, то перечень отображается в виде отсортированного по алфавиту линейного списка.
5) В перечне доступных объектов (в правой части окна) – выбрать требуемый объект и нажать кнопку «ОК» окна. Выбранный объект (в данном случае – стандартный функциональный блок) будет вставлен в редактируемый программный компонент проекта. Для отказа от добавления блока – нажать кнопку «Отмена» окна.

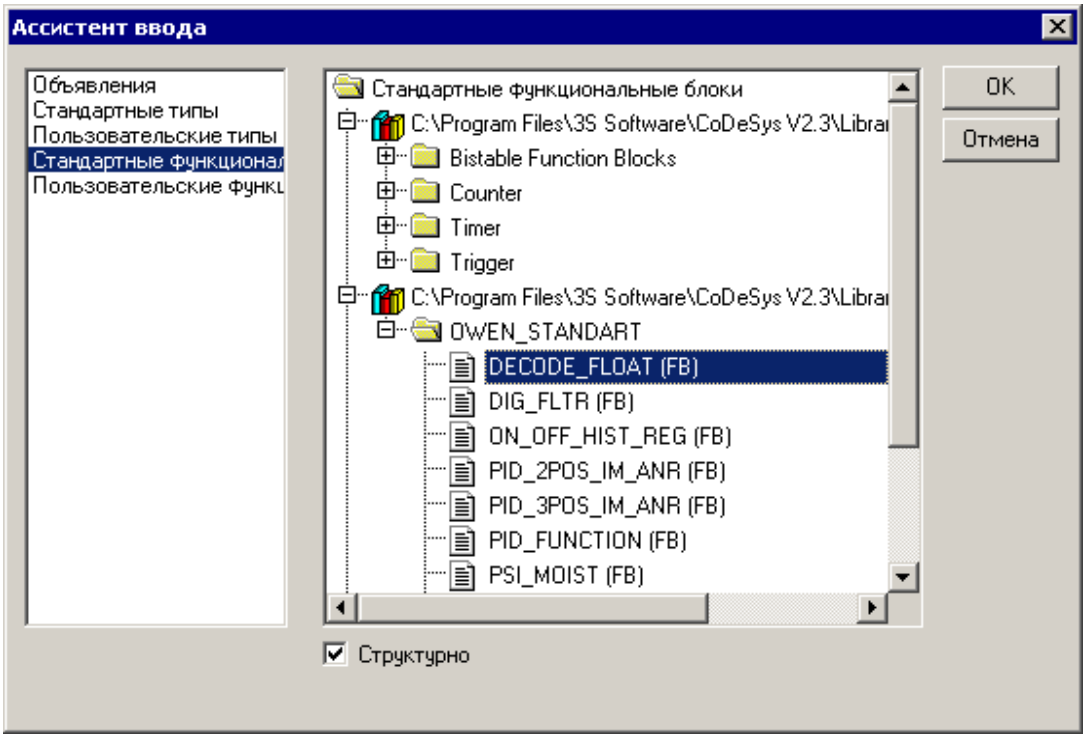


Рисунок 4.8 – Окно «Ассистент ввода (Input Assistant)»

4.2.4.1 Доступные дополнительные программные модули

В дистрибутив ПЛК включен ряд дополнительных программных модулей (библиотек). Подробное описание библиотек ОВЕН приведено в документах, описывающих библиотеки программных компонентов для ПЛК ОВЕН (см. приложение Н, [7 - 14]).

Описания системных библиотек CoDeSys доступны на сайте компании 3S Software и на дистрибутивном диске ПЛК.

ОВЕН ПЛК63 и ОВЕН ПЛК73 поддерживают следующие библиотеки программных компонентов:

- ModBus.lib;
- owen_fb.lib;
- OwenNet.lib;
- pid_reg2.lib;
- ARM7_specific.lib;
- sound.lib;
- Ind_Mode.lib
- Work_Mode.lib

В данном документе описаны только библиотеки «Ind_Mode.lib» и «Work_Mode.lib», содержащие функции, предназначенные для программирования ЖКИ контроллера.

4.2.5 Библиотеки программных компонентов для работы с ЖКИ ПЛК

В состав дистрибутива ПЛК включены две специализированные библиотеки дополнительных программных компонентов, содержащие функции, предназначенные для программирования ЖКИ контроллера.

Более подробное описание этих библиотек см. в разделе 8.4 и в документе «Библиотеки “Ind_mode” и “Work_Mode”» (см. приложение Н [7]).

4.2.6 Создание и использование дополнительных программных модулей

При необходимости дополнительные программные модули могут быть разработаны и применены пользователем. Такая необходимость может возникнуть в том случае, если применяемая программа должна содержать алгоритмы, которые не могут быть написаны с использованием готовых программных модулей.

Для создания пользовательского программного модуля следует:

1) Создать новый проект (см. раздел 3.3.2).
2) В рамках проекта – создать объект типа «Функциональный блок» (например, см. рисунок 4.9).

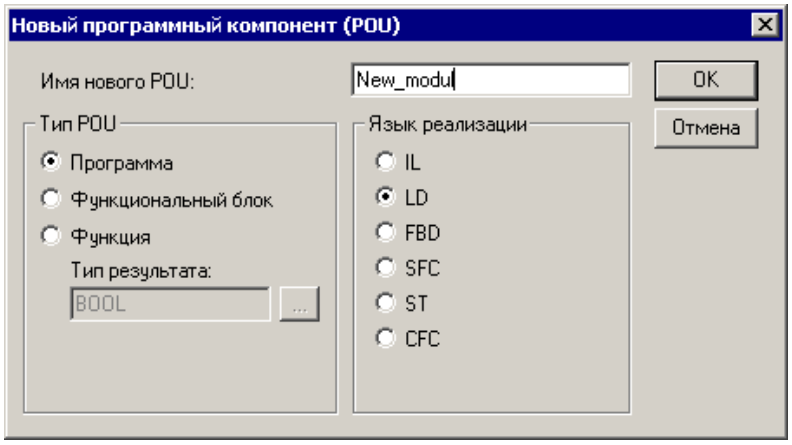


Рисунок 4.9 – Окно «Новый программный компонент (New Pou)»

3) Написать программу этого функционального блока, которую предполагается использовать в качестве пользовательского программного модуля (например, см. рисунок 4.10).

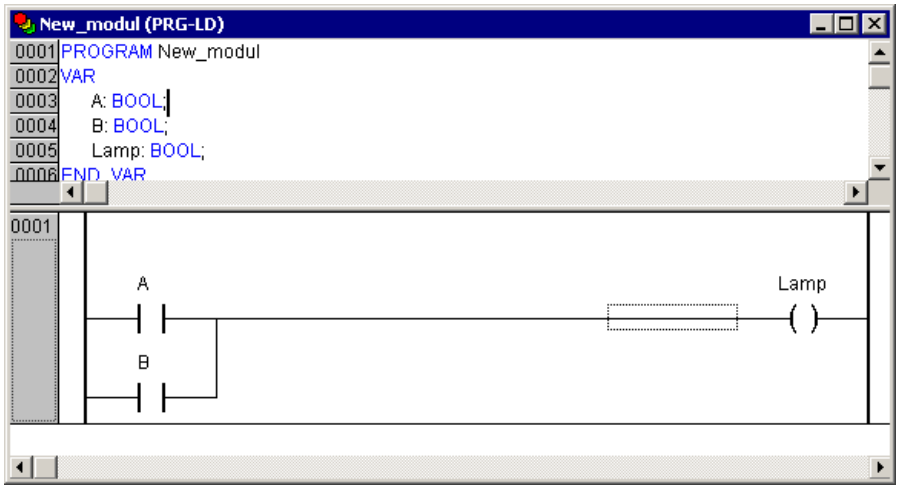


Рисунок 4.10 – Пример текста нового программного компонента

4) Удалить из проекта программный компонент PLC_PRG (команда «Удалить объект» контекстного меню объекта в дереве объектов), оставив в нем только требуемый функциональный блок.
5) Сохранить POU командой «Файл | Сохранить как», задав в открывшемся окне, в поле «Тип файла» тип файла – «Внешняя библиотека (*.lib)» и нажав кнопку «Сохранить».
6) Сохраненный таким образом функциональный блок следует затем

подключить к разрабатываемому проекту аналогично тому, как подключаются готовые программные модули (см. раздел 4.2.4)

7) После этого новый функциональный блок отобразится в перечне доступных стандартных функциональных блоков окна «Ассистент ввода» и может быть добавлен в текущий проект аналогично тому, как это выполняется для функциональных блоков из состава поставляемых библиотек (см. раздел 4.2.4), см. рисунок 4.11.

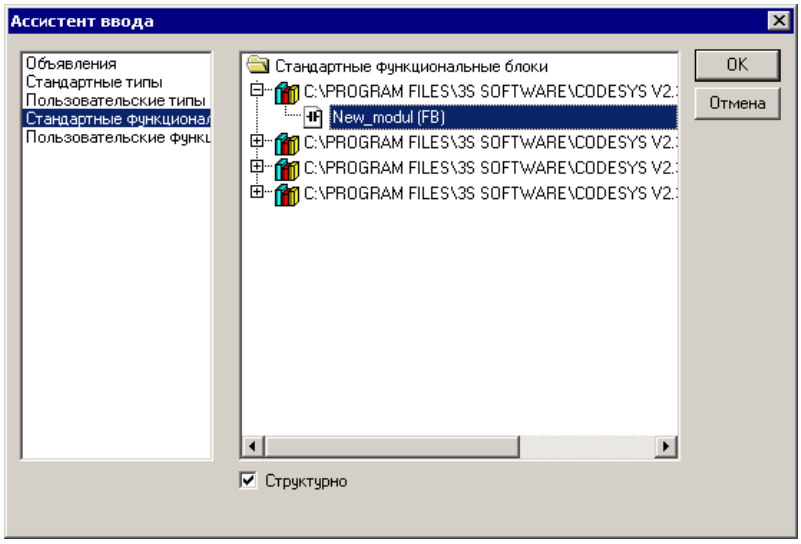


Рисунок 4.11 – Окно «Ассистент ввода»

4.3 Использование ЖКИ

ЖКИ контроллера предназначен для отображения и установки значений программируемых параметров прибора, определяющих его настройку и работу в процессе эксплуатации.

- **Рабочий режим индикации** – режим индикации, включающийся по умолчанию, в котором ЖКИ управляется пользовательской программой.
- **Конфигурационный режим индикации (режим меню)** – режим индикации, в котором производится задание конфигурационных параметров: параметров конфигурации периферийного оборудования (входов, выходов и т.д.) и пользовательских конфигурационных параметров (уставок, аварийных порогов, коэффициентов ПИД-регуляторов и т.д.).
- **Отладочный режим индикации** – служебный режим индикации, необходимый для калибровки и заводской проверки контроллера, просмотра возможных ошибок контроллера.

Подробнее о работе с ЖКИ см. раздел 8 .

4.4 Отладка проекта

ПО CoDeSys располагает несколькими инструментами отладки проекта. Их краткие описания приведены ниже.

Внимание!
В режиме отладки ПЛК работает существенно медленнее. Из-за этого возможны прерывания связи по таймауту.
Поэтому при отладке следует перенастроить таймауты в окне настройки, вызываемом командой «Проект | Опции (Project | Options)» главного меню ПО CoDeSys: на вкладке «Рабочий стол (Desktop)», значение па-

раметра «Таймаут связи (Communication timeout)» увеличить до 15000.

4.5 Многозадачность

По умолчанию в проекте всегда создается единственная «главная» программа PLC_PRG, выполняемая циклически (см. раздел 3.3.3).

Кроме того, в проекте можно явно определить несколько **задач** с различными условиями выполнения. Задача – это единица обработки программы.

	Использование задач при создании пользовательских программ ПЛК63 и ПЛК73 не рекомендуется ввиду их ресурсоемкости.

Каждая задача имеет название, приоритет и тип:

- Название служит идентификатором задачи.
- Тип определяет условие вызова задачи. Условием может служить время (циклическое или свободное – «freewheeling» – выполнение) или событие (например, принятие определенной переменной значения TRUE).
- Приоритет задается числом (от 1 до 15) и в сочетании заданными условиями вызова задачи определяет хронологический порядок выполнения задач.

Для каждой задачи назначается одна или несколько программ (программных компонентов проекта), которые будут в ней выполняться. Если задача выполняется в текущем цикле, то это означает, что выполняются включенные в неё программы (по одному циклу каждая).

Порядок выполнения задач определяется комбинацией приоритетов и условий вызова задач.

Каждую задачу можно разрешить или запретить независимо от других.

Определение задач в рамках редактируемого проекта производится в окне «Конфигурация задач (Task Configuration)», открывающемся при выборе строки «Конфигурация задач (Task Configuration)» в дереве объектов («ресурсов») на вкладке Ресурсы (Resources) Организатора объектов.

Окно «Конфигурация задач» (см. рисунок 4.12) разделено на две части.

В левой части окна отображается перечень задач текущего проекта в виде дерева конфигурации. В корневой позиции обязательно присутствует элемент «Конфигурация задач (Task Configuration)». Под ним раскрывается список конкретных задач, представленных по именам.

Для добавления в проект новой задачи следует выбрать команду «Вставка | Вставить задачу» главного меню или команду «Вставить задачу» контекстного меню дерева задач.

При добавлении задачи в проект она включается в дерево задач и снабжается пиктограммой, отображающей тип задачи:

- – циклически выполняемые задачи;
- – задачи, выполняемые по времени (свободному);
- – задачи, выполняемые по событию (связанному с глобальными переменными проекта);
- – задачи, выполняемые по внешнему событию.

В правой части окна отображаются поля задания атрибутов текущей (выбранной в дереве задач) задачи. Набор полей соответствует атрибутам задачи выбранного типа.

Задавая значения атрибутов, можно конфигурировать свойства задач (Task properties), вызова программ (Program call).

	Внимание! Если в Конфигураторе задач (Task Configuration) определена последовательность выполнения задач, то проект может не содержать PLC_PRG. Но если ‘Конфигурация задач (Task Configuration) не используется в проекте, то удалять или переименовывать POU PLC_PRG нельзя: PLC_PRG является главной программой в однозадачном проекте.

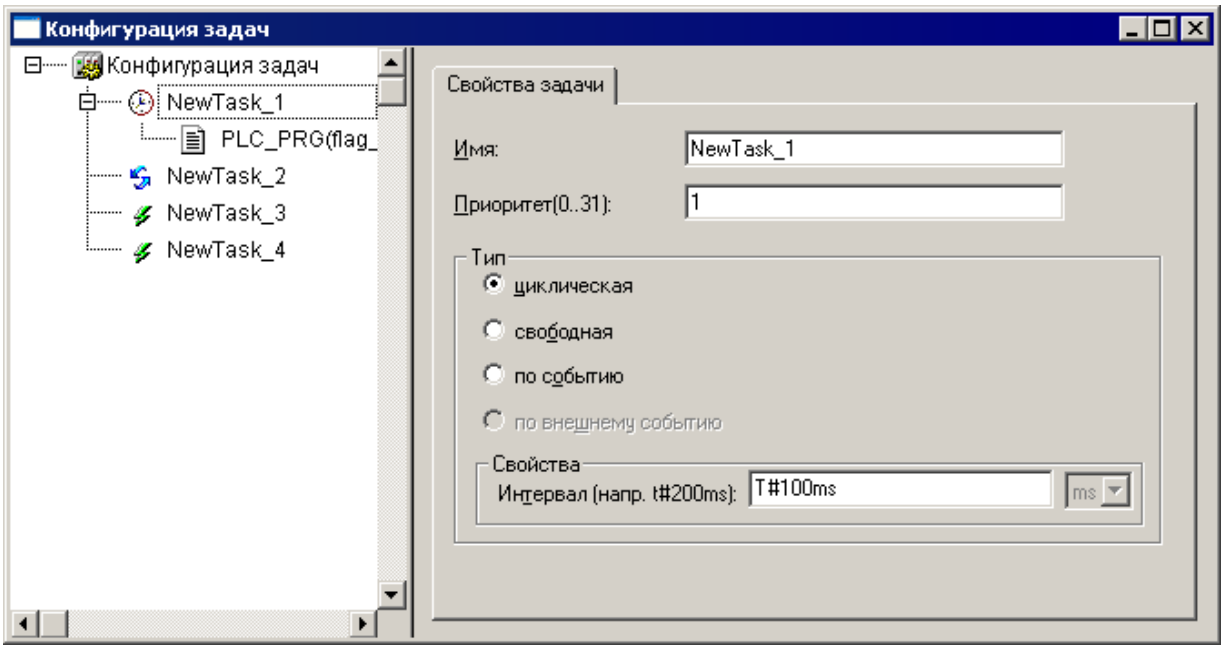


Рисунок 4.12 – Окно «Конфигурация задач (Task Configuration)». Конфигурирование задач

Для привязки к текущей задаче вызова определенного программного компонента следует выбрать в контекстном меню наименования задачи (в дереве задач) команды «Добавить Вызов программы». В открывшемся окне «Ассистент ввода» – выбрать требуемый программный компонент и нажать кнопку «ОК».

Примечание. Не следует использовать одни и те же строковые функции в разных задачах: это может привести к ошибкам перезаписи данных.

В режиме «Онлайн» выполнение задач можно наблюдать в виде графической диаграммы.

4.5.1 Отладка

Опция отладки ПО CoDeSys заставляет компилятор формировать дополнительный код, упрощающий поиск ошибок. Опция «Отладочный код (Debugging)» включается установкой флажка переключателя «Отладочный код (Debugging)» в окне «Опции (Options)», вызываемом командой «Проект | Опции (Project | Options)» главного меню, на вкладке «Генератор кода (Build)» (см. рисунок 4.13).

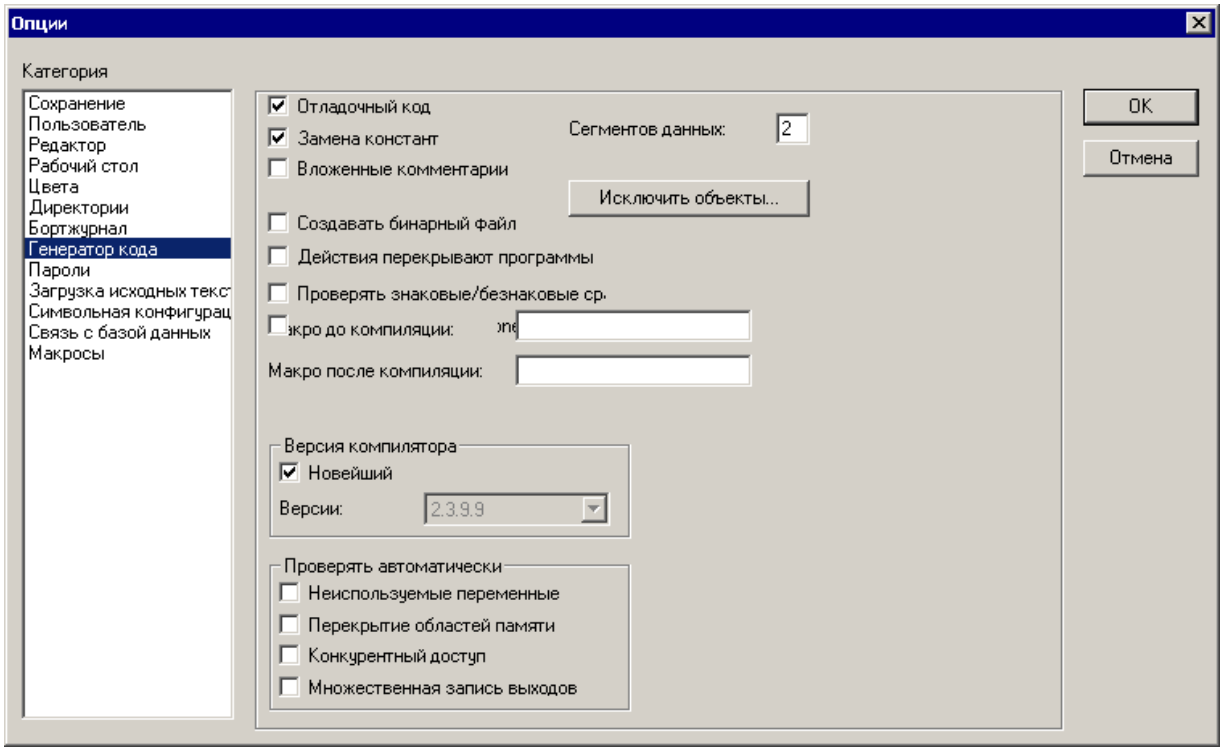


Рисунок 4.13 – Окно «Опции (Options)», вкладка «Генератор кода»

4.5.2 Точки останова

Точки останова – это места, в которых выполнение программы будет приостанавливаться, что позволяет просмотреть значения переменных на определенном этапе работы программы. Точки останова можно задавать во всех редакторах. В текстовом редакторе точка останова устанавливается на номер строки, в FBD и LD – на графический элемент, в SFC – на шаг.

Внимание! **Система исполнения CoDeSys SP32 Bit Full автоматически деактивирует сторожевой таймер задачи, если она выходит на точку останова.**

4.5.3 Пошаговое выполнение

Пошаговое выполнение позволяет проверить логическую правильность программы.

Под «шагом» подразумевается:

- В IL: Выполнить программу до следующего оператора CAL, LD or JMP.
- В ST: Выполнить следующую инструкцию.
- В FBD, LD: Выполнить следующую цепь.
- В SFC: Продолжить действие до следующего шага.

4.5.4 Выполнение по циклам

Команда «Онлайн | Один цикл (Online | Single Cycle)» выполняет один рабочий цикл и останавливает контроллер после выполнения.

4.5.5 Эмуляция

Режим эмуляции последовательно включается и отключается выбором команды «Онлайн | Режим эмуляции (Online | Emulation)» главного мню. Включенный режим маркируется устновленным флажком в строке главного меню и записью «Эмул.» в строке состояния главного окна.

Во время эмуляции созданная программа выполняется не в ПЛК, а в компьютере, на котором запущено ПО CoDeSys. В этом режиме допустимы все функции онлайн, что позволяет проверить логическую правильность программ, не используя контроллер.

Внимание! **В режиме эмуляции функции внутренних библиотек не выполняются.**

4.5.6 Бортжурнал (Log)

«Бортжурнал (Log)» хронологически записывает действия пользователя, внутренние сообщения системы исполнения, изменения состояния и исключения в режиме онлайн. Это позволяет анализировать условия возникновения ошибки при отладке программы.

Просмотр записей «Бортжурнала (Log)» производится в режиме, вызываемом выбором строки «Бортжурнал (Log)» дерева ресурсов проекта на вкладке «Ресурсы» организатора объектов CoDeSys.

5 Использование сложных структур данных

Кроме стандартных типов данных (см. раздел 4.2.3.1), в проектах можно использовать определяемые пользователем сложные типы данных (массивы, перечисления, структуры и некоторые другие): объекты (переменные или постоянные) которые имеют внутреннюю структуру, доступную программисту. Их использование позволяет произвольно конструировать требуемые структуры данных из небольшого набора предопределённых типов.

Чем адекватнее используемая в программе структура данных реальному объекту автоматизации, тем безошибочнее и долговечнее будет функционировать разработанная программа.

При программировании ПЛК63 и ПЛК73 используются пользовательские типы данных, заданные в библиотеке дополнительных программных компонентов Ind_Mode.lib (управление режимами ЖКИ и вывод символьных строк на ЖКИ в монопольном рабочем режиме).

5.1 Пользовательские типы данных

5.1.1 Массивы

Элементарные типы данных могут образовывать одно-, двух-, и трехмерные массивы. Массивы могут быть объявлены в разделе объявлений POU или в списке глобальных переменных.

Путем вложения массивов можно получить многомерные массивы, но не более 9-мерных ("ARRAY[0..2] OF ARRAY[0..3] OF …").

Синтаксис:

<Имя_массива>:ARRAY [<ll1>..<ul1>,<ll2>..<ul2>] OF <базовый тип>

Здесь ll1, ll2, ll3 указывают нижний предел индексов; ul1, ul2 и ul3 указывают верхние пределы. Индексы должны быть целого типа. Нельзя использовать отрицательные индексы.

5.1.2 Перечисления

Перечисление – это определяемый пользователем тип данных, задающий несколько строковых псевдонимов для числовых констант.

Перечисление доступно в любой части проекта, даже при локальном его объявлении внутри POU. Поэтому рационально создавать все перечисления на вкладке «Типы данных» Организатора Объектов.

Объявление должно начинаться с ключевого слова TYPE и заканчиваться строкой END_TYPE.

Синтаксис:

TYPE <Имя_перечисления>:(<Элемент_0> ,< Элемент _1>, …
…, < Элемент_n>); END_TYPE

Переменная типа <Имя_перечисления> может принимать только перечисленные значения. При инициализации переменная получает первое значение из заданного списка. Если числовые значения элементов перечисления не указаны явно, то им присваиваются последовательно возрастающие числа, начиная с 0. Фактически элемент перечисления – это число типа INT, и работать с ними можно точно так же. Можно напрямую присвоить число переменной типа перечисление.

Элемент, уже включенный в перечисление, нельзя повторно включать в другое перечисление.

5.1.3 Структуры

Структуры создаются командой «Добавить объект (Add Object)» контекстного меню вкладки «Типы данных» Организатора Объектов.

Новый объект отображается в дереве объектов, окно задания параметров объекта открывается в рабочей области главного окна ПО CoDeSys (см. рисунок 5.1).

Объявление должно начинаться с ключевых слов TYPE и STRUCT и заканчиваться строками END_STRUCT и END_TYPE.

Синтаксис:

TYPE <Имя _структуры>:
STRUCT
<Объявление переменной 1>
.
.
<Объявление переменной n>
END_STRUCT
END_TYPE

<Имя _структуры> образует новый тип данных, который может быть использован в любой части проекта наряду с базовыми типами.

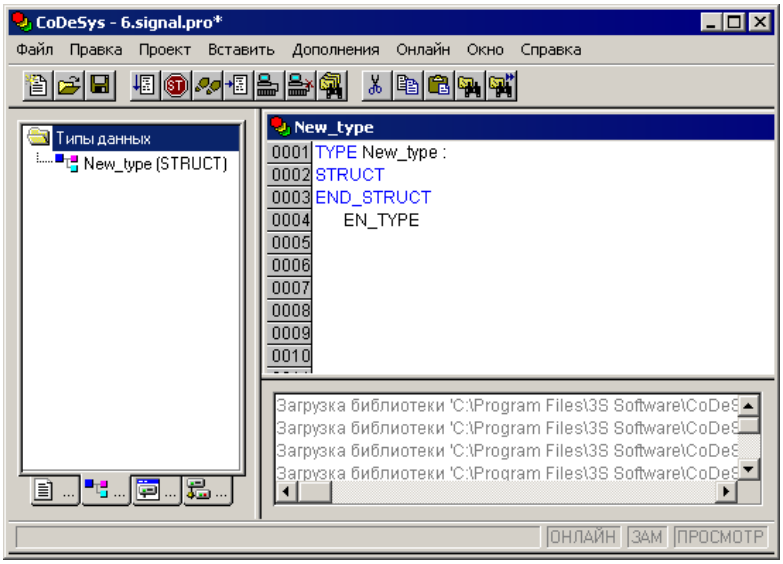


Рисунок 5.1 – Вкладка «Типы данных»

Допускаются вложенные структуры, но запрещено размещение элементов структуры по прямым адресам (в частности, недопустимы **AT** объявления).

Для доступа к элементам структуры используется следующий синтаксис:

<Имя_структуры>.<Имя_компонента>

Например, если структура "Week" содержит компонент "Monday", то обращение к нему будет выглядеть так: **Week.Monday.**

5.1.4 Указатели

Указатели позволяют работать с адресами переменных или функциональных блоков.

Синтаксис:

<Имя_указателя>: POINTER TO <Тип данных/Функциональный блок>;

Указатели применимы для всех базовых типов данных или функциональных блоков, включая определяемые пользователем.

6 Визуализация проекта

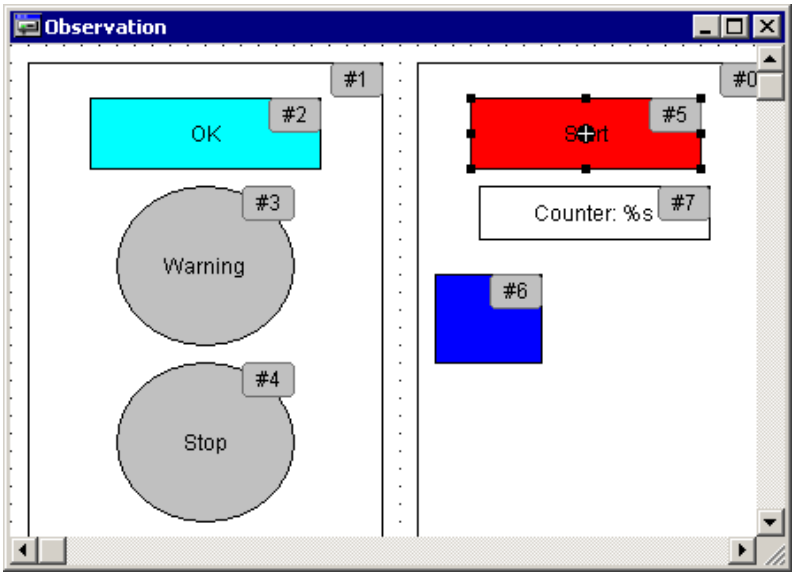
Визуализация проекта предназначена для графического представления объекта управления и непосредственно связана с созданной в ПО CoDeSys программой контроллера. Редактор визуализации CoDeSys предоставляет набор готовых графических элементов, которые могут быть связаны требуемым образом с переменными проекта.

Например, если в программе доступна переменная, связанная с уровнем заполнения некоторой емкости, то в визуализации ее можно изобразить графическим элементом в виде полосы, которая, в зависимости от значения переменной проекта, будет изменять свою длину и/или цвет.

В Online режиме представление элементов визуализации на экране изменяется в зависимости от значений переменных.

Свойства отдельных элементов визуализации, а также визуализации в целом устанавливаются в соответствующих диалогах конфигурации и диалоге свойств объекта (см. рисунок 6.1, а, б). Здесь определяется начальный вид элементов и выполняется привязка динамических свойств к значениям переменных проекта.

а)



б)

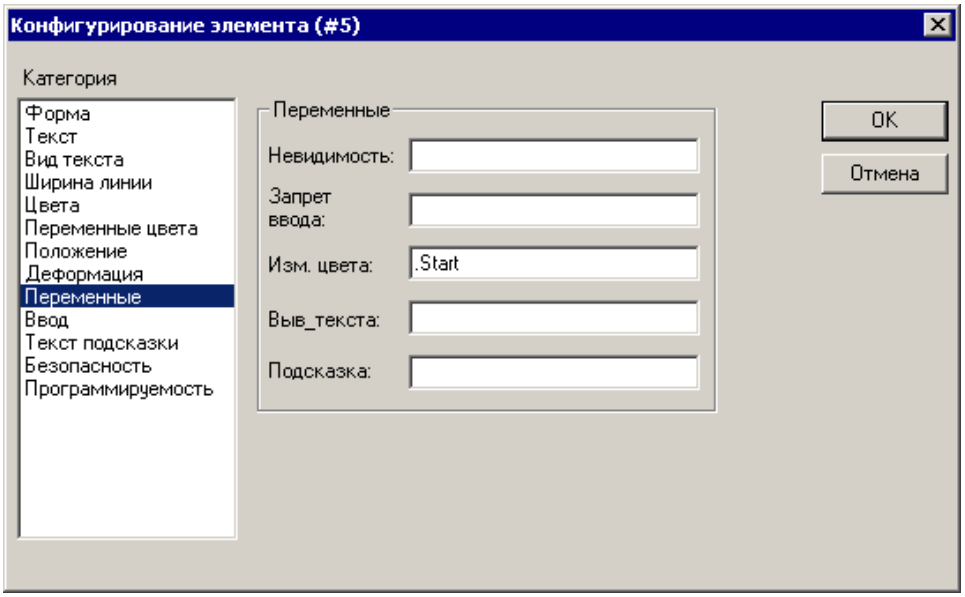


Рисунок 6.1 – Окно визуализации проекта (а) и окно конфигурирования элемента визуализации (б)

Визуализация проекта может использоваться как пользовательский интерфейс для контроля и управления работой ПЛК программы в рабочем режиме. В определенных ситуациях – при необходимости, например, исключить возможность вмешательства оператора в программу работы – как единственный пользовательский интерфейс.

В этом случае ввод данных для программы должен выполняться исключительно посредством элементов визуализации. Такую возможность обеспечивают специальные возможности ввода, задаваемые в процессе конфигурации. Кроме того, предусмотрено создание клавиш быстрого ввода для каждой конкретной визуализации.

Созданная в CoDeSys визуализация может использоваться еще несколькими способами:

- Программа Win32 CoDeSys HMI отображает формы визуализации на ПК в полноэкранном режиме. В отличие от ПО CoDeSys, эта программа не бесплатна.
- Web-визуализация отображает данные и предоставляет возможность удаленного управления через Интернет.

Подробнее о создании окон визуализации см. документ «Визуализация CoDeSys. Дополнение к руководству пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [4]).

6.1 CoDeSys HMI

CoDeSys HMI – это система исполнения визуализаций созданных в среде программирования CoDeSys.

Если проект содержит визуализацию, то при запуске CoDeSys HMI она будет воспроизводиться в полноэкранном режиме. Пользователь сможет использовать заданные в программе функции управления и отображения при помощи мыши и клавиатуры, причем, даже если проект CoDeSys защищен от чтения.

Возможность редактирования программ, меню и панели инструментов CoDeSys не доступны пользователю. Поэтому все необходимые функции управления и отображения данных должны быть сопоставлены соответствующим элементам визуализации. Для этого в диалоге конфигурации элементов визуализации предусмотрены специальные возможности ввода для CoDeSys HMI.

7 Конфигурирование контроллера

7.1 Конфигурация памяти ввода / вывода

В процессе создания и отладки проекта необходимо настроить конфигурацию входов, выходов и интерфейсов связи ПЛК с внешними модулями ввода-вывода, устройствами индикации или иными устройствами, обмен данными с которыми будет производиться по сети (см. раздел 3.4).

Внешние устройства обмениваются данными с пользовательской программой ПЛК через специальную область памяти ПЛК: область памяти ввода / вывода ПЛК (**%I** и **%Q**). Она включает дискретные и аналоговые входы и выходы, модули расширения функционала (в том числе организующие обмен информацией между ПЛК и отдельными приборами и устройствами, связанными по сети с ПЛК).

Размер памяти ввода / вывода определяется типом лицензии CoDeSys контроллера ОВЕН ПЛК (см. раздел 2.2).

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

Для входа в режим редактирования конфигурации ПЛК следует перейти на вкладку «Ресурсы» Организатора объектов. В «дереве ресурсов» следует выбрать пункт «Конфигурация ПЛК (PLC Configuration)». В рабочей области главного окна откроется окно редактора (см. рисунок 7.1).

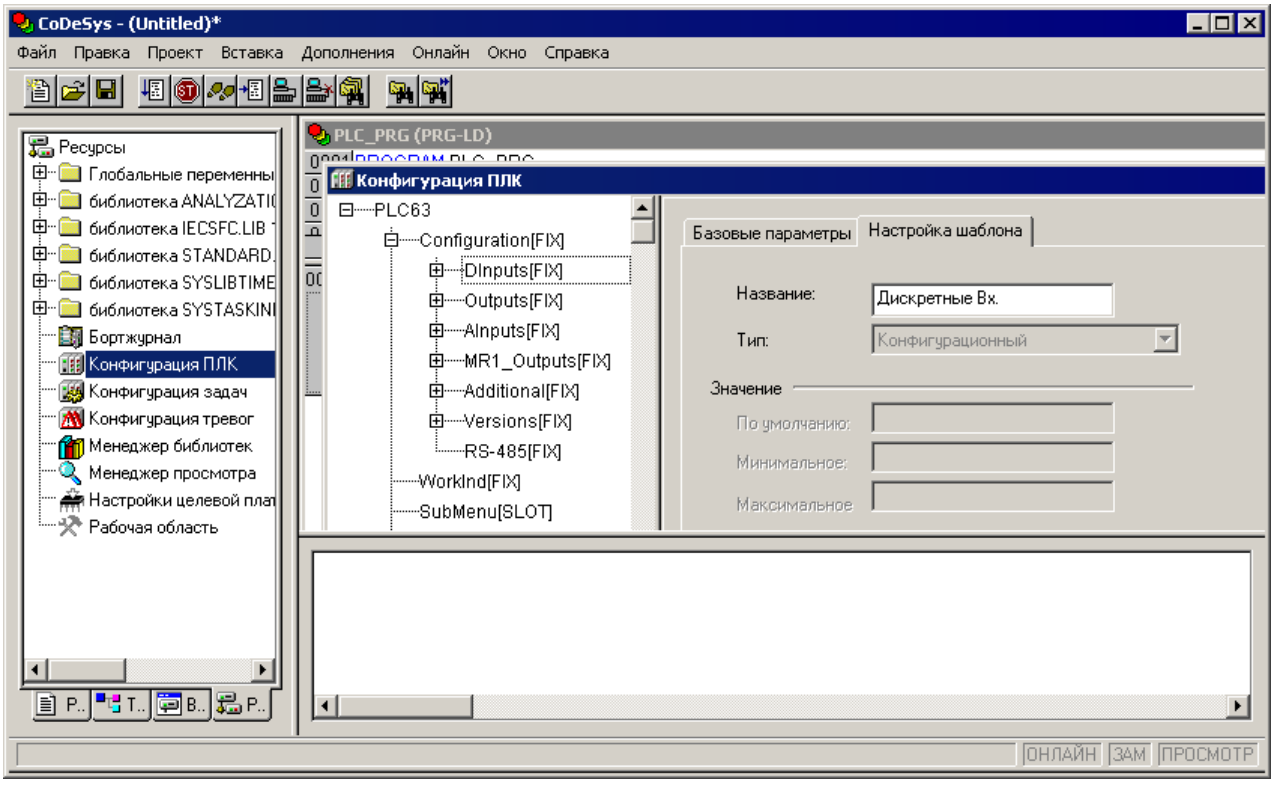


Рисунок 7.1 – Вход в режим «Конфигурация ПЛК (PLC Configuration)»

Окно редактора конфигуратора 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 (см. раздел 3.2) конфигурации, но могут быть изменены пользователем CoDeSys. В правой части окна отображаются диалоги конфигурации, доступные для текущего (выделенного) элемента дерева конфигурации. Диалоги отображаются в виде одной или нескольких табличных вкладок (см. рисунок 7.2). В полях, расположенных на вкладках диалогов, задаются требуемые значения параметров канала или модуля.

Значение параметра устанавливается интерактивно до компиляции проекта. Оно передается в ПЛК и влияет на работу аппаратуры.

Примечание. Правая часть окна видна по умолчанию, но может быть скрыта выбором команды меню «Дополнения (Extras) | Свойства (Properties)» главного меню: последовательные щелчки левой кнопкой мыши на строке команды включают (в строке при этом отображается «галочка») и отключают («галочка» отсутствует) отображение.

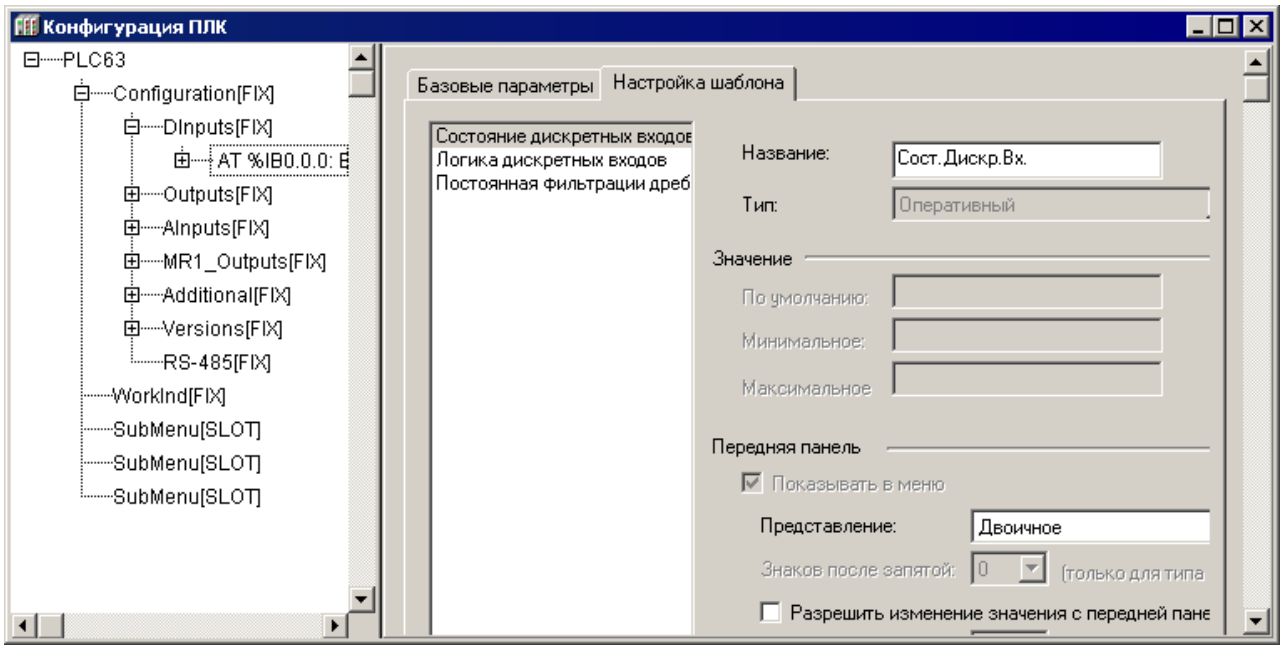


Рисунок 7.2 – Окно режима «Конфигурация ПЛК (PLC Configuration)»

Корневой элемент конфигурационного дерева определяется используемым файлом настроек целевой платформы. Если проект создается без установки настроек целевой платформы, или если в процессе создания проекта файл настроек целевой платформы был заменен другим (т.е. был совершен переход на другую платформу), то вместо отображения дерева конфигурации может отобразиться запись «Not found (Не найдено)». В этом случае следует выбрать команду «Дополнения | Стандартная конфигурация (Extras | Standart Configuration)» главного меню, и в окне отобразится дерево конфигурации, соответствующее текущему файлу настроек целевой платформы.

Конфигурация ПЛК определяет аппаратные средства системы. В дереве конфигурации задается распределение адресов входов / выходов контроллера, что определяет привязку проекта к аппаратным средствам. На основе описания конфигурации ПЛК CoDeSys проверяет правильность задания МЭК адресов, используемых в программах, на их соответствие фактически имеющимся аппаратным средствам.

В дереве конфигурации отображаются следующие элементы:

- **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. Могут иметь вложенные **подмодули** (подэлементы конфигурации).
- **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:** идентификатор отдельного бита в многобитном канале.

В конфигурации присутствуют модули, отвечающие за структурирование областей ввода и/или вывода, каждый из которых может содержать вложенные подэлементы (субмодули и каналы). Для каналов могут быть назначены символические имена. Прямые МЭК адреса отображаются в конфигурации для каждого символического имени.

Определение адресов каналов в области ввода / вывода ПЛК рекомендуется выполнять в автоматическом режиме. Для этого следует установить флажок переключателя «Автоматическое вычисление адресов (Automatic calculation of addresses)» на вкладке «Настройки (Settings)». В этом случае при изменении положения модуля адреса его каналов соответствующим образом смещаются. Альтернативой может служить фиксированная адресация. В этом случае для каждого модуля отводится фиксированное адресное окно, которое определяется физическим расположением (номером слота) модуля. Например: %QB0, %IB26, %MW4. Подробнее см. раздел «Конфигуратор ПЛК (PLC Configuration)» документа «Руководство пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [2]).

Некоторые элементы конфигурации пользователь настраивает самостоятельно. Настройка может заключаться в добавлении и/или удалении модулей и подмодулей, а также в задании требуемых значений параметрам элементов конфигурации.

Внимание!
1) Добавление и удаление модулей конфигурации, а также настройка их параметров осуществляются при контроллере, отключенном от ПО CoDeSys. Для отключения контроллера следует вызвать команду «Онлайн | Отключение (Online | Logout)» главного меню или нажать кнопку «Отключение (Logout)» ( ) панели инструментов.
2) При конфигурировании ПЛК следует иметь в виду, что можно изменять значения только переменных, лежащих в области вывода. Значения переменных из области ввода можно только считывать.

Если в процессе создания программы требуется изменить используемый ПЛК (сменить настройки целевой платформы), то следует:

1) В окне «Настройки целевой платформы» (вкладка «Ресурсы» Организатора объектов) открыть настройки целевой платформы, и выбрать новый Target-файл (соответствующий новому ПЛК).
2) Перейти в режим «Конфигурация ПЛК» и выбрать команду Дополнения | Стандартная конфигурация (Extras | Standard Configuaration)» главного меню.

При этом, если предполагается переход от одного типа контроллера к другому, то переменные следует задавать в режиме («ресурсе») «Глобальные переменные (Global Variables)». Связано это с тем, что при задании стандартной конфигурации («Standard Configuaration») переменные, заданные в редакторе «Конфигурация ПЛК (PLC Configuration)», пропадают, и ранее созданное распределение и именование переменных теряется. При объявлении глобальных переменных их имена не будут потеряны, и при переходе к другому Target-файлу достаточно только скорректировать адреса.

Внимание!
Все переменные, привязанные к каналам конфигурации ПЛК, автоматически объявляются глобальными переменными.

Для объявления глобальной переменной следует:

1) Войти в режим «Глобальные переменные (Global Variables)», в открывшемся окне режима (редакторе) – выбрать команду «Автообъявление» контекстного меню (см. рисунок 7.3).
2) В открывшемся окне «Автообъявление переменной» (см. рисунок 7.4) – задать значения параметров вводимой переменной.

Подробнее об объявлении и применении глобальных переменных см раздел 6.2 «Глобальные и конфигурационные переменные, файл комментариев» документа «Руководство пользователя по программированию ПЛК в CoDeSys 2.3» (см. приложение Н, [2]).

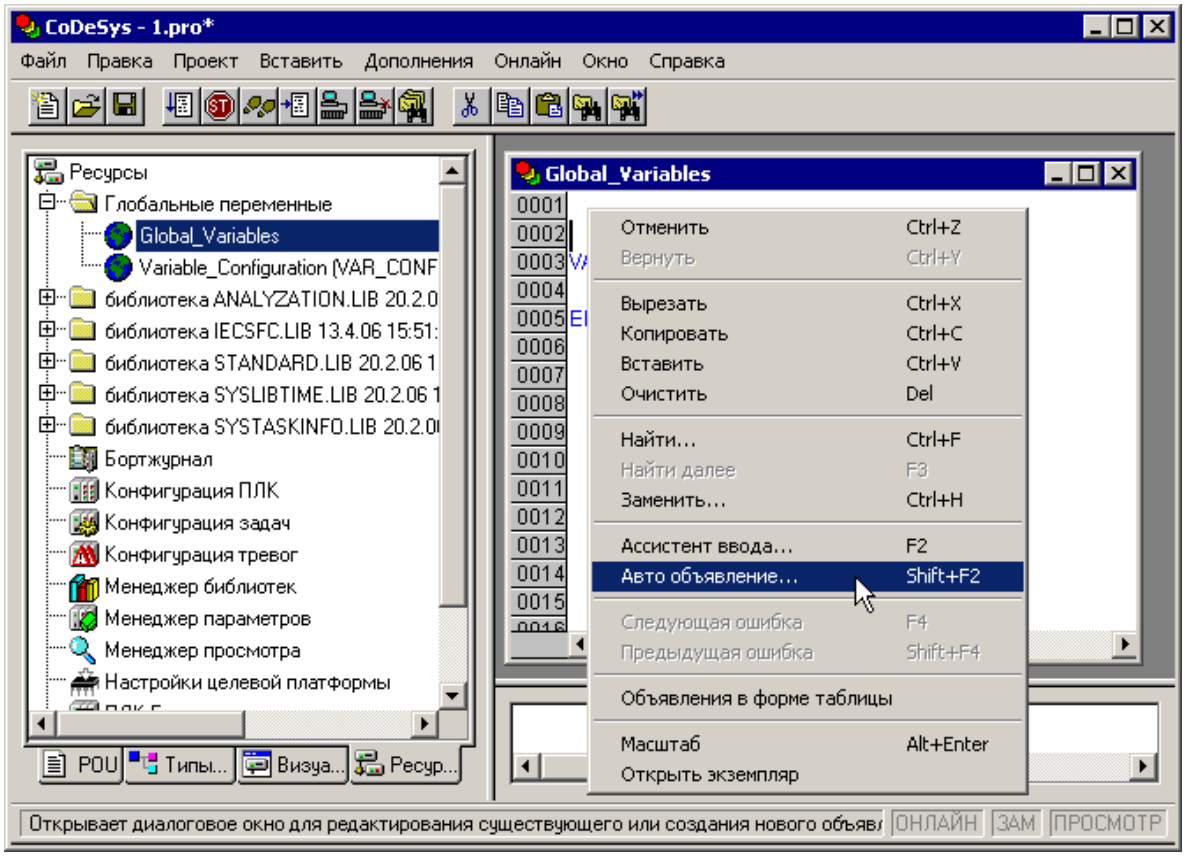


Рисунок 7.3 – Глобальные переменные. Вызов окна «Объявление переменной»

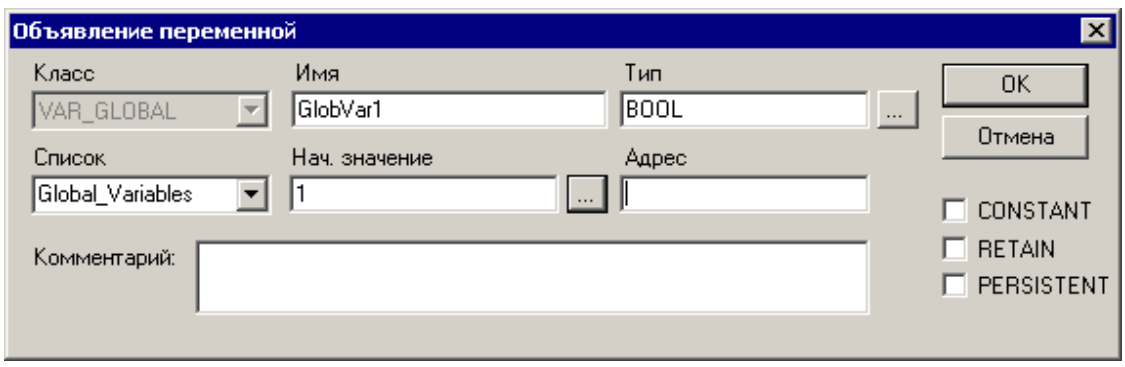


Рисунок 7.4 – Окно «Объявление переменной»

7.1.1 Приемы редактирования конфигурации ПЛК

Начальный вид конфигурации ПЛК задает файл конфигурации (*.cfg) ПЛК, расположенный в директории, определенной в установленном целевом файле (Target-файле) и считываемый при открытии проекта в ПО CoDeSys.

Редактирование элементов конфигурации ПЛК заключается в выполнении операций над элементами «дерева» конфигурации ПЛК, отображаемого в левой части окна режима (добавлении, замене и удалении модулей, подмодулей и каналов) и редактировании значений параметров элементов «дерева» конфигурации ПЛК (в правой части окна «Конфигурация ПЛК»).

7.1.1.1 Типы и виды модулей в конфигурации

Существует два вида модулей:

- **Фиксированные модули** – предустановленные конфигурационные и оперативные параметры ПЛК, которые не могут быть удалены или заменены. Допускается только редактирование их значений. К фиксированным модулям относятся параметры аппаратных модулей: дискретные и аналоговые входы, выходные устройства, настройки RS-485, параметры версии программ, прошивок. Параметры конфигурации ПЛК доступны для просмотра и/или редактирования с ЖКИ ПЛК. Структура меню ЖКИ полностью идентична структуре дерева Конфигурация ПЛК. Подробнее о фиксированных параметрах конфигурации ПЛК см. п. **3**. Список параметров приведен в Приложении Б.
- **Добавляемые модули** – добавляются (заменяются, удаляются) пользователем в процессе конфигурирования ПЛК. Пользовательские параметры можно объединить в ветви, ветви включить в другие ветви, тем самым, задавая вложенность параметров. Количество добавленных пользовательских параметров ограничено объемом доступной памяти (типом используемой лицензии, см. п. 2.2). Подробно о добавляемых пользовательских параметрах см. п. 4 .

  Подразделяются на два типа:

  - **Тип «SLOT»** – означает, что для модуля зарезервировано место, которое может быть занято или оставлено пустым. На одно зарезервированное место может быть установлен один модуль.
  - **Тип «VAR»** (свободный) – означает возможность установить любое количество модулей (с учетом физических возможностей области ввода / вывода).

    Внимание. Добавление и удаление модулей, а также настройка их параметров запишутся в ПЛК только после вызова команды меню **Online | Login** или воспользоваться соответствующей кнопкой на панели инструментов в среде программирования контроллера. Если контроллер уже находится в режиме связи с ПО CoDeSys (Online), то для записи параметров необходимо сначала разорвать связь (командой Онлайн | Отключение (Online | Logout)) и потом заново войти в режим связи (Онлайн | Подключение (Online | Login)).

7.1.1.2 Добавление подмодулей (подэлементов)

К модулям конфигурации могут быть добавлены подмодули («подэлементы»), которые расширяют функционал или изменяют алгоритм работы модуля.

Чтобы **добавить** подмодуль (подэлемент) в текущую конфигурацию, следует:

1) Либо: Выделить требуемый модуль (элемент) конфигурации и нажатием правой кнопки мыши вызвать контекстное меню.

Выбрать в контекстном меню требуемую команду: «Добавить Подэлемент (Append Subelement) | <Имя Подэлемента>».
Выбранный подэлемент будет добавлен в редактируемую конфигурацию.

2) Либо: Выделить требуемый элемент (модуль) конфигурации и нажатием левой кнопки мыши выбрать команду «Вставка | Добавить Подэлемент (Append Subelement) | <Имя Подэлемента>» главного меню.
Выбранный подэлемент будет добавлен в редактируемую конфигурацию.

7.1.1.3 Замена модулей (элементов)

Чтобы **заменить** модуль (элемент) в текущей конфигурации, следует:

1) Либо: Выделить требуемый модуль (элемент) конфигурации и нажатием правой кнопки мыши вызвать контекстное меню.
Выбрать в контекстном меню требуемую команду: «Заменить Элемент | <Имя элемента>».
Выбранный элемент заместит собою выделенный модуль (элемент) редактируемой конфигурации.

2) Либо: Выделить требуемый модуль (элемент) конфигурации и нажатием левой кнопки мыши выбрать команду «Дополнения | Заменить элемент | <Имя элемента>» главного меню.
Выбранный элемент заместит собою выделенный модуль (элемент) редактируемой конфигурации.

Процедуры подключения к модулю подчиненного подэлемента и замены элемента продемонстрированы на рисунках 7.5 (а) и 7.5 (б).

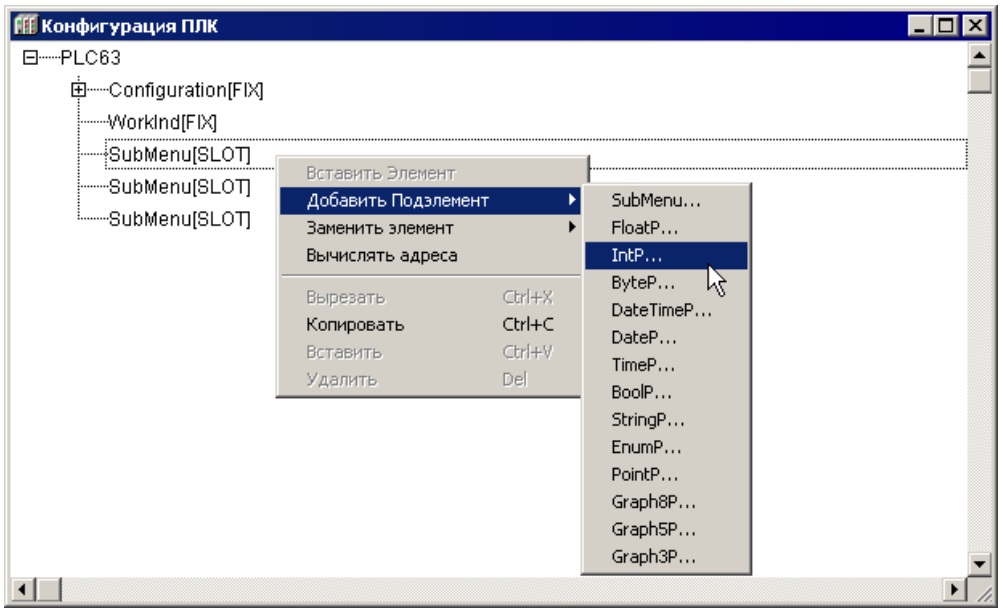


Рисунок 7.5, а – Подключение подчиненного подмодуля

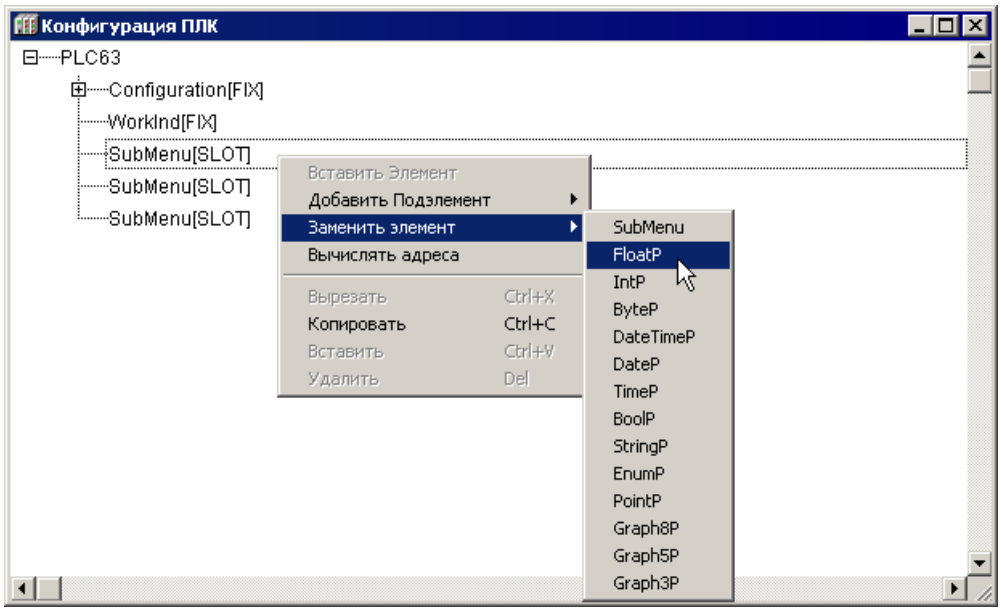


Рисунок 7.5, б – Замена подчиненного подмодуля

7.1.1.4 Удаление подмодулей (подэлементов)

Чтобы **удалить модуль** (элемент) из текущей конфигурации (удалены могут быть только добавляемые модули, фиксированные модули не могут быть удалены из конфигурации), следует:

1) Выделить требуемый модуль в дереве конфигурации и выбрать команду **Удалить (Delete)** контекстного меню дерева конфигурации.
2) В открывшемся окне запроса подтверждения операции – нажать кнопку «Да» для подтверждения операции (или кнопку «Нет» для отказа от завершения операции удаления). Выделенный модуль будет удален из дерева конфигурации.

7.1.1.5 Параметры модулей

Параметры текущего (выделенного в дереве конфигурации) модуля отображаются на вкладках в правой части окна режима «Конфигурация ПЛК (PLC Configuration)».

7.1.1.5.1 Вкладка «Базовые параметры» модуля

В полях вкладки «Базовые параметры (Base parameters)» отображаются значения параметров:

- **«Идент(ификатор) модуля (Modul id)»** – идентификационный номер модуля.
- **«Идент(ификатор) узла (Node id)»** – определяет положение модуля на его уровне иерархии в общей конфигурации –. Это значение можно редактировать, в таком случае аналогичные идентификаторы других модулей одного уровня иерархии будут сдвигаться;
- **«Адрес входов (Input Addres), Адрес выходов (Output Addres), Адрес диагностики (Diagnostic Addres)** – отображаются адреса областей ввода-вывода (приводятся конкретные номера). Они могут использоваться для обращения при программировании; значения недоступны для редактирования.
- **«Комментарий (Comment)»** – произвольный текст комментария.

На рисунке 7.6, на примере модуля дискретных входов, представлено окно режима «**Конфигурация ПЛК (PLC Configuration)**» с вкладкой базовых параметров модуля в правой части экранной формы.

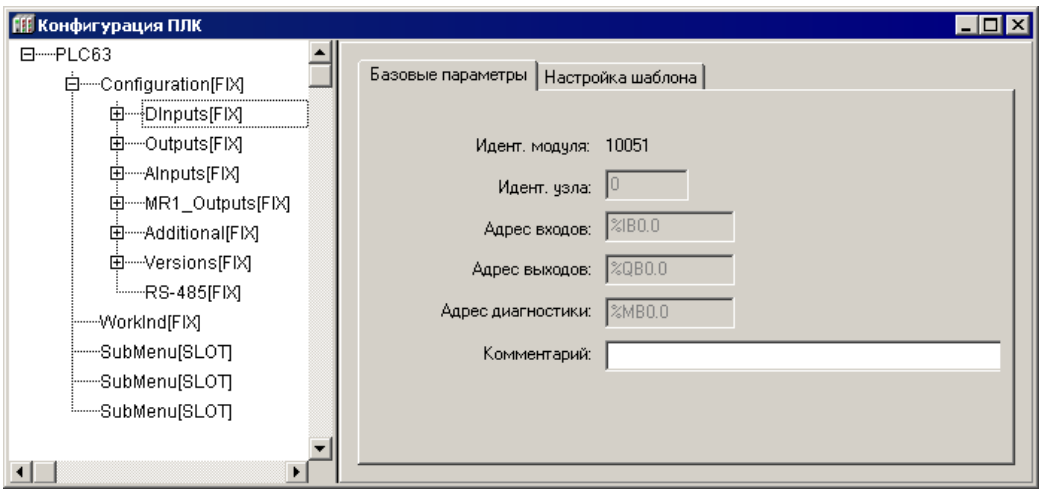


Рисунок 7.6 – Окно режима «Конфигурация ПЛК (PLC Configuration)». Модуль дискретных входов. Вкладка «Базовые параметры»

7.1.1.5.2 Вкладка «Настройка шаблона»

На вкладке «Настройка шаблона» (см. рисунок 7.7) отображаются значения параметров текущего (выделенного в дереве конфигурации) элемента или подэлемента конфигурации, представленные в виде диалогового окна, содержащего поля и группы полей. Каждое поле соответствует параметру редактируемого элемента конфигурации. Те поля, значения которых для текущего элемента конфигурации не могут быть отредактированы, отображаются в режиме просмотра (недоступны для редактирования).

Типовые поля окна:

- «Название» – название параметра, которое будет отображаться на ЖКИ. Максимальное возможное количество отображаемых символов (включая точки) равно 15.
- «Тип» – тип параметра: конфигурационный или оперативный. Значение оперативного параметра сохраняется только в ОЗУ и при выключении питания пропадает. Значение конфигурационного параметра записывается в энергонезависимую EEPROM память и при выключении питания сохраняется.
- «Значение» – группа полей, включающая поля:
  - «По умолчанию» – значение конфигурационного параметра, которое будет отображаться на ЖКИ после загрузки проекта. В дальнейшем значение может быть изменено с передней панели или из программы конфигуратора.

    Внимание! При повторном подключении контроллера к среде программирования и повторной загрузке проекта введенное с передней панели значение конфигурационного параметра будет заменено на значение по умолчанию.
  - «Минимальное» – минимальное значение параметра;
  - «Максимальное» – максимальное значение параметра.

    Примечание: Если поля «Минимальное» и «Максимальное» пусты (значения не заданы), то границы изменения значения параметра будут соответствовать границам типа параметра. Если установлена только одна из границ, то границы изменения значения параметра от «0» до установленного

максимального значения, либо от установленного минимального значения до «0».

- «Передняя панель» – группа полей, включающая поля:
  - «Показывать в меню» – если флажок переключателя установлен, то параметр будет отображаться в Главном меню ПЛК на ЖКИ; если не установлен – не будет.
  - «Представление» – представление значения параметра на ЖКИ: «десятичное, двоичное, шестнадцатеричное, с ошибкой измерителя». Тип представления «С ошибкой измерителя» устанавливается только для аналоговых вводов для расшифровки возможных ошибок измерителей.
  - «Знаков после запятой» – количество знаков в после запятой в значении параметра, которое будет отображаться на ЖКИ;
  - «Разрешить изменение с передней панели» – если флажок переключателя установлен, то параметр будет доступен для изменения с передней панели в Главном меню ПЛК; если не установлен – не будет.
  - «Защитить паролем» – если флажок переключателя установлен, то параметр доступен для изменения с передней панели в Главном меню ПЛК, при условии ввода пароля, заданного в поле справа от переключателя.
- «Сеть» – группа полей, включающая поля:
  - «Протокол ОВЕН» – если флажок переключателя установлен, то к параметру можно будет обратиться по протоколу ОВЕН по заданным (см. ниже) имени и индексу.
  - «Имя» – уникальное имя параметра для доступа к нему по протоколу ОВЕН.
    Для параметров конфигурации ПЛК имя параметра предустановленно и недоступно для изменения.
    Для пользовательских параметров для ввода имени следует установить флажок переключателя «Протокол ОВЕН» и в поле ввести имя параметра; имя может содержать до четырех символов, после каждого из которых может стоять либо не стоять точка.
  - «Индекс» – если флажок переключателя установлен, то параметр имеет индекс. В поле справа от переключателя указывается значение индекса. **Внимание!** Оперативные параметры индексируются в адресе, а конфигурационные – в индексе.
  - «Hash-код» – hash-код для протокола ОВЕН, рассчитывается автоматически в зависимости от введенного имени параметра для протокола ОВЕН. **Внимание!** Пользователь должен проследить, чтобы все параметры различались hash-кодом или, при одинаковом hash-коде, индексом (при его наличии). Следует отметить, что у двух параметров, имеющих разные имена, может совпадать hash код.

  - «Протокол Modbus» – если флажок переключателя установлен, то к параметру можно будет обратиться по протоколу Modbus.
  - «Регистр» – номер регистра для доступа к параметру по протоколу Modbus.
    Для параметров конфигурации номер регистра жестко предустановлен и не доступен для изменения.
    Для пользовательских параметров (добавляемые модули) задание номера регистра будет доступно после установки флажка переключателя. Справа от поля указан диапазон значений, из которого необходимо задать значение номера регистра. Для 4-х байтных параметров (параметров типа stre-

ang, floatP, TimeP, DataP, DataTimeP, PointP, а также для координат точек для модулей Graph8P, Graph5P, Graph3P) – диапазон для ввода номера регистра – от 82 до 286; следует учитывать, что параметр занимает два регистра. Для 2-х байтных параметров (IntP, Bool, Byte, Enum) – диапазон ввода значения регистров – от 332 до 511.

Примечание: Вся память ввода/вывода (%I и %Q) доступна для прямого последовательного считывания по протоколу Modbus функциями 03, 04, 05, 10, f и, начиная с версии 2.05 прошивки, – также функциями 01, 02, начиная с адреса регистра 4096 (подробнее см. п. 4).

– «Разрешить изменение значения по сети» – если флажок переключателя установлен, то разрешен доступ к параметру по сети на чтение и запись.

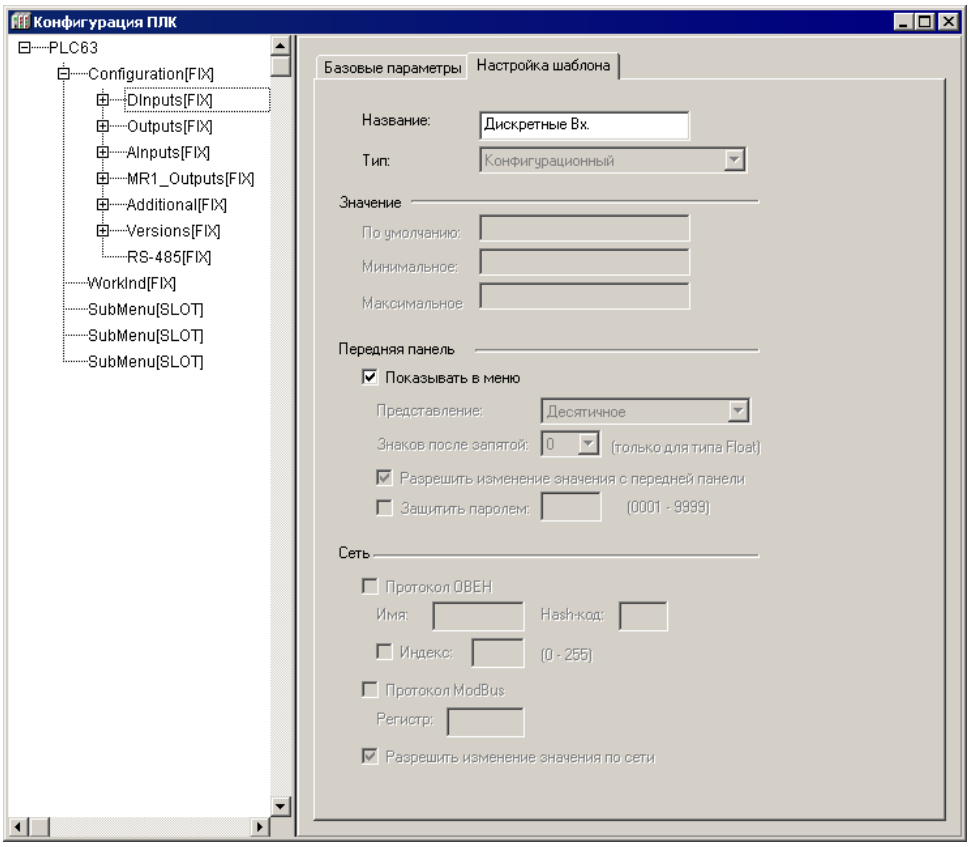


Рисунок 7.7 – Окно режима «Конфигурация ПЛК (PLC Configuration)». Модуль дискретных входов. Вкладка «Настройка шаблона»

7.1.1.5.3 Вкладка «Параметры модуля»

На вкладке «Параметры модуля (Module parameters)» отображаются значения параметров, представленные в виде таблицы, содержащей столбцы:

– индекс (**Index**),
– имя (**Name**),
– значение (текущее) (**Value**),
– (значение) по умолчанию (**Default**),
– мин(имальная) (**Min**) величина диапазона возможных значений,
– макс(имальная) (**Max**) величина диапазона возможных значений.

Примечание. Значения параметров по умолчанию, минимальные и максимальные значения – опциональные и не всегда присутствуют во вкладках параметров модулей.

Для редактирования цифровых или символьных значений параметров следует щелкнуть левой кнопкой мыши на требуемом значении, после чего запись переключается в режим редактирования, и ввести требуемое значение параметра с клавиатуры.

Для редактирования значений параметров, которые могут принимать определенное значение из списка значений, следует щелкнуть левой кнопкой мыши кнопку с треугольной стрелкой, отображаемую рядом со значением параметра. Нажатие этой кнопки раскрывает список допустимых значений параметра, в котором следует щелкнуть левой кнопкой мыши на требуемом значении. Оно будет подставлено в перечень параметров. После щелчка левой кнопкой мыши в любой другой области окна выбранное значение сохраняется в списке (см. рисунок 7.8).

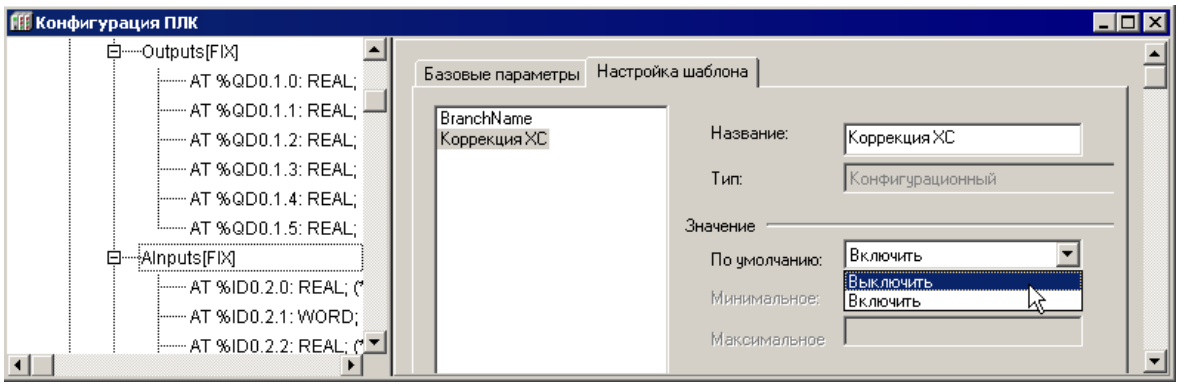


Рисунок 7.8 – Окно режима «Конфигурация ПЛК (PLC Configuration)». Модуль аналоговых входов. Вкладка «Настройка шаблона»

7.1.1.6 Каналы модуля

В состав модуля входят **каналы** (битовые, байтовые, каналы для данных типа REAL или STRING).

Каждый канал – это переносчик единиц информации (данных) от внешнего оборудования в область памяти ввода/вывода ПЛК: через канал передается значение входов / выходов (физических или сетевых), также в канале указывается, в каком месте памяти области ввода / вывода хранится данное значение (каждому каналу соответствует переменная в области ввода / вывода).

Канал и соответствующая ему переменная могут быть поименованы, и по присвоенному имени к переменной можно обращаться в программе, либо в программе возможен вызов переменной канала по тому адресу, который ей установлен аппаратно: например, «%IX 0.0.1».

Восьмибитовый канал может быть использован в программе либо как 8-битное число, либо как 4 или 8 отдельных друг от друга битов.

При именовании переменных следует соблюдать правила именования:

1) Имя может состоять из латинских букв, цифр и знака «_» (подчеркивание).
2) Имя должно начинаться с буквы или знака «_».
3) Имя должно быть уникальным.
4) В некоторых случаях редактирование имен каналов может быть запрещено.

Окно, представленное на рисунке 7.9, иллюстрируют процесс именования канала – отображение поля ввода символов.

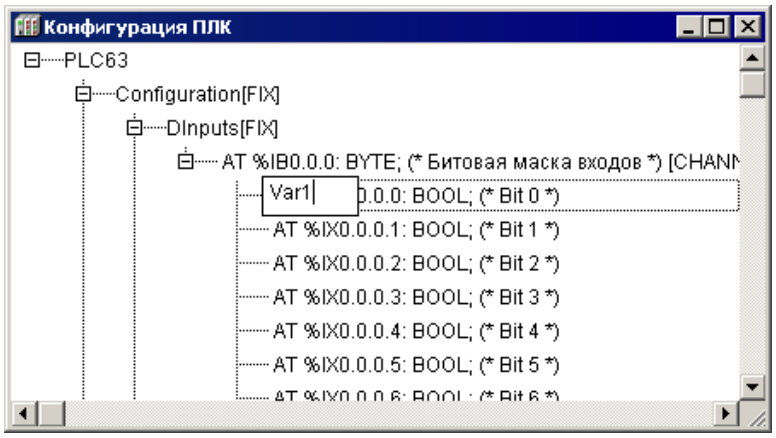


Рисунок 7.9 – Ввод и редактирование имени переменной канала

Данные, отображаемые в полях вкладки «Базовые параметры», носят информационный характер и (за исключением текста комментария) не редактируются. Для байтового канала отображаются следующие данные:

- **комментарий** – характеристика канала (например, для модуля дискретных входов – «8 discrete inputs» = «8 дискретных входов»);
- **ID канала** – идентификационный номер канала в общем списке;
- **размер** – в битах

Для битового канала программа выводит только комментарий с номером битового канала, например, «Bit 3».

Значения в каналы модулей ветви Configuration конфигурации ПЛК и пользовательских параметров поступает из источников / приемников, перечисленных в таблице 7.1

Таблица 7.1 – Источники / приемники значения в канале

Тип модуля	Источник / приемник значения в канале	Примечание
Модули ветви «Configuration» конфигурации ПЛК, оперативные параметры	1) драйверы портов/ввода вывода; 2) Значение по сети RS; 3) Значение из окна «Конфигурация ПЛК (PLC Configuration)» 4) Значение с передней панели контроллера.	
Модули ветви Configuration конфигурации ПЛК, конфигурационные параметры	1) Считываются из Flash-памяти при запуске контроллера (сохраняются во Flash-памяти, поэтому команда «Сброс EEPROM» из отладочного меню на них не действует); 2) Значение по сети RS; 3) Значение с передней панели.	Относятся к ветви Additional. Измененное значение из окна «Конфигурация ПЛК (PLC Configuration)» не сохраняется в EEPROM и недоступно по сети RS. (введенное значение канала доступно только из программы).
Модули ветви пользовательских параметров, оперативные параметры	1) Значение по сети RS; 2) Значение из окна PLC_Configuration 3) Значение с передней панели контроллера.	
Модули ветви пользовательских параметров, конфигурационные параметры	1) Считываются из EEPROM при запуске контроллера; 2) Значение по сети RS; 3) Значение с передней панели.	Измененное значение из окна «Конфигурация ПЛК (PLC Configuration)» не сохраняется в EEPROM и не доступно по сети RS. (введенное значение канала доступно только из программы).

7.2 Задание времени цикла ПЛК

Для изменения параметров времени цикла ПЛК, следует:

1) В дереве конфигурации выделить корневой элемент («PLC63», см. рисунок 7.9).
2) В области задания параметров – перейти на вкладку «Параметры модуля (Module Parameters)», щелчком левой кнопки мыши в требуемой ячейке списка параметров, переведя запись значения, установленного по умолчанию, в режим редактирования (см. рисунок 7.9).

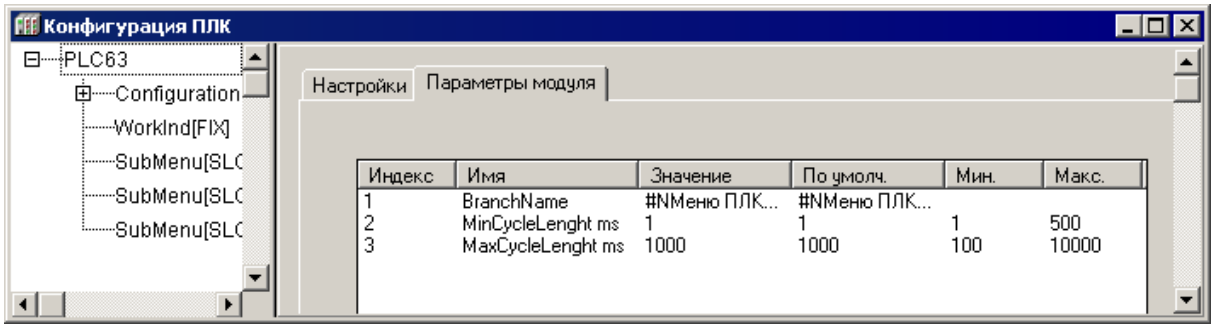


Рисунок 7.10 – Окно настройки параметров работы ПЛК

3) Задать требуемые значения параметров времени цикла ПЛК:

– **MinCycleLength, ms (Минимальное значение цикла работы ПЛК, в мс) –** параметр определяет минимальный период, с которым ПЛК выполняет полный цикл своей работы. Диапазон значений от 1 до 500 мс, значение по умолчанию – 1 мс.
Программная реализация ПЛК обеспечивает вызов цикла ПЛК не чаще, чем 1 раз в установленное число миллисекунд. Поэтому, задавая значение параметра**,** следует учесть, что после выполнения цикла ПЛК (т.е. после выполнения операции ввода данных, выполнения пользовательской программы и вывода данных) выполняется еще ряд сервисных функций (обеспечивающих сетевой обмен, работу с файлами и т.д.), на выполнение которых также требуется процессорное время. И если пользовательская программа ПЛК выполняется за время, превышающее 70-80% от значения, заданного в параметре «MinCycleLength», то на выполнение сервисных операций контроллеру не остается времени. При этом возможны сбои, замедление или прекращение сетевого обмена с модулями ввода-вывода, сбои в записи архивов и т.д. Для исправления некорректной ситуации следует увеличить значение параметра.
Для нормальной работы рекомендуется, чтобы время простоя процессора составляло не менее 20% от значения, заданного в параметре «MinCycleLength».
Значение параметра «MinCycleLength» может быть задано равным нулю. Тогда в контроллере отключается контроль времени вызова цикла ПЛК. После выполнения предшествующего цикла и после выполнения всех сервисных функций происходит вызов следующего цикла ПЛК. При этом не гарантируется строгое выполнение цикла через равные промежутки времени, т.к. длительность выполнения сервисных функций может изменяться от цикла к циклу.
– **MaxCycleLength, ms (Максимальное значение цикла работы ПЛК, в мс)** – параметр определяет максимально допустимое время, за которое ПЛК выполняет полный цикл своей работы. Если в процессе работы ПЛК

заданная величина будет превышена (при зависании программы или при выполнении бесконечного цикла), то ПЛК будет принудительно перезагружен. Т.е., параметр «MaxCycleLength» задает время ожидания **сторожевого таймера** («WatchDog Timer»).
Диапазон значений от 100 до 10000 мс, значение по умолчанию – 1000 мс.

7.3 Фиксированные элементы (модули) конфигурации

К фиксированным модулям относятся аппаратные входы (дискретные и аналоговые), универсальные выходы, модуль интерфейса и дополнительные периферийные устройства ПЛК. В фиксированных модулях задаются параметры конфигурации ПЛК.

7.3.1 DInputs (модуль дискретных входов)

Модуль дискретных входов («DInputs») отображает значения, характеризующие состояния дискретных входов ПЛК. Модуль имеет один восьмибитовый канал, каждый бит которого соответствует состоянию одного дискретного входа (см. рисунок 7.11).

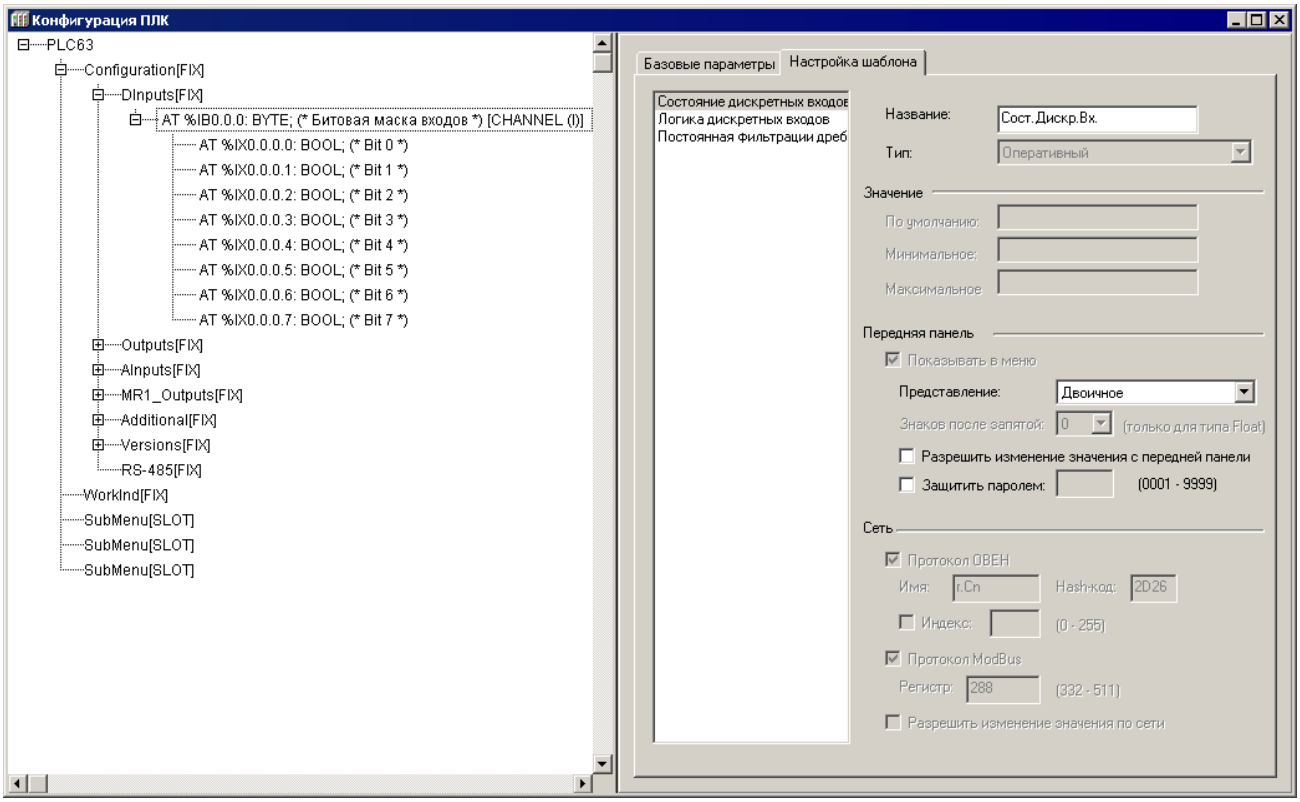


Рисунок 7.11 – Параметры канала модуля дискретных входов («DInputs»)

Параметры канала:

- «Состояние дискретных входов» («Сост.Дискр.Вх.»)– отображает текущее состояние дискретных входов ПЛК. Оперативный параметр. На ЖКИ состояние дискретных входов отображается в виде битовой маски (см. рисунок 7.12). Соответствие разрядов ЖКИ дискретным входам отображено в таблице 7.2.
- «Логика дискретных входов» («Логика Дискр.Вх») – диапазон значений от 0 до 255, значение по умолчанию 0. Параметр задает логику инвертирования дискретных сигналов. По умолчанию в параметре задано значение «0», соответствующее в битовом формате «00000000» (см. рисунок 7.13, а). Для изменения логики необходимо выставить «1» в разряде, соответствующему номеру дискретного входа, где замыкание сухого контакта должно восприни-

маться прибором, как неактивное состояние (= False). В каждом разряде значение может быть выставлено или «0» или «1». Например, если необходимо изменить логику инвертирования первого, второго и шестого входов, то в параметре «Логика дискретных входов» надо задать значение «11000100».
Внимание! Для того чтобы изменения, внесенные с передней панели, в параметрах «Логика Дискр.Вх» («Логика дискретных входов») и «Пост.Ф.ДребКонт» («Постоянная фильтрации дребезга», см. ниже), вступили в действие, необходимо выключить и снова включить питание прибора.

– «Постоянная фильтрации дребезга» – задает период опроса состояния одного дискретного входа, задается в единицах миллисекунд (1 ед. = 1 мс); диапазон значений от 0 до 255, значение по умолчанию 10 (см. рисунок 7.13, б).

Фильтрация применяется для подавления «дребезга» контактов. Принцип действия фильтрации:

- в сдвиговом регистре в драйвере каждого дискретного входа накапливаются значения восьми последних состояний, полученных в результате опроса с периодом, заданным в параметре «Постоянная фильтрации дребезга»;
- если состояние битового канала дискретного входа равно **1 (TRUE)**, а количество единиц в сдвиговом регистре менее двух, то битовый канал переключается на **0 (FALSE);**
- если состояние битового канала равно **0 (FALSE)**, а количество единиц в сдвиговом регистре больше пяти, то битовый канал переключается на **1 (TRUE);**
- если количество единиц в сдвиговом регистре от 2 до 5, то состояние битового канала дискретного входа не меняется.

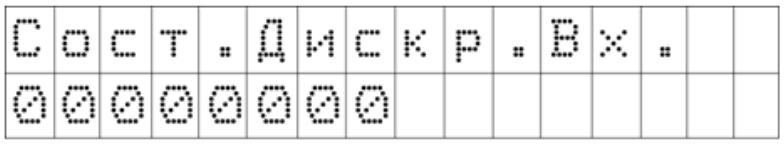
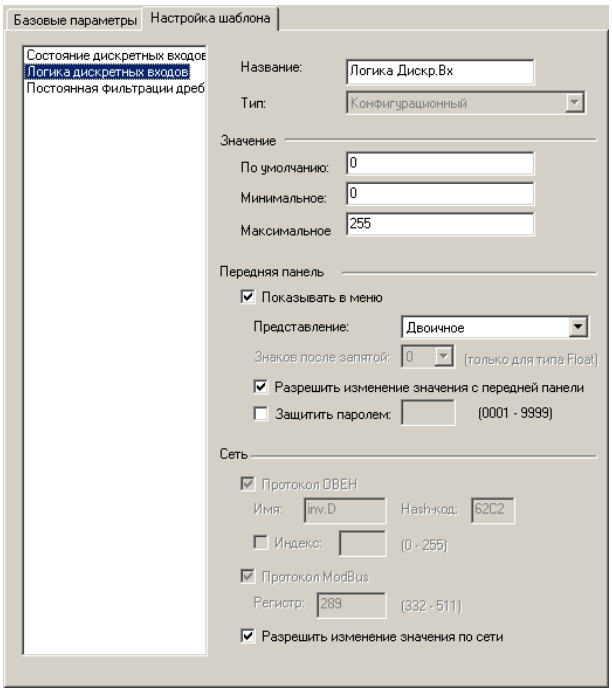


Рисунок 7.12 – Отображение состояние дискретных входов на ЖКИ

Таблица 7.2 – Соответствие разрядов ЖКИ дискретным входам (X принимает значение «0» или «1», в зависимости от того, замкнут или разомкнут контакт датчика, подключенного к дискретному входу)

Дискретный вход	C1	C2	C3	C4	C5	C6	C7	C8
Соответствующий разряд на ЖКИ	X	X	X	X	X	X	X	X

а)



б)

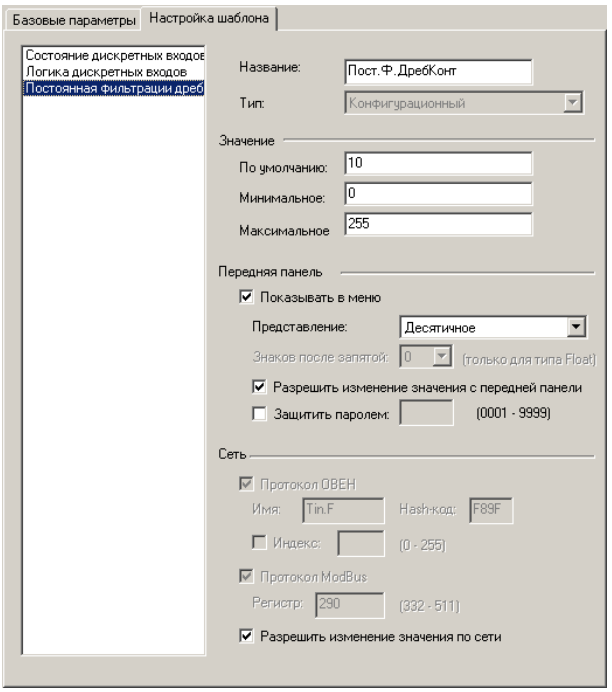


Рисунок 7.13 – Настройка параметров «Логика дискретных входов» (а) и «Постоянная фильтрации дребезга контактов» (б)

7.3.2 Outputs (модуль выходных устройств)

Модуль выходных устройств **(Outputs)** отображает значения выходов ПЛК. Модуль имеет шесть каналов типа Real. В каждый канал из пользовательской программы записывается состояние выхода, в диапазоне от 0,0 до 1,0.

В зависимости от исполнения ПЛК может быть оснащен следующими выходными устройствами: э/м реле (**Р**) или транзисторными ключами (**К**), симисторными ключами (**С**), выходами для управления твердотельными реле (**Т**), ЦАП «параметр-напряжение 0..10В» (**У**), ЦАП «параметр-ток 4…20мА» (**И**).

При работе с ВУ типа ЦАП значение канала N, (где N – номер выхода от 2 до 6, так как 1-ое ВУ всегда «Р»), преобразуется в пропорциональный ему ток или напряжение. Аналоговое управление осуществляется по прямо-пропорциональному закону (т.е. 4 мА или 0 В соответствует входному сигналу, равному 0, а 20 мА или 10 В соответствует входному сигналу, равному 1).

При работе с ВУ дискретного типа значение канала преобразуется в ШИМ-сигнал для которого необходимо задать период следования импульсов **«Период ШИМ ВУ{N}»**, а также минимальное значение времени включения ВУ в параметре **«Мин.имп.ШИМ ВУ{N}»**. В канал записывается значение, соответствующее коэффициенту заполнения ШИМ, что также прямо пропорционально выходной мощности.

Параметры каналов:

- «Сост.ВУ{N}» – отображает текущее состояние выходных устройств. Диапазон изменения значения от 0,0 до 1,0. Оперативный параметр, равняющийся значению, записанному в канал с номером **N**.
- «Период ШИМ ВУ{N}» – для каждого канала (**{N}** – номер канала) задает период выходного ШИМ-сигнала в миллисекундах. Диапазон значений от 2 до 65000, значение по умолчанию 1000. Уменьшение параметра позволяет осуществлять управление ИМ более точно, увеличение – позволяет исключить частые переключения ВУ с целью снижения уровня помех в сети и снижения износа силовых контактов реле вследствие слишком частых переключений.

- «Мин.имп.ШИМ ВУ{N}» – для каждого канала (**{N}** – номер канала) задает ограничение на минимальную длительность единичного импульса ШИМ-сигнала (длительность состояния включения). Параметр задается в миллисекундах. Диапазон значений от 1 до 5000, значение по умолчанию 50.
  Если [«Сост.ВУ{N}» × «Период ШИМ ВУ{N}» < «Мин.имп.ШИМ{N}»], то «Сост.ВУ{N}» = 0. То есть, если на ВУ задана такая мощность, что для её реализации методом ШИМ потребовалось бы включать ВУ на время, меньшее, чем значение параметра, то такое включение будет блокировано. Меньшее значение параметра позволяет выдавать сигналы малых уровней, большее значение – позволяет блокировать включение на короткие промежутки времени (например, для исключения импульсов на реле такой длительности, за которые оно не успеет включиться – с целью предотвращения обгорания контактов).
- «Безоп.сост.{N}», для каждого канала ({N} – номер канала) определяет состояния выходных элементов в момент, когда основная программа контроллера не выполняется. Значения – от 0,0 до 1,0, значение по умолчанию «0,0» (тип REAL).
  В момент загрузки ПЛК или при сбое в его работе, выходы прибора могут оказаться в неопределенном состоянии: выключены или включены. Однако это состояние может оказаться недопустимым при эксплуатации управляемого оборудования. Для исключения такой ситуации ПЛК переводит выходы при «зависании» или во время загрузки в состояние, заданное в параметре «Безопасное состояние» выхода.

Параметры «**Период ШИМ ВУ{N}**» и «**Мин.имп.ШИМ ВУ{N}**» на работу аналоговых ВУ (тип выхода **У** и **И**) влияния не оказывают.

Окно конфигурирования выходных устройств представлено на рисунке 7.14.

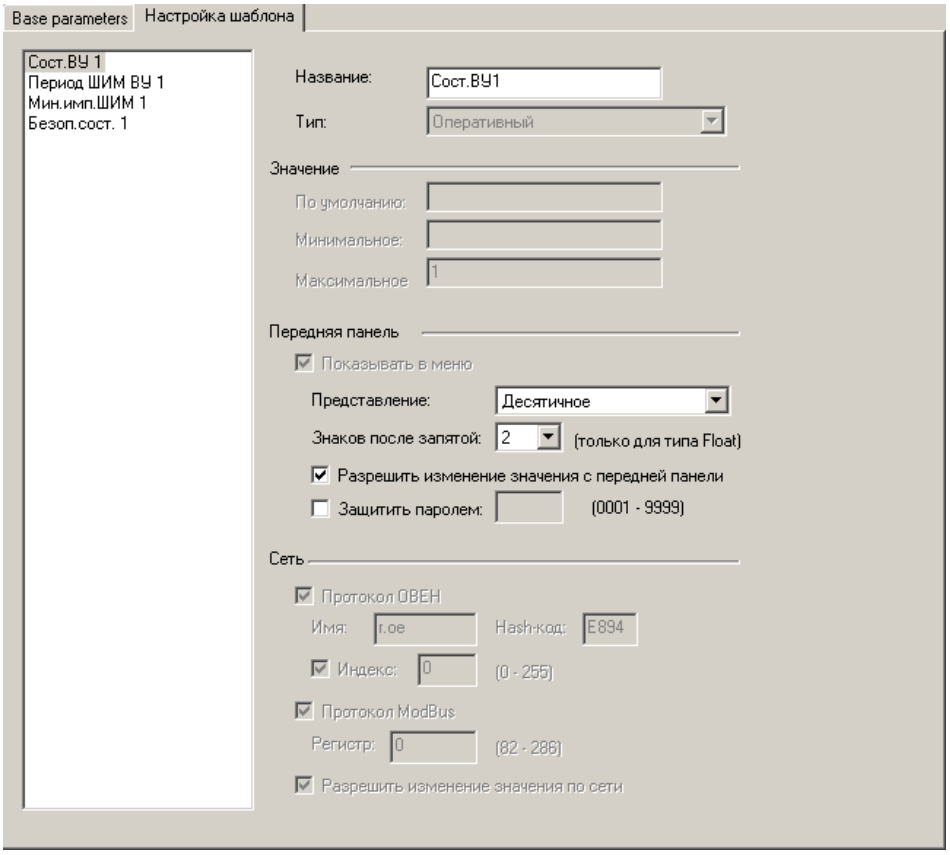


Рисунок 7.14 – Параметры канала модуля выходных устройств («Outputs»)

7.3.3 AInputs (модуль аналоговых входов)

Модуль предназначен для приведения результатов измерения к значениям физической величины, измеряемой датчиком. Приведенное значение физической величины может быть считано в программу ПЛК из каналов модуля.

Модуль имеет по два канала на каждый аналоговый вход. В первом канале содержится результат измерений в преобразованном виде, во втором канале – циклическое время измерения.

Для температурных датчиков (ТП и ТС) в первом канале выдается значение измеренной температуры.

Для резистивных датчиков выдается значение в омах, для датчиков тока и напряжения в канале выдается значение от 0 до 100 – в процентах.

Время измерения – это время, которое точно соответствует времени проведения измерения в данном канале. Оно измеряется в тактах внутреннего времени прибора (1 такт составляет 0,01 секунды). Отсчет времени измерения начинается при включении прибора и каждые 65 536 тактов (656.36 секунд) время обнуляется, отсчет начинается сначала.

Параметры модуля:

- «Branch name» – задает наименование ветви меню, отображаемого на ЖКИ. Значение по умолчанию – «Аналоговые Вх.».
- «Коррекция ХС» – параметр определяет необходимость коррекции холодного спая. В обычном режиме работы коррекцию холодного спая рекомендуется включать. Отключение коррекции холодного спая требуется для проведения метрологической поверки аналоговых входов ПЛК. Параметр распространяется на все аналоговые входы контроллера. Значения выбираются из списка «Включить» и «Выключить», значение по умолчанию – «Включить».

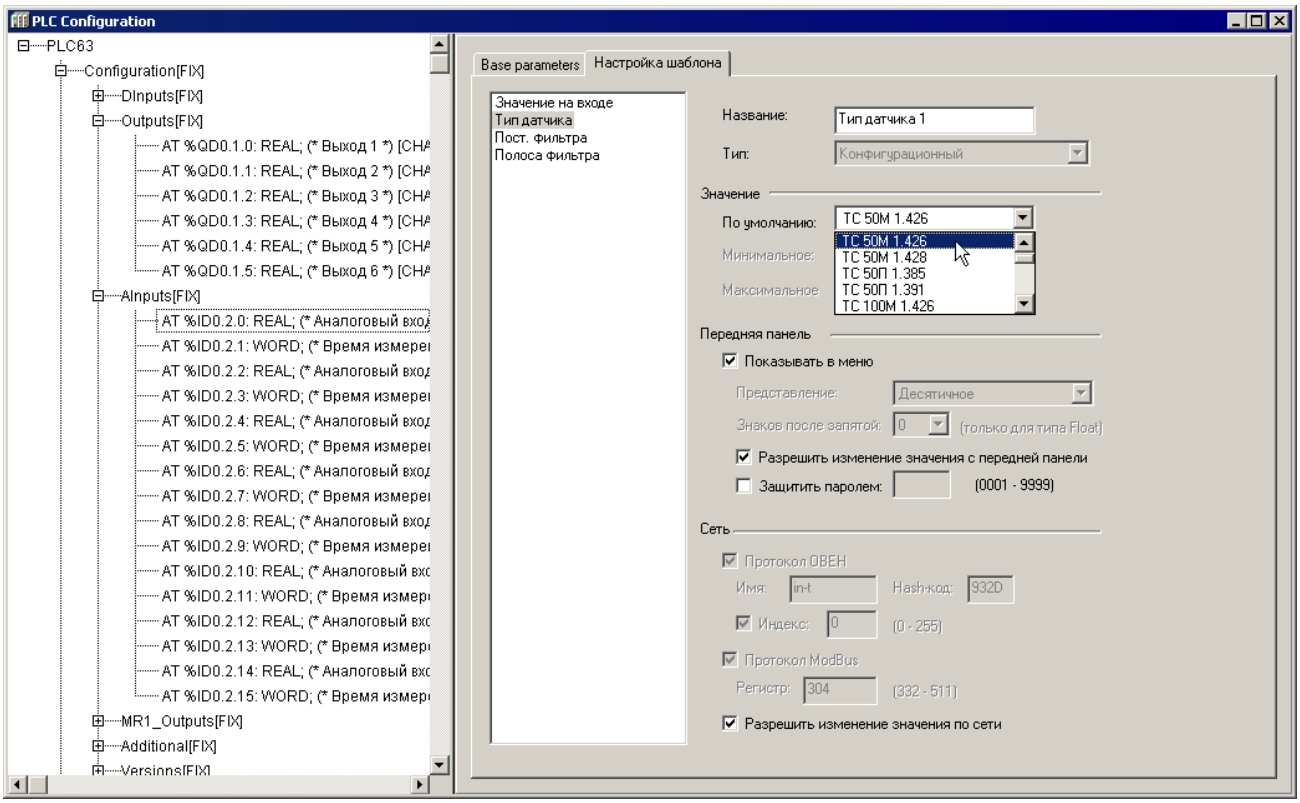


Рисунок 7.15 – Параметры канала аналогового входа (модуль «AInputs»)

Параметры каналов (Окно конфигурирования аналогового входа представлено на рисунке 7.15):

- Значение на входе {N} («Вход {N}») – показывает измеренное входом значение с учетом всех поправок. Оперативный параметр, {N} – номер канала, тип REAL.
  Примечание. При ошибке датчика (выходе за диапазон, обрыве или КЗ) значение содержит код ошибки в специальном формате. Для расшифровки сообщения об ошибки используется функциональный блок DECODE_FLOAT из библиотеки «OWEN_FB». Сообщения об ошибках в ПЛК представлены в Руководстве по эксплуатации (см. приложение Н, [5,6]).
- «Тип датчика {N}» – задается тип датчика, показания которого будут обрабатываться каналом аналогового входа. Тип датчика выбирается из раскрывающегося списка. Перечень используемых типов датчиков представлен в Приложении Г. Значение по умолчанию – ТС 50М 1.426.
- «Полоса Фильтра{N}», тип REAL – величина предельного отклонения результатов двух соседних измерений; задается индивидуально для всех датчиков в единицах измеряемых ими физических величин. Диапазон значений от 0 до 9990, значение по умолчанию – 10. Установка значения «0» отключает фильтр. Процедура фильтрации заключается в том, что из текущих измерений входных параметров отфильтровываются значения, имеющие выраженные «провалы» или «выбросы». Для этого прибор непрерывно вычисляет разность между результатами двух последних измерений входного параметра, и сравнивает ее с заданным предельным отклонением (параметром «Полоса фильтра»). Если вычисленная разность превышает заданный предел, то результат, полученный в последнем цикле опроса, считается недостоверным, дальнейшая обработка его приостанавливается и производится повторное измерение. Если недостоверный результат был вызван воздействием помехи, то повторное измерение подтвердит этот факт и ложное значение аннулируется. Это позволяет защитить прибор от воздействия единичных импульсных и коммутационных помех, возникающих на производстве при работе силового оборудования. Чем меньше заданное значение параметра, тем лучше помехозащищенность измерительного канала, но при этом (из-за возможных повторных измерений) хуже реакция прибора на быстрое фактическое изменение входного параметра. Во избежание повторных измерений при задании «Полосы фильтра» следует руководствоваться максимальной скоростью изменения контролируемого им параметра при эксплуатации, а также установленной для него периодичностью опроса.
- «Пост.Фильтра {N}» («Постоянная времени фильтра») – интервал, в течение которого изменение выходного сигнала фильтра достигает 0,63 от изменения входного сигнала. Задается в секундах для каждого входа; диапазон значений от 0 до 1800, значение по умолчанию – 5.
  Увеличение значения постоянной времени фильтра улучшает помехозащищенность канала измерения, но одновременно увеличивает его инерционность, т. е. реакция прибора на быстрые изменения входной величины замедляется. Установка значения «0» отключает фильтр.

7.3.4 MR1_Outputs (модуль выходного устройства МР1)

Фиксированный модуль расширения выходных элементов МР1 предназначен для увеличения количества выходных элементов.

Модуль имеет восьмибитовый канал (см. рисунок 7.16), каждый бит которого соответствует одному дискретному выходу модуля МР1. Единичное значение бита означает, что выход МР1 должен быть включен.

Параметр модуля: «Состояние ВУ MP1», тип Byte – значения на выходе ПЛК, определяющее состояние MP1 – оперативный параметр, равный значению канала модуля. Значение переменной автоматически записывается в MP1 с частотой 13,5 герц.

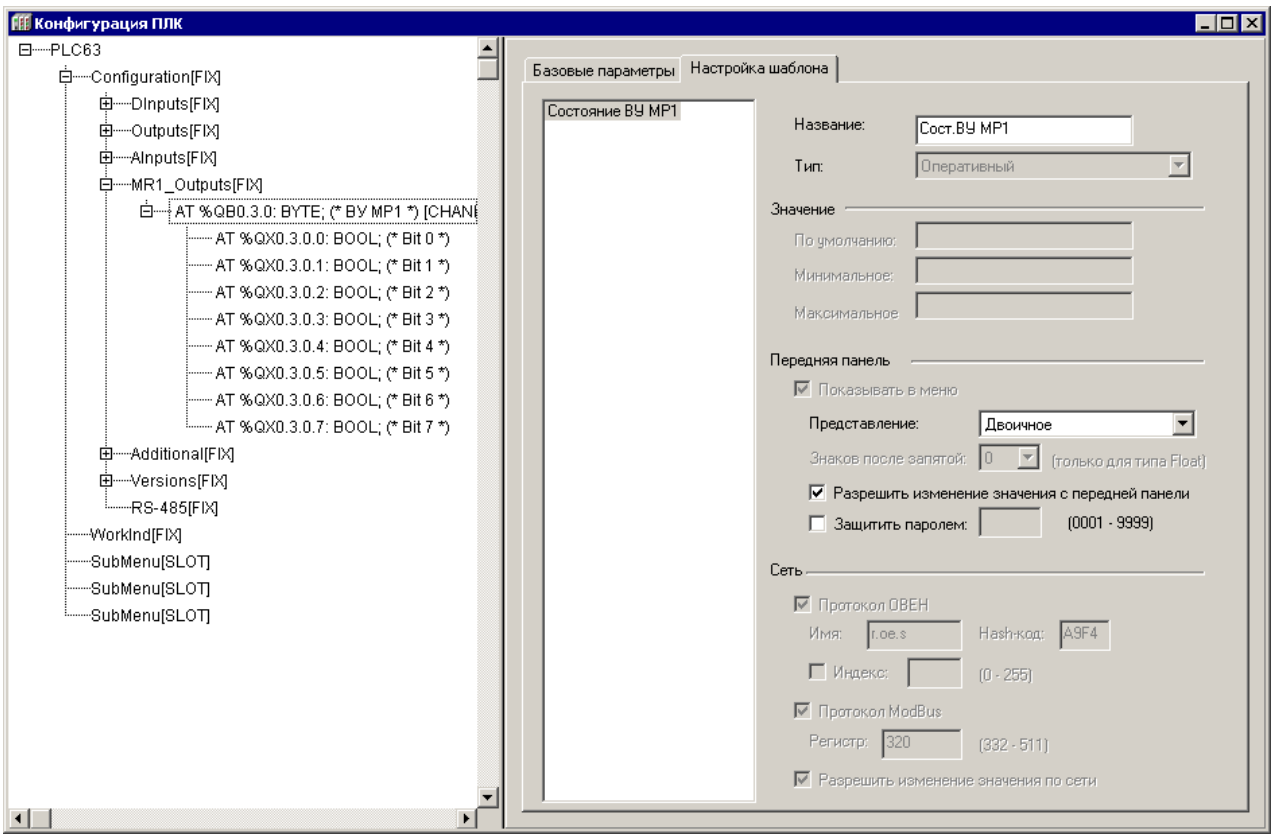


Рисунок 7.16 – Параметры канала ВУ MP1 (модуль «MR1_Outputs»)

7.3.5 Additional (модуль дополнительных параметров)

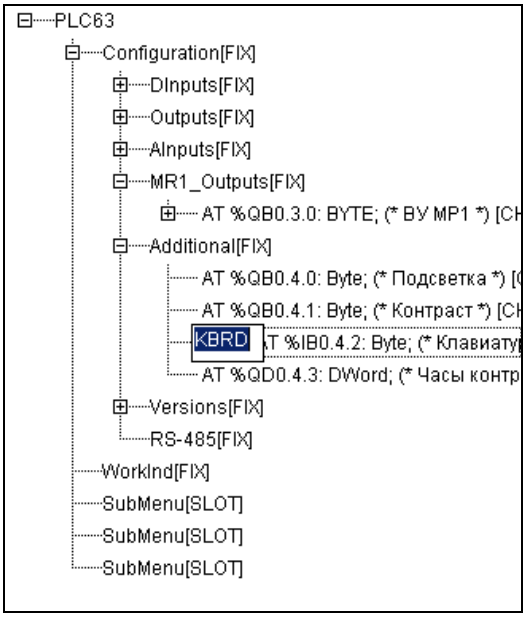
Модуль дополнительных параметров предназначен для настройки ЖКИ, клавиатуры, звука и часов реального времени на ПЛК (см. рисунок 7.17).

Параметры модуля:

- «Ярк.<ость> подсветки ЖКИ» – задает яркость подсветки ЖКИ, диапазон значений от 0 до 50, значение по умолчанию – 45.
- «Контрастность ЖКИ» – задает значение контрастности ЖКИ, диапазон значений от 5 до 32, значение по умолчанию – 25.
- «Состояние клавиатуры» – оперативный параметр, доступный только для чтения, отражающий состояние клавиатуры.
  Состояние клавиатуры – это оперативный параметр, соответствующий сумме кодов нажатых клавиш (см. ниже).
- «Звук» – задает звук, подтверждающий нажатие клавиш, значения выбираются из списка (Включить / Выключить), значение по умолчанию – «Включить». Для работы со звуковым излучателем может быть применена специализированная библиотека программных компонентов sound.lib (см. приложение Н [13])
- «Время и дата» – часы реального времени. Оперативный параметр, который устанавливается с передней панели ПЛК и сохраняется при выключении питания. При загрузке конфигурации из CoDeSys значение параметра не изменяется. Диапазон значений от 01.01.00 00:00:00 до 31.12.99 23:59:59. Работа с системными часами ПЛК63 – считывание и запись – производится с ис-

пользованием библиотеки Syslibtime (производства 3S Software). Переменную «часы контроллера» не требуется использовать в программе CoDeSys.

а)



б)

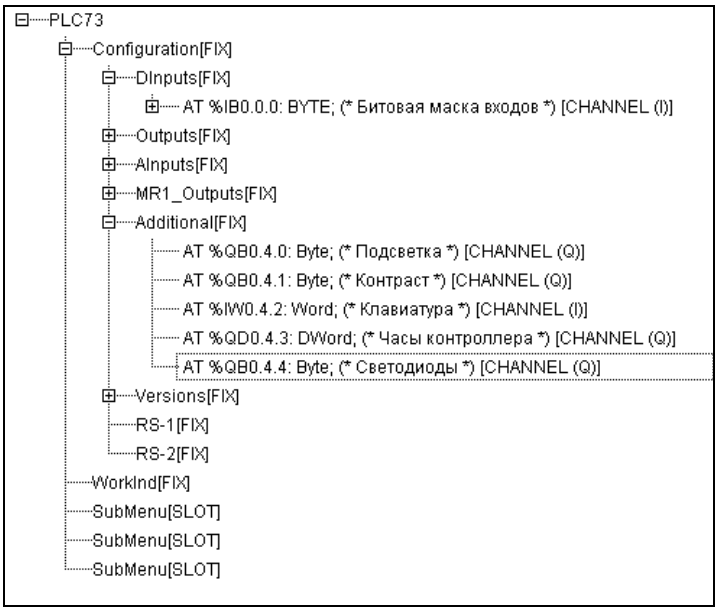


Рисунок 7.17 – Именование переменной «Состояние клавиатуры», а и переменная «Светодиоды» (только для ПЛК73), б

Настройка дополнительных функций клавиатуры производится, при необходимости, следующим образом:

1) Переменной «Состояние клавиатуры» – присвоить имя (например «KBRD», см. рисунок 7.17, а).
2) Выбрать значение оперативного параметра, соответствующего этой переменной, не зарезервированное для выполнения кнопками штатных операций (см. таблицы 8.2 – 8.5).
   Значения оперативного параметра определяются суммой кодов нажатых клавиш (см. таблицу 7.3).

Таблица 7.3 – Маски для выделения клавиш лицевой панели ПЛК63 и ПЛК73

Клавиша	Значения оперативного параметра «Состояние клавиатуры»	Клавиша	Значения оперативного параметра «Состояние клавиатуры»
ПЛК63 (8 бит)		**ПЛК73 (16 бит, тип Word)**	
	4		1
	8		2
	16		4
	32		8
	64		16
	128		32
			64
			128
			256

Зарезервированные сочетания клавиш приведены в таблицах 8.2 – 8.5.

3) Выбрать заданную переменную в программе ПЛК, связав определенные процедуры с незарезервированными значениями. В этом случае эти процедуры будут запускаться нажатием назначенного сочетания кнопок.

Примеры использования клавиатуры в программе работы ПЛК63 приведены в примерах 1 и 2, размещённых на дистрибутивном диске ПЛК (файлы keyboard_1.pro и keyboard_2.pro), см. рисунок 7.18.

PLC_PRG (PRG-ST)

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     starting:BOOL:=FALSE;
0004     run_stop:BOOL:=FALSE; (*false = stop, true = run*)
0005     fm:F_TRIG;
0006     tm: TON;
0007 END_VAR
0008 VAR CONSTANT
0009     key_exit: BYTE := 64;
0010     key_start: BYTE := 32;
0011 END_VAR
```

```
0001 IF NOT starting THEN (*что выполняется при начальной загрузке*)
0002     SetWorkScreenCount(1);
0003     ClearScreen(0);
0004     ShowString(0,0,0,'Пуск/стоп запуск');
0005     ShowString(0,0,1,'Выход останов');
0006     starting:=TRUE;
0007 END_IF
0008 fm(clk:=kbrd=key_start AND NOT run_stop ); (*если прибор в режиме стоп и пользователь нажал и отпустил клавишу пуск/стоп*)
0009 IF fm.Q THEN (*то будет выполнен этот цикл*)
0010     ClearScreen(0);
0011     ShowString(0,0,0,'Run');
0012     run_stop:=TRUE;
0013 END_IF
0014 tm(in:=kbrd=key_exit AND run_stop, pt:=t#1s); (*при нажатии клавиши выход на время более 1 сек при условии что прибор запущен*)
0015 IF tm.Q THEN (*то будет выполнен этот цикл*)
0016     ClearScreen(0);
0017     ShowString(0,0,0,'Stop');
0018     run_stop:=FALSE;
0019 END_IF
0020
```

Рисунок 7.18 – Пример использования клавиатуры в программе работы ПЛК (пример 1 , файл keyboard_1.pro)

Настройка работы светодиодов, расположенных на лицевой панели ПЛК73, производится, при необходимости, заданием значений переменной, определяющей состояние светодиодов (см. рисунок 7.19). Переменная принимает значения от о до 63, причем каждому биту из 6 бит соответствует свой светодиод: самому младшему – верхний светодиод, старшему – нижний светодиод.

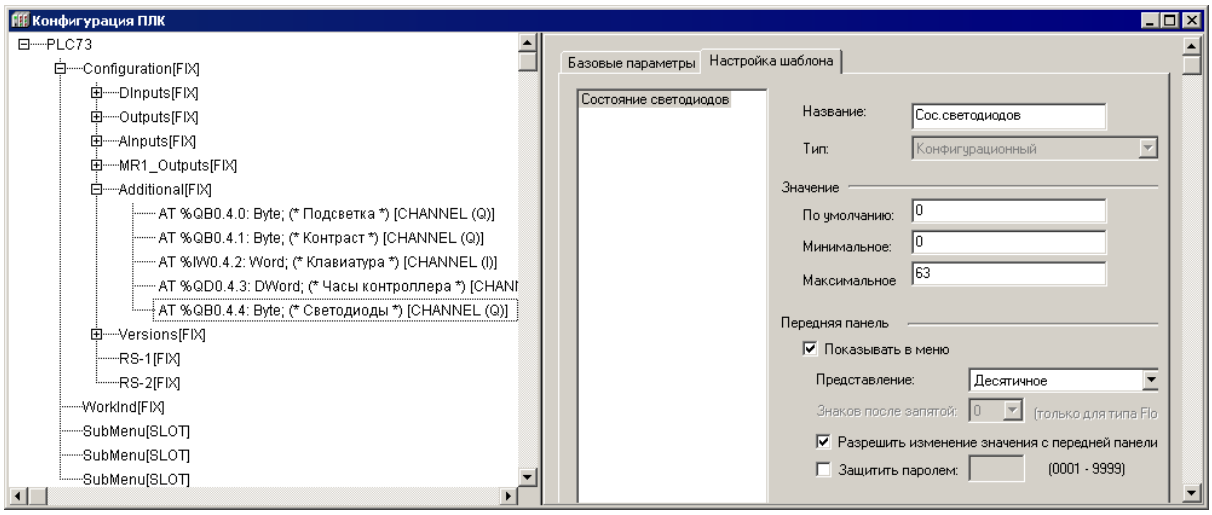


Рисунок 7.19 – Параметры конфигурирования светодиодов лицевой панели ПЛК73

7.3.6 Versions (модуль «версии»)

Модуль «Versions» (версии) предназначен для информирования пользователя о версии ПО микроконтроллера («прошивки») и о версии пользовательской программы (устанавливается в самой пользовательской программе), см. рисунок 7.18.

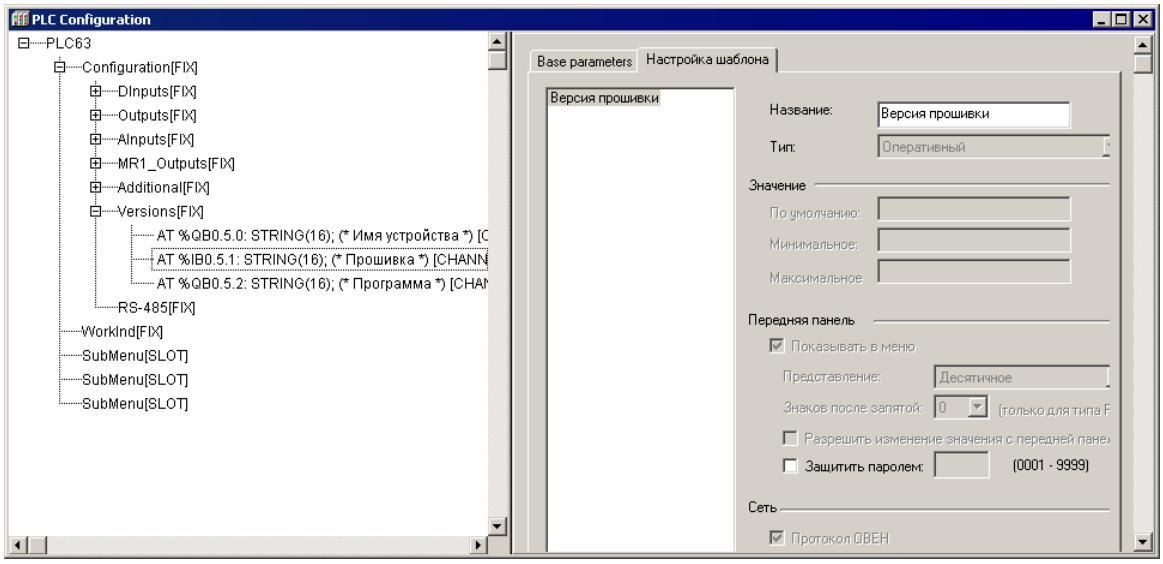


Рисунок 7.18 – Параметры конфигурирования версии прошивок

Параметры канала:

- «Имя устройства» – Имя устройства. Не изменяемое значение.
- «Прошивка» («Версия прошивки») – версия встроенного ПО микроконтроллера в формате:
  - Версия ядра встроенного ПО микроконтроллера, которая может обновляться через DBGU, в формате x.xx; например, 2.03.
  - Версия прошивки вспомогательного процессора (hex)
  - Версия прошивки контроллера ввода-вывода (hex)
- «Программа» («Версия программы») – версия пользовательской программы. Значение задаётся пользователем

7.3.7 RS-485 (модуль настройки RS-485)

Модуль настройки RS-485 определяет настройки прибора для работы в сети в режиме Slave-устройства. Определение протокола, на котором поступает запрос от внешнего мастера сети, контроллер производит автоматически.

Для работы интерфейса в режиме Master в ПО CoDeSys следует подключить библиотеку «SysLibCom» и библиотеку, используемую для работы с конкретным протоколом: «OwenNet» для работы интерфейса по протоколу ОВЕН или «Modbus» для работы по протоколу Modbus. При вызове функционального блока из библиотек «OwenNet» или «Modbus» требуется задать настройки передачи данных по сети.

Модуль не имеет каналов.

Параметры модуля (см. рисунок 7.19):

- «Скорость» – задает скорость работы порта RS485, бит/с, значения выбираются из списка (1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200), значение по умолчанию – 115200.
- «Длина слова» – задает длину слова данных в битах, значения выбираются из списка (5, 6, 7, 8), значение по умолчанию – 8.
  Примечание – Длина слова 5 и 6 бит не поддерживается в протоколах ОВЕН и MODBUS. Такая длина предназначена для реализации собственных пользовательских протоколов.
- «Четность» – задает наличие бита контроля четности, значения выбираются из списка: «Even» – четное число единиц; «Odd» – нечетное число единиц; «Space» – нулевой бит четности; «Mark» – единичный бит четности; «No Parity» – бит четности не используется. Значение по умолчанию – «No Parity», (отсутствие проверки четности)
- «Кол-во стоп биты» – задает количество стоп бит, значения выбираются из списка (один, полтора или два стоп–бита), значение по умолчанию – один стоп–бит.
- «Длина адреса» - задает базового адреса прибора для протокола ОВЕН, значение выбирается из списка (8, 11), значение по умолчанию – 11. Значение параметра устанавливает адрес прибора для обращения к нему внешним мастером. Значение адреса прибора зависит от установленного параметра «Длина адреса» (см. ниже) – при 8-битной адресации значение параметра «Адрес прибора» может принимать значение от 1 до 255, а при 11-битной адресации – от 1 до 2047. Для протокола Modbus данный параметр не действует.
- «Адрес прибора» - задает Slave-адрес контроллера в сети RS485, диапазон значений от 1 до 2047, значение по умолчанию – 16.
- «Задержка ответа» - задает временную задержку между последним байтом принятого пакета и первым байтом, передаваемым в ответ. Задержка бывает необходима для работы с устройствами, имеющими низкие скорости информационного обмена. Задается в миллисекундах. Рекомендуемый диапазон значений от 0 до 50 мс, значение по умолчанию – 5.

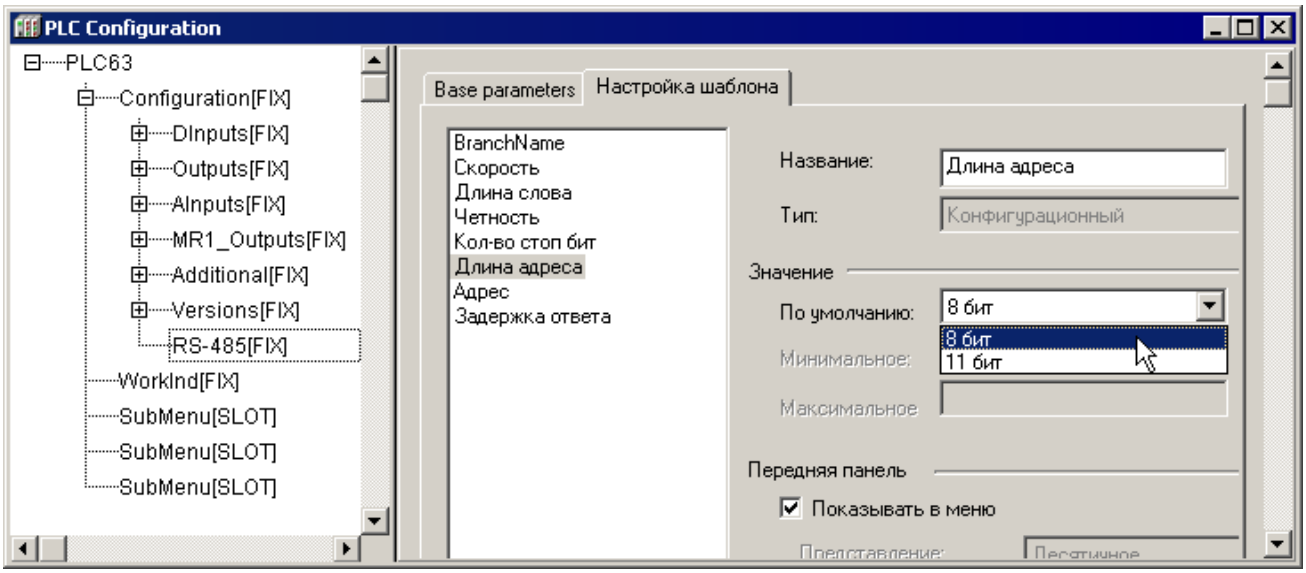


Рисунок 7.19 – Параметры конфигурирования интерфейса RS-485

7.3.8 WorkInd (модуль настройки меню)

В модуле «WorkInd» определяется количество рабочих экранов, отображаемых на ЖКИ в рабочем («экранном») режиме индикации. Количество рабочих экранов задается в параметре «Quantity of work screens» модуля «WorkInd» (см. рисунок 7.20). Диапазон значений параметра – от 1 до 16, значение по умолчанию – 6.

Для работы контроллера в рабочем («экранном») режиме индикации используется библиотека «Work_Mode.lib» (см. раздел 8.4).

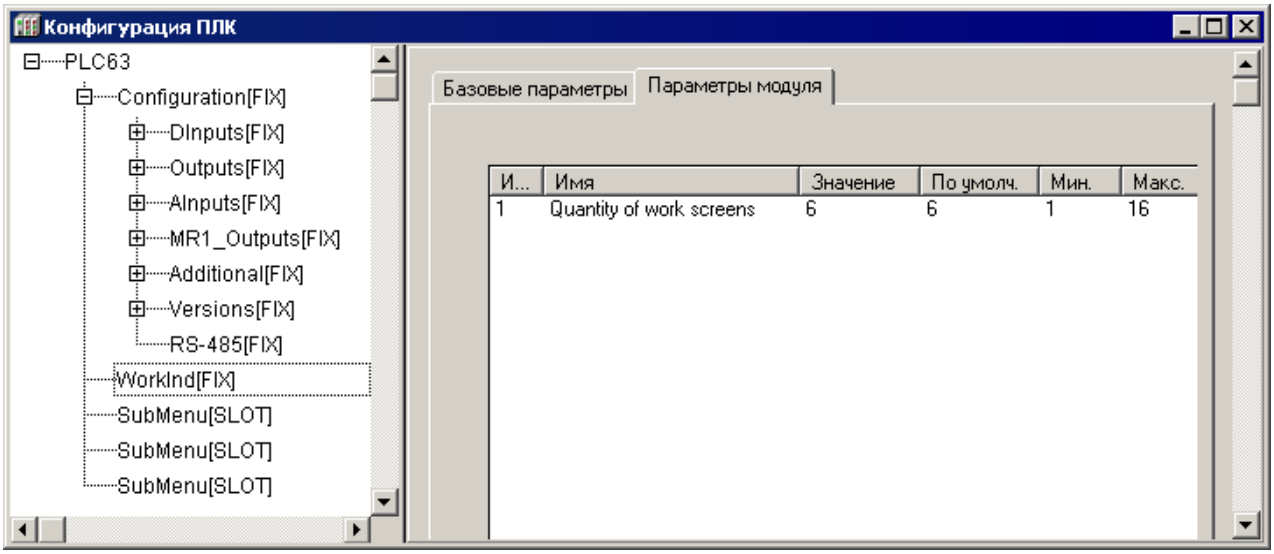


Рисунок 7.20 – Параметр конфигурирования модуля «WorkInd»

7.4 Добавляемые подэлементы (подмодули) конфигурации ПЛК (пользовательские параметры)

В данном разделе описано добавление в конфигурацию ПЛК модулей пользовательских параметров.

Подмодули (подэлементы) пользовательских параметров могут добавляться или удаляться пользователем в модулях (элементах) конфигурации «SubMenu (Ветки пользовательского меню)». Настройка пользовательских параметров производится аналогично настройке фиксированных модулей.

Процедура добавления параметров описана в разделе 7.1.1.2 и проиллюстрирована на рисунке 7.21.

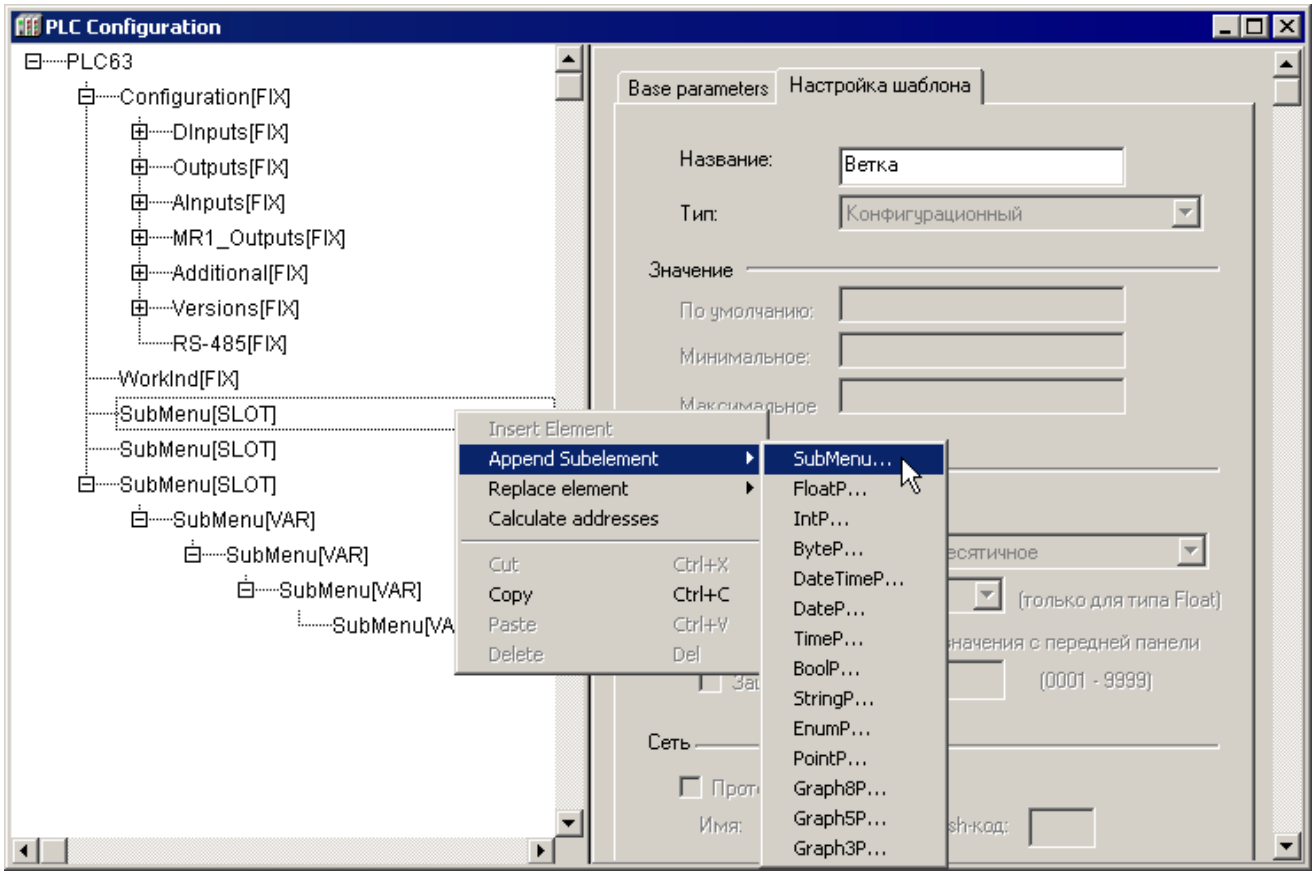


Рисунок 7.21 – Подключение нового уровня вложенности либо нового пользовательского параметра

При добавлении пользовательских параметров требуется задавать их тип – «Конфигурационный» либо «Оперативный» (см. рисунок 7.22).

Конфигурационные пользовательские параметры – значение этих параметров считывается из EEPROM при запуске контроллера. Конфигурационные пользовательские параметры доступны для редактирования с передней панели и для чтения / записи по сети (при условии установленных атрибутов).

Введенные в ПО CoDeSys (в окне режима «Конфигурация ПЛК (PLC Configuration)») новые значения этих параметров не сохраняются в EEPROM и не доступны по сети RS.

Введенные с передней панели ПЛК или по сети значения конфигурационных параметров сохраняются в EEPROM, и при выключении питания будут сохранены.

Оперативные пользовательские параметры доступны для редактирования с передней панели, по сети (при условии установленных атрибутов) и из ПО CoDeSys (в окне режима «Конфигурация ПЛК (PLC Configuration)») при редактировании значения канала в рамках установленных минимального и максимального значения.

Внимание!
При задании значения числа в поле «индекс» раздела «сеть» дерева «Конфигурация ПЛК (PLC Configuration)» следует помнить, что оперативные параметры индексируются в адресе, а конфигурационные – в индексе.

Значения оперативных параметров сохраняются в ОЗУ, и при выключении питания последнее значение параметра не будет сохранено. При включении питания значение параметра примет значение, установленное по умолчанию.

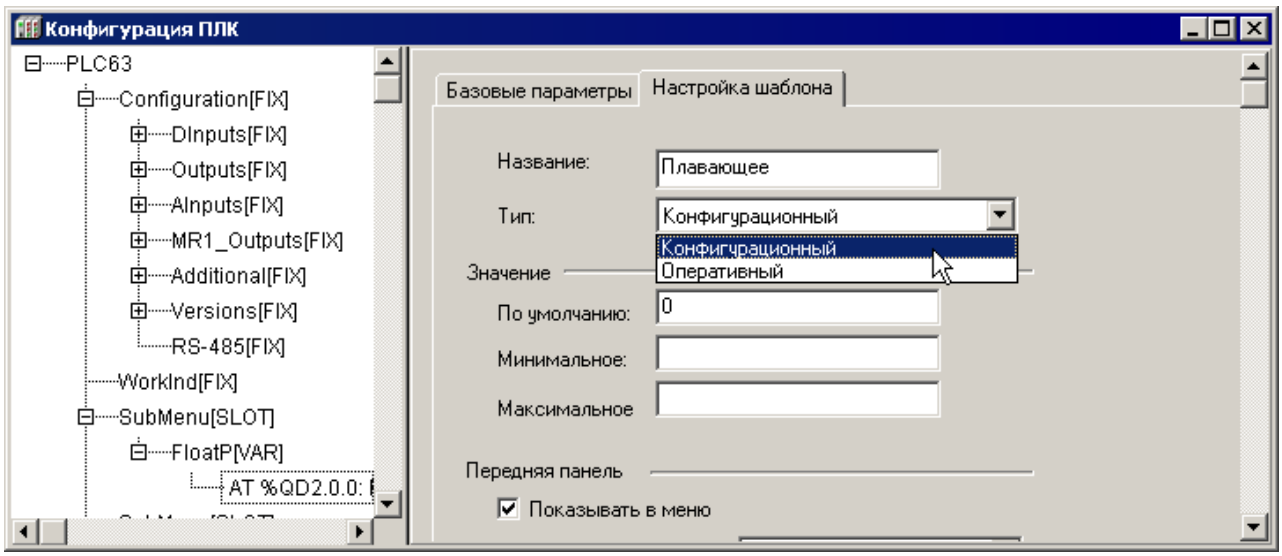


Рисунок 7.22 – Задание типа пользовательского параметра

7.4.1 Подэлемент «SubMenu (Ветка)»

В модули «SubMenu (Ветки)» можно добавлять подмодули (подэлементы) «SubMenu» (Ветка), представляющие собой вложенные ветви меню для объединения параметров по группам (веткам). Вложенность Submenu может достигать четырех уровней (см. рисунок 7.20).

7.4.2 Пользовательские параметры

В модули «SubMenu (Ветки)» можно добавлять подэлементы (пользовательские параметры) типов (см. рисунок 7.20), перечисленных ниже.

Примечание. Параметры типа «DateTimeP», «DateP», «TimeP» не влияют на задание значений часов реального времени ПЛК, а служат для передачи в пользовательскую программу информации об определенном времени, например – о времени включения какого-то устройства или о времени и дате перехода в специальный режим.

- «**FloatP**» – подэлемент, задающий число с плавающей точкой типа Real (IEEE Float 32); имеет канал типа «Real». Для задания номера регистра по протоколу Modbus следует учитывать, что значение должно быть четным, т.к. параметр занимает 2 регистра (4 байта).
- «**IntP**» – подэлемент, задающий целое число в диапазоне от 0 до 65535; имеет канал типа «Int».
- «**ByteP**» – подэлемент, задающий значение стандартного типа Byte.
- «**DateTimeP**» – подэлемент, содержащий дату и время; имеет канал типа DWord, в котором хранится значение времени и даты в специальном формате. Значение параметра задается в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС, см. рисунок 7.23. Максимальное и минимальное значение для данного параметра не устанавливается. Для преобразования данных подэлемента «DateTimeP» в стандартный формат используется специальная функция «DT_TO_SystemTimeDate» из библиотеки «Owen_fb», которая возвращает указатель на структуру «SystemTimeDate» из библиотеки «SYSLIBTIME». Для обратного

преобразования используется функция «SystemTimeDate_TO_DT» из той же библиотеки («Owen_fb»).

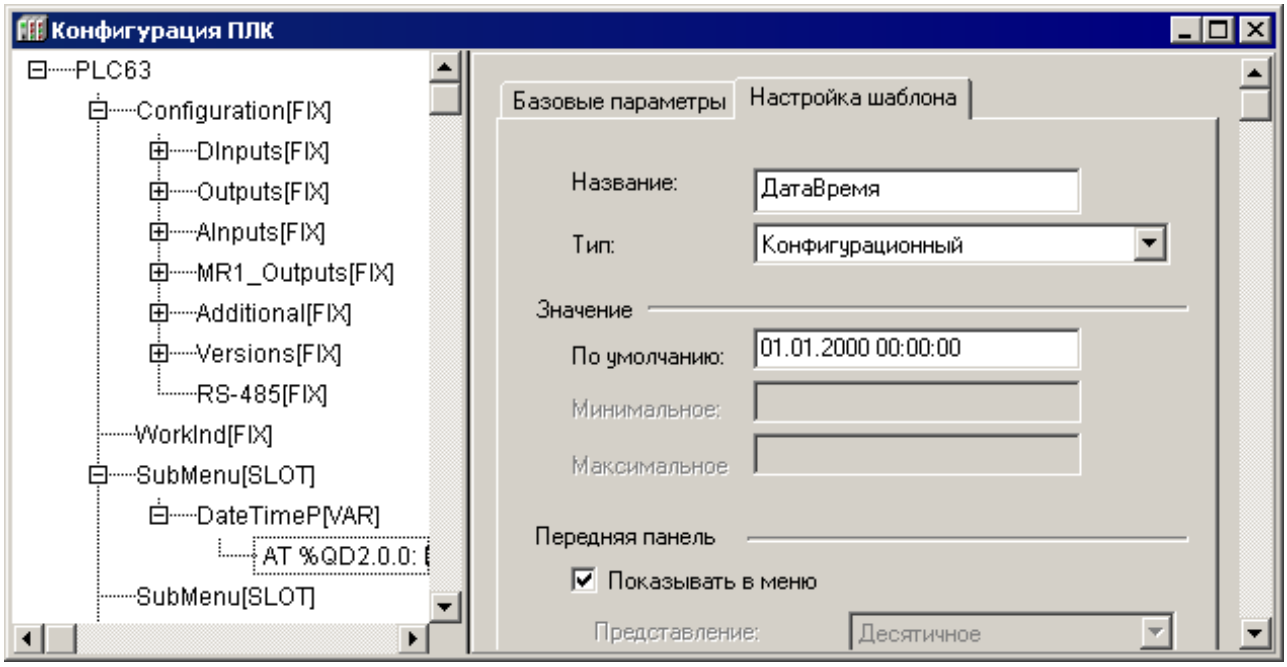


Рисунок 7.23 – Формат значения параметра «DateTimeP»

- «**DateP**» – подэлемент, устанавливающий дату; имеет канал типа DWord, в котором хранится значение даты в специальном формате.
  Значение параметра задается в формате ДД.ММ.ГГГГ. Максимальное и минимальное значение для данного параметра не устанавливаются.
  Для преобразования данных параметра «DataP» в стандартный формат используется специальная функция «DT_TO_SystemTimeDate» из библиотеки «Owen_specific_tools», которая возвращает указатель на структуру «SystemTimeDate» из библиотеки «SYSLIBTIME». Для обратного преобразования используется функция «SystemTimeDate_TO_DT» из той же библиотеки («Owen_specific_tools»).
- «**TimeP**» – подэлемент, задающий время; имеет канал типа DWord, в котором хранится значение времени в специальном формате. Значение параметра задается в формате ЧЧ:ММ:СС. В аналогичном формате задается минимальное и максимальное значение.
  Для преобразования данных параметра TimeP в стандартный формат используется специальная функция «DT_TO_SystemTimeDate» из библиотеки «Owen_specific_tools», которая возвращает указатель на структуру «SystemTimeDate» из библиотеки «SYSLIBTIME». Для обратного преобразования используется функция «SystemTimeDate_TO_DT» из той же библиотеки («Owen_specific_tools»).
- «**BoolP**» – подэлемент, задающий булевский параметр: состояние «Вкл.» (Включено) или «Выкл.» (Выключено); имеет канал типа Bool. Максимальное и минимальное значение для данного параметра не устанавливается.
- «**StringP**» – подэлемент, задающий строку; имеет канал типа String 16. Параметр может использоваться для передачи в программу ПЛК строковой информации по сети. Параметр не доступен для изменения с передней панели. Если установлен флажок переключателя «Разрешить изменения значения по сети», то параметр доступен для изменения его по сети и из программы ПЛК. Если флажок не установлен, то в параметр можно записывать только значе-

ния из программы ПЛК.

Примечание. Если для параметра задан тип «Конфигурационный», то параметр будет отображать статическую строку, недоступную для редактирования ни с передней панели ПЛК, ни по сети.

– «**EnumP**» – подэлемент, имеющий вид списка с набором заранее определенных значений; при редактировании параметра с передней панели на ЖКИ отображаются строки из списка. Список изначально определен в ПО контроллера и не может быть изменен пользователем. Модуль имеет канал типа Byte, в котором содержится позиционный номер выбранного элемента списка.

  Для параметра можно задавать минимальное и максимальное значения, и значение по умолчанию, которое должно входить в установленные ограничения. Для удобства ориентации по списку при конфигурировании показывается позиционный номер выбранной строки списка (при отображении на ЖКИ номер не показывается), см. рисунок 7.24. Содержание списка «Enum» представлено в Приложении В.

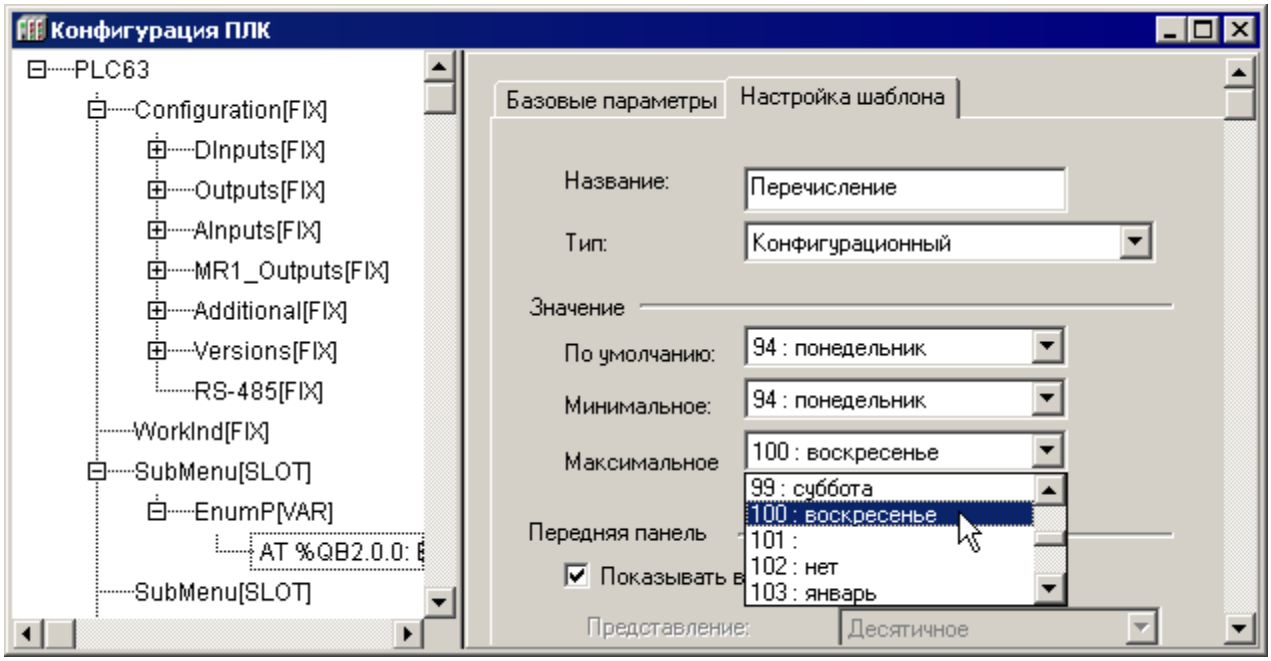


Рисунок 7.24 – Формат значения параметра «EnumP»

– «**PointP**» – подэлемент задает точку графика – соотношение значений переменных (X и Y). Добавив требуемое количество подэлементов, можно определить зависимость (график) Y=f(X). Значение вводится в поле «Значение / По умолчанию» в формате: «Координата X; Координата Y». Диапазон задания координат X и У от минус 3276.7 до 3276.7. Модуль имеет канал типа DWord имеющий формат: 4-х байтовое поле, в котором первые 2 байта – это значение координаты Y, вторые два байта – координаты X, значения которых умножено на 10 в степени, заданной параметром «Знаков после запятой» (см. рисунок 7.25).

  Значение параметра «Знаков после запятой» (0, 1, 2 или 3) задается для обеих координат. На ЖКИ будет отображаться столько знаков после запятой, сколько указанно в этом параметре. Например, если в поле «По умолчанию» указаны значения 11; 12.5 , а параметру «Знаков после запятой» задано значение 2, то на ЖКИ отобразятся значения 11.00 и 12.50.

  Для преобразования координаты точки графика в стандартный формат ис-

пользуется специальная функция «DWORD_TO_POINT» из библиотеки owen_fb. Для преобразования из стандартного формата в формат «DWORD» используется функция «POINT_TO_DWORD» из библиотеки owen_fb

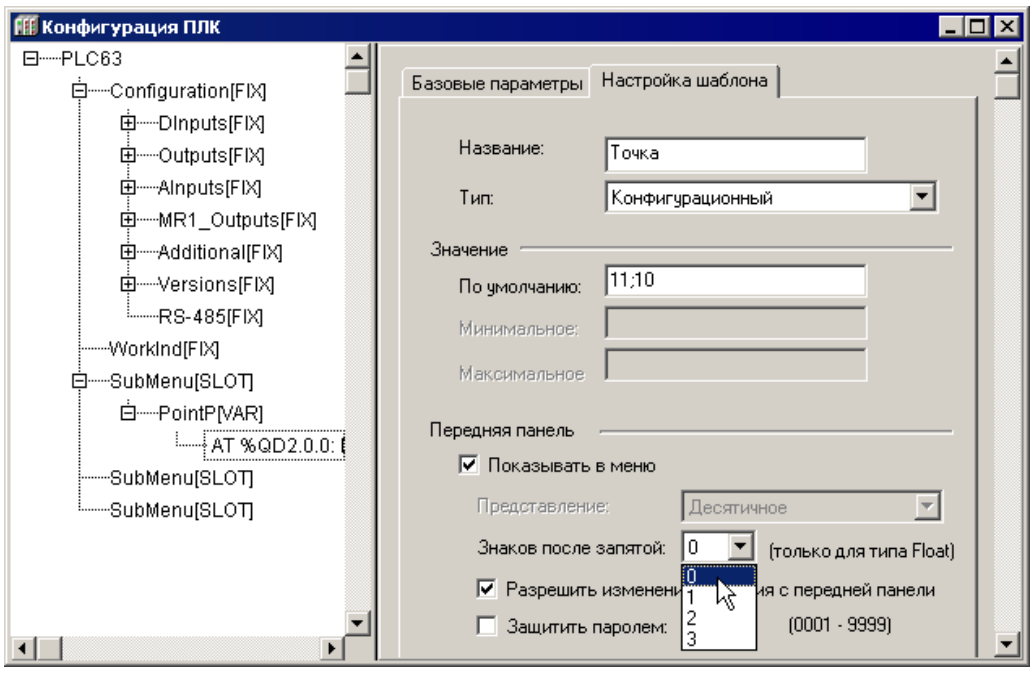


Рисунок 7.25 – Формат значения параметра «PointP»

– «**Graph8P**», «**Graph5P**», и «**Graph3P**» – подэлемент задает совокупность точек графика – соотношения значений переменных (X и Y) в точках, количество которых определяется наименованием выбранного подэлемента – 8, 5 или 3. Добавив требуемое количество подэлементов, можно определить зависимость (график) Y=f(X). Каждый подмодуль представляет из себя «ветку» конфигурации, содержащую группу параметров (каналов) и параметр – количество точек.

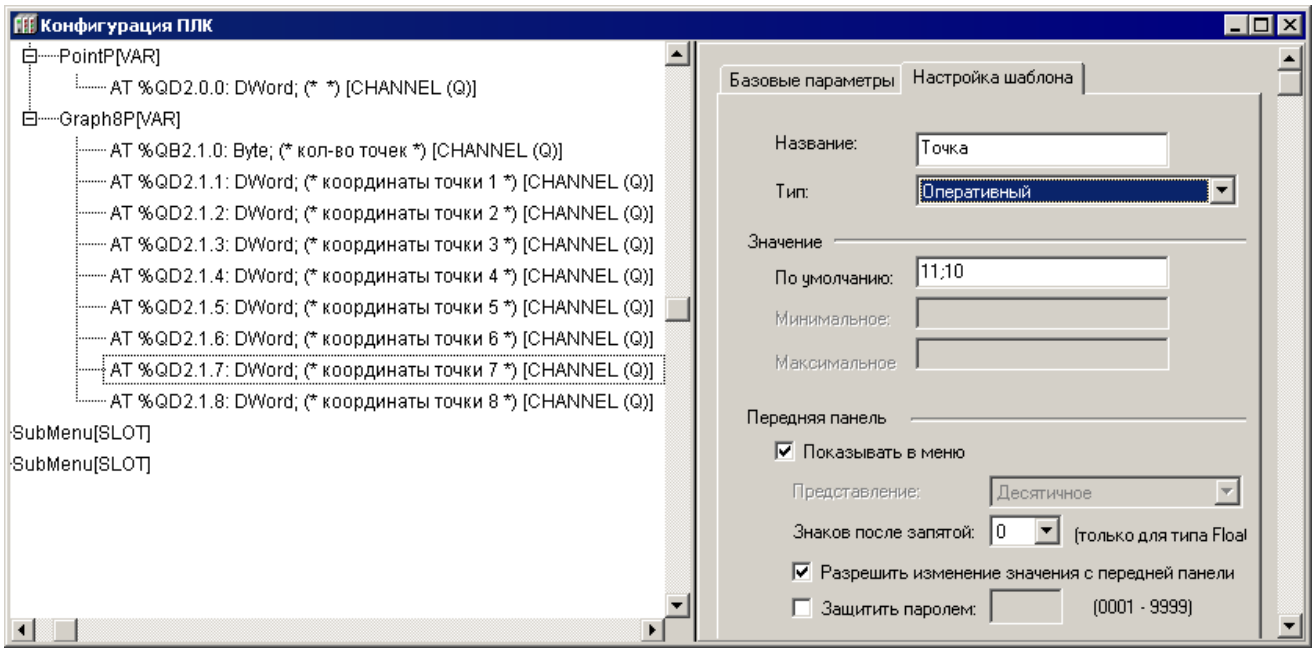


Рисунок 7.26 – Формат значения параметра «Graph8»

	Внимание! Число одновременно отображаемых переменных в окнах отладки и/или визуализации сокращено. Точное количество переменных для отладки зависит от их типа и от частоты их обновления, но в среднем количество переменных для отладки и визуализации не должно превышать 30-50 штук. Это ограничение не относится к общему количеству переменных, используемых в проекте.

7.5 Размещение переменных в области памяти протокола Modbus при работе в режиме Slave-устройства

Slave режим работы портов RS в ПЛК63 позволяет обращаться к оперативным и конфигурационным параметрам, определенным в окне режима «Конфигурация ПЛК (PLC Configuration)», по протоколу Modbus.

Возможны два способа обращения: одиночный доступ по адресу, заданному в окне режима «Конфигурация ПЛК (PLC Configuration)», и групповой доступ к памяти ввода / вывода.

При одиночном доступе возможно чтение и запись значений параметров при обращении по протоколу Modbus по конкретному адресу, заданному в окне в окне режима «Конфигурация ПЛК (PLC Configuration)» (см. раздел 7.1.1.5.1). Доступ возможен только к одному регистру (при работе с двухбайтными параметрами) или к двум регистрам (при работе с 4-х байтными параметрами). Чтение осуществляется функциями 03, 04, 01, 02 запись функциями 06, 16, 05, 15.

Область памяти от 0 до 287 – это область хранения четырехбайтных переменных (и переменных формата string), причем область от 0 до 81 зарезервирована под конфигурационные параметры, а область от 82 до 287 доступна под пользовательские параметры (см. рисунок 5.1).

Область памяти от 288 до 512 – это область двух-байтных переменных, причем область от 288 до 311 зарезервирована под конфигурационные параметры, а область от 332 до 511 доступна под пользовательские параметры.

Для группового чтения параметров по протоколу Modbus возможно обращение к области регистров Modbus, начинающейся с адреса 4096. В ней располагается копия памяти ввода/вывода, доступно групповое чтение значений параметров из области этой области функциями 03, 04, 01, 02. Область %I доступна с регистра 4096 до 4184 и область %Q от 4185. Вторая граница области %Q не определена, т.к. пользовательские параметры при создании добавляются именно в нее, а их количество заранее не известно.

Распределение области регистров протокола Modbus показано на рис. 5.1.

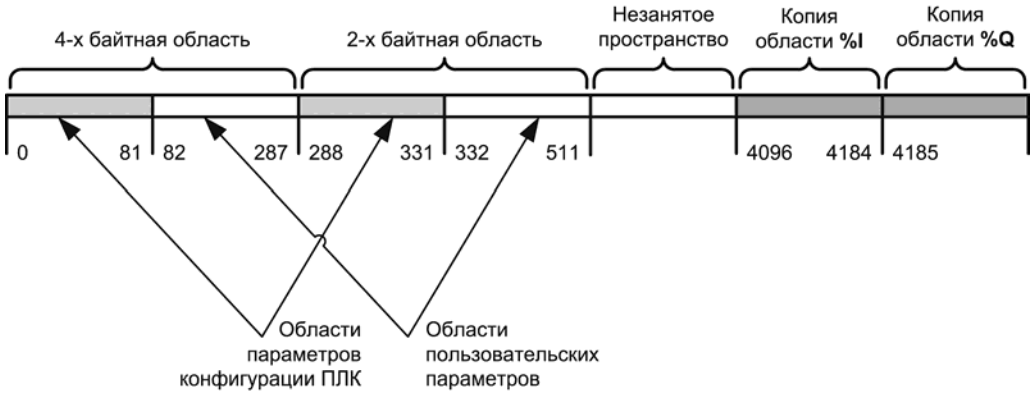


Рисунок 7.27 – Распределение области регистров протокола Modbus при работе ПЛК в режиме Slave-устройства.

8 Работа с ЖКИ ПЛК

ЖКИ контроллера предназначен для отображения и установки значений программируемых параметров прибора, определяющих его настройку и работу в процессе эксплуатации.

Таким образом, ЖКИ в совокупности с кнопками передней панели контроллера может выполнять отдельные функции программатора ПЛК, хотя и в объеме, значительно меньшем, чем ПК с установленным на нем ПО CoDeSys.

На ЖКИ отображается меню прибора (далее – меню), предназначенное для доступа к программируемым параметрам прибора, см. рисунок 8.1.

Рисунок 8.1 – Передняя панель ПЛК63

Двухстрочное (в ПЛК63) или четырехстрочное (для ПЛК73) меню контроллера отображает конфигурацию ПЛК (см. раздел 7). В строках меню отображаются наименования элементов (подэлементов) конфигурации.

При отображении наименования ветви дерева конфигурации (то есть, элемента, имеющего вложенные подэлементы), перед наименованием отображается символ « ⁘ ».

Символ « > » перед названием ветви в верхней строке указывает на уровень иерархии текущей ветви (“>” первый уровень, “>>” второй уровень).

При отображении наименования параметра, перед наименованием отображается символ « * ».

Переход в выбранную подветвь (на уровень вложенности, дочерний относительно текущего) производится нажатием кнопки ВВОД.

Выход из подветви (на уровень вложенности, родительский относительно текущего) осуществляется кнопкой ВЫХОД.

ЖКИ контроллера может функционировать в следующих режимах:

- **Рабочий режим** – режим индикации, действующий по умолчанию: ЖКИ управляется пользовательской программой. Подробнее см. раздел 8.1 . Включает:
  - **Монопольный** режим индикации (single_mode) – режим индикации, при котором на ЖКИ контроллера отображается только один выбранный экран и блокированы системные функции клавиатуры.
  - **Рабочий** режим индикации (work_mode) – режим, при котором пользователь может управлять выбором отображаемого на ЖКИ экрана и работой ПЛК при помощи кнопок на лицевой панели.
- **Конфигурационный режим (режим меню)** – режим индикации, в котором производится задание конфигурационных параметров: параметров конфигурации периферийного оборудования (входов, выходов и т.д.) и пользовательских конфигурационных параметров. Подробнее см. раздел 8.2 . Включает:

– Режим **редактирования** (editor_mode) – режим редактирования элемента меню на ЖКИ.

– **Отладочный режим** – служебный режим индикации, используемый для калибровки и заводской проверки контроллера, просмотра возможных ошибок контроллера. Подробнее см. раздел 8.3 .

Схема режимов индикации ЖКИ, использования кнопок передней панели ПЛК и функций библиотек программных компонентов для переходов между режимами и для работы в каждом режиме представлена на рисунке 8.2.

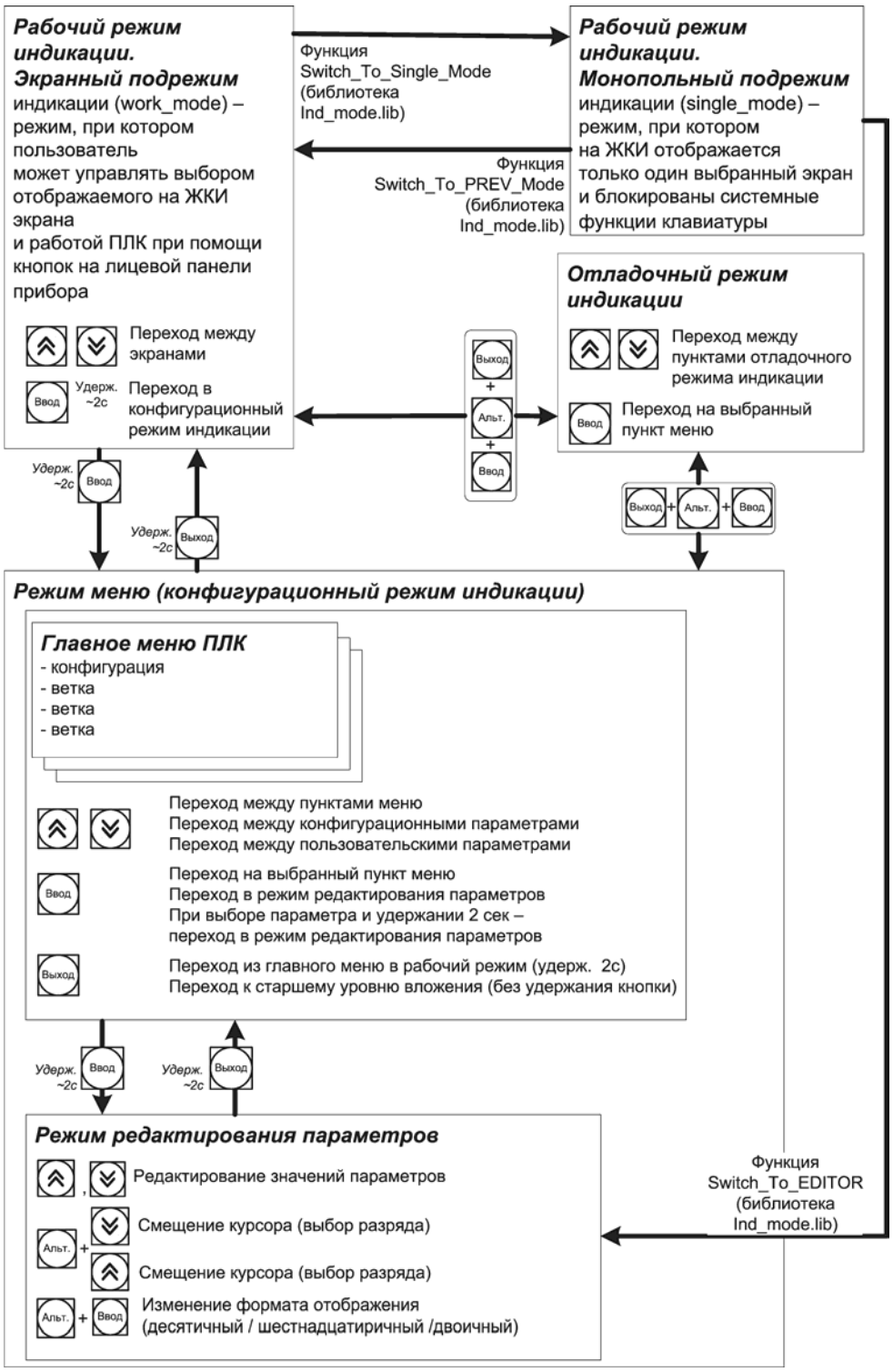


Рисунок 8.2 – Режимы индикации ЖКИ

8.1 Рабочий режим индикации

В рабочем режиме индикации ЖКИ может функционировать в одном из двух подрежимов:

- экранном;
- монопольном.

8.1.1 Экранный подрежим индикации

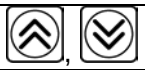
В экранном подрежиме индикации на ЖКИ отображается один из экранов, содержание которых задано конфигурацией контроллера. Количество рабочих экранов устанавливается в параметре «Quantity of work screens» модуля «WorkInd» конфигурации контроллера (см. раздел 7.3.8).Переключение между экранами производится нажатием кнопок и клавиатуры контроллера. Для работы в экранном рабочем режиме используется библиотека «Work_Mode.lib». Использование компонентов библиотеки позволяет выполнять следующие функции (подробнее см. раздел 8.4.1):

- очистки экрана;
- получения номера текущего экрана;
- получения числа рабочих экранов;
- установки номера текущего экрана;
- вывода строки в указанную позицию экрана;
- вывода числа типа «Real» в указанную позицию экрана;
- вывода числа типа «DINT» в указанную позицию экрана.

В экранном режиме индикации кнопки клавиатуры контроллера выполняют функции, перечисленные в таблице 8.1.

Для перехода в монопольный подрежим режима используется функция Switch_To_Single_Mode (библиотека Ind_mode.lib).

Таблица 8.1 – Экранный подрежим. Функциональное назначение кнопок

Кнопки	Функциональное назначение
	Переход в Конфигурационный режим индикации: вход в главное меню ПЛК63 (по продолжительному нажатию: ~2 сек). Только из режима work_mode.
	Переход между экранами. Только для режима work_mode.
+ Включение питания	Пользовательская программа не будет запущена
	Перезагрузка контроллера
	Переход в Отладочный режим индикации

8.1.2 Монопольный подрежим индикации

В монопольном подрежиме индикации на ЖКИ отображается один экран. При этом кнопки , и могут быть использованы в рамках функционирования пользовательской программы.

Для отображения символьной информации в монопольном рабочем режиме и для переключения между режимами используется библиотека «Ind_Mode.lib».

Использование компонентов библиотеки позволяет выполнять следующие функции (подробнее см. раздел 8.4.2):

- получение информации о текущем режиме работы ЖКИ;

- переход в рабочий монопольный режим работы ЖКИ.
- возврат в предшествующий режим индикации.
- очистка содержимого ЖКИ в монопольном и экранном подрежимах.
- вывод строки на ЖКИ в монопольном режиме индикации.
- чтение строки с ЖКИ в монопольном режиме индикации.
- возврат из рабочего монопольного режима в режим редактирования параметра.

В монопольном режиме индикации кнопки клавиатуры контроллера выполняют функции, перечисленные в таблице 8.2.

Для перехода в экранный (предыдущий) подрежим режима используется функция Switch_To_PREV_Mode (библиотека Ind_mode.lib).

Для перехода в режим редактирования параметров используется функция Switch_To_EDITOR_Mode (библиотека Ind_mode.lib).

Таблица 8.2 – Монопольный подрежим. Функциональное назначение кнопок

Кнопки	Функциональное назначение
ПУСК/СТОП + ВЫХОД + ВВОД	Перезагрузка контроллера
АЛЬТ. + ВЫХОД + ВВОД	Переход в Отладочный режим индикации

8.2 Конфигурационный режим индикации

Конфигурационный режим индикации используется для редактирования параметров конфигурации ПЛК63.

Конфигурационные параметры контроллера предназначены для настройки периферийного оборудования контроллера (описание см. в разделе 7.3) и для настройки пользовательской программы описание см. в разделе 7.4 .

Конфигурационный режим работы ЖКИ предназначен для доступа к конфигурационным параметрам. Переход в конфигурационный режим индикации из рабочего экранного режима индикации осуществляется нажатием кнопки ВВОД приблизительно на 2 секунды. После перехода в этот режим на ЖКИ отображается главное меню ПЛК.

8.2.1 Главное меню контроллера

Главное меню контроллера представляет из себя многоуровневую иерархическую древовидную структуру, включающую конфигурационные параметры, распределенные по ветвям.

При пролистывании ветвей и переходу по ветви конфигурационные параметры помечаются символом на ЖКИ:

- Символ « ⁘ » перед названием пункта меню указывает на то, что пункт является ветвью.
- Символ « * » перед названием пункта меню указывает на то, что пункт является параметром.
- Символ « **>** » перед названием ветви в верхней строке указывает на уровень иерархии текущей ветви (« **>** » – первый уровень, « **>>** » – второй уровень и т.д.).

В конфигурационном режиме индикации кнопки клавиатуры контроллера выполняют функции, перечисленные в таблице 8.3.

Таблица 8.3 – Конфигурационный режим. Функциональное назначение кнопок

Кнопки	Конфигурационный режим индикации
ВВОД	1) Переход к младшему уровню вложенности. 2) Переход в режим редактирования параметров (продолжительное нажатие ~2 сек).
ВЫХОД	1) Переход на старший уровень вложенности. 2) Выход из конфигурационного режима индикации в главное Меню ПЛК63 (продолжительное нажатие 2 сек).
,	Переход между ветками, конфигурационными и пользовательскими параметрами
ПУСК СТОП + ВЫХОД + ВВОД	Перезагрузка прибора
АЛЬТ. + ВЫХОД + ВВОД	Переход в Отладочный режим индикации

8.2.2 Редактирование значений параметров

В контроллере предусмотрено редактирование значений параметров следующих типов: число с плавающей точкой; целые числа; координаты точки; время; дата; время и дата; параметр со значением, выбираемым из списка.

Для перехода в режим редактирования параметра следует выбрать требуемый параметр (см. п. 1.2.1), нажать кнопку ВВОД и удерживать ее приблизительно 2 секунды.

Для изменения значения параметров, например, числа с плавающей точкой, используются кнопки и . При этом нажатием кнопок АЛЬТ. + курсор передвигается к старшим разрядам, нажатием кнопок АЛЬТ. + – к младшим.

Положение десятичной точки (порядок значения) можно изменять нажатием кнопок и , при курсоре, установленном на символе «.».

Для записи измененного значения следует нажать кнопку ВВОД и удерживать ее приблизительно 2 секунды.

Для выхода из режима редактирования без изменения значения параметра следует нажать кнопку ВЫХОД и удерживать ее приблизительно 2 секунды.

Для смены режима отображения значения параметра (десятичный / двоичный / шестнадцатеричный) следует в режиме редактирования нажать кнопки АЛЬТ. + ВВОД.

Если доступ к параметру требует ввода пароля, то ввод пароля производится нажатием кнопок и , а нажатие кнопок АЛЬТ. + подтверждает ввод пароля.

В подрежиме редактирования параметра конфигурационного режима индикации кнопки клавиатуры контроллера выполняют функции, перечисленные в таблице 8.4.

Таблица 8.4 – Режим редактирования параметра. Функциональное назначение кнопок

Кнопки	Режим редактирования значения параметра (в конфигурационном режиме индикации)
ВВОД	Запись в память измененных пользователем значений параметров (продолжительное нажатие 2 сек).
ВЫХОД	Переход из режима Редактирования в Конфигурационный режим без сохранения значения параметра.
▲, ▼	1) Изменение значение параметра 2) Ввод значения пароля
АЛЬТ. + ▲ АЛЬТ. + ▼	1) Изменение положения курсора; 2) Переход к дополнительному окну редактирования и возврат при редактировании составных параметров. 3) Подтверждение пароля
АЛЬТ. + ВВОД	Смена вида отображения

8.3 Отладочный режим индикации

Отладочный режим индикации необходим пользователю для возможной перезагрузки прибора и для просмотра кода ошибки при возникновении таковой. Расшифровка кодов ошибок приведена в документе «РЭ ПЛК63» (см. приложение Н [5]).

Отладочный режим индикации содержит следующие пункты меню:

- «Коды ошибок» – включает в себя просмотр кодов ошибок. Перечень и расшифровка кодов ошибок приведен в документе РЭ ПЛК63 и в Приложении Д.
- «Перезагрузка» – приводит к перезагрузке прибора.
- «Сброс EEPROM» – сброс всех конфигурационных и пользовательских параметров в значения, установленные по умолчанию.
- «Калибровка АЦП, ХС, ЦАП» - используются только на заводе-изготовителе.

Переходы по пунктам меню осуществляются так же, как и в конфигурационном режиме индикации.

В отладочном режиме индикации кнопки клавиатуры контроллера выполняют функции, перечисленные в таблице 8.5.

Таблица 8.5 – Отладочный режим. Функциональное назначение кнопок

Кнопки	Отладочный режим индикации
ВВОД	Переход к младшему (дочернему) уровню вложенности
ВЫХОД	Переход на старший (родительский) уровень вложенности
▲, ▼	Переход по меню
АЛЬТ. + ВЫХОД + ВВОД	Выход из Отладочного режима индикации

8.4 Библиотеки программных компонентов для работы с ЖКИ

При программировании ПЛК используются две специализированные библиотеки дополнительных программных компонентов, предназначенные для работы с ЖКИ контроллеров.

8.4.1 Библиотека Work_Mode.lib (работа с ЖКИ в рабочем экранном режиме)

Библиотека предназначена для обеспечения работы ЖКИ контроллера в экранном рабочем режиме (режиме по умолчанию). Компоненты библиотеки (функции) рекомендуется использовать для отображения цифровых значений и символьных строк на ЖКИ при выполнении пользовательской программы.

Библиотека содержит следующие функции:

- «ClearScreen» – очистка экрана в монопольном и экранном режимах.
- «GetWorkScreen» – получение номера текущего экрана.
- «GetWorkScreenCount» – получение числа рабочих экранов.
- «SetWorkScreen» – установление номера текущего экрана.
- «ShowDint» – вывод числа типа DINT в указанной позиции.
- «ShowReal» – вывод числа типа Real в указанной позиции.
- «ShowString» – вывод строки в указанной позиции.

Подробное описание библиотеки приведено в документе «Библиотеки Ind_Mode и Work_Mode» (см. Приложение Н, [7]).

8.4.1.1 Отображение специализированных символов

При необходимости функции «ShowDint», «ShowReal» и «ShowString» библиотеки Work_Mode позволяют вывести на заданную позицию (или позиции) ЖКИ требуемые специализированные символы.

Синтаксис задания кодов символов одинаков для всех трех функций: ‘$F2’ (в кавычках, после знака $, приводится старший («F») и младший («2») биты кода. Например:

- (* выводим символ*) ShowString(1, 2, ‘$9F’);
- (*выводим символ*) ShowDInt(0,1,0,’%d $9F’, DInt1);
- (*выводим символ*) ShowReal(0,1,0,’%2.1f $9F’, real1);

Во всех трех случаях на вторую позицию первой строки ЖКИ будет выведен символ с кодом 9f: «¾».

Подробно об использовании перечисленных функций см. документ «Библиотеки Ind_Mode и Work_Mode» (см. Приложение Н, [7]).

Доступные для отображения символы и их коды приведены в приложении Ж.

8.4.2 Библиотека Ind_Mode.lib (управление режимами ЖКИ и вывод символьных строк на ЖКИ в монопольном рабочем режиме)

Библиотека предназначена для переключения между режимами индикации ЖКИ и для вывода строковых констант на ЖКИ в монопольном рабочем режиме. Библиотеку следует использовать при создании нестандартных способов отображения сообщений на ЖКИ, когда функций экранного рабочего режима не хватает.

Библиотека содержит три программных компонента (определяющих используемые пользовательские типа данных), используемые при программировании работы ЖКИ (см. рисунок 8.3):

- «TYPE DISP_MODES» – задаёт меню отладочного режима индикации ЖКИ.
- «TYPE PARAM_NODE» – структура описания параметра (пользовательского или параметра конфигурирования ПЛК), доступного для редактирования с передней панели; **Внимание!** В этой структуре – не изменять ничего!
- «TYPE PARAMETER_DESCRIPTOR» – используется функцией «read_by_hash» (возвращаемое значение).

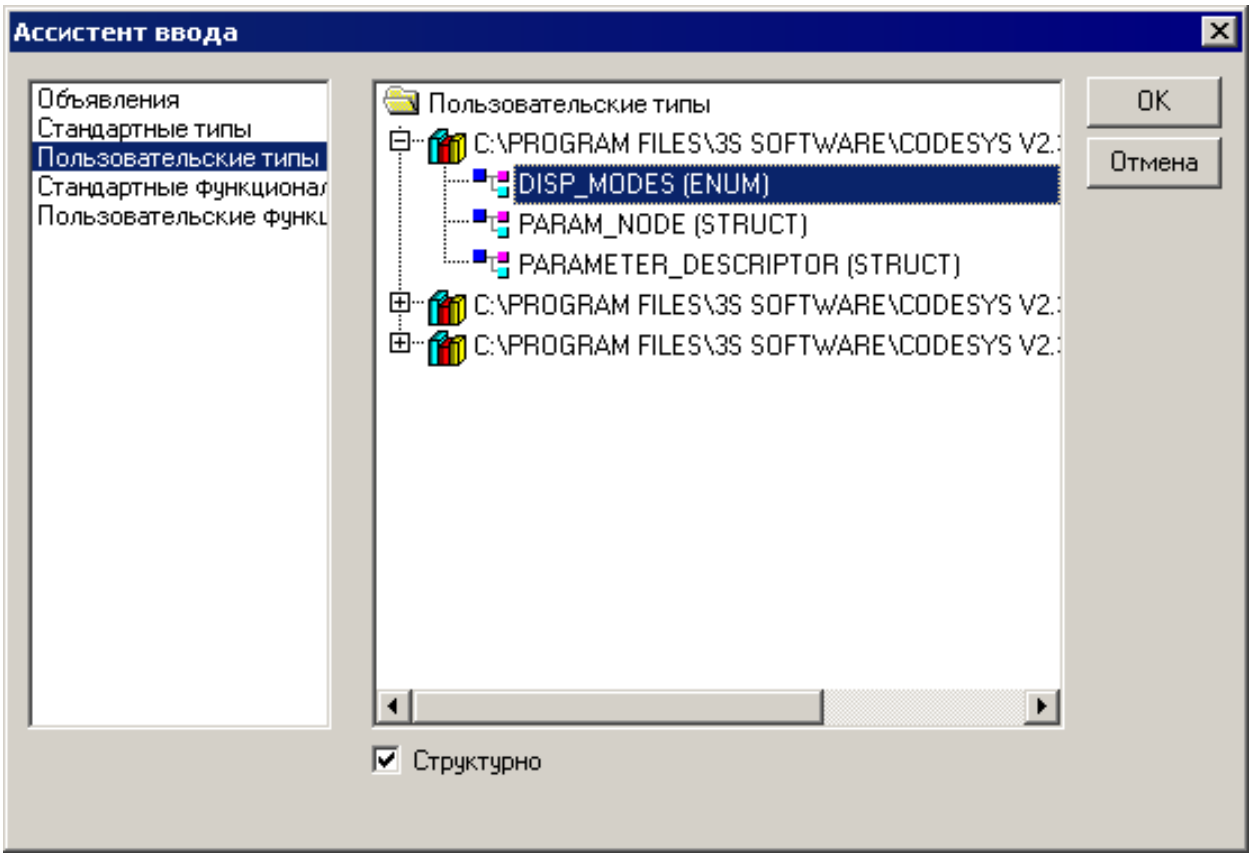


Рисунок 8.3 – Библиотека Ind_Mode.lib. Пользовательские типы данных

Библиотека содержит функции:

- «Clear_Display» – очистка содержимого ЖКИ в монопольном и экранном подрежимах; в конфигурационном и отладочном режимах – не работает.
- «Get_Display_Mode» – получение текущего режима работы ЖКИ.
- «Read_Display» – чтение строки с ЖКИ в монопольном режиме индикации.
- «Switch_To_EDITOR» – возврат из рабочего монопольного режима в редактор параметра.
- «Switch_To_PREV_Mode» – возврат из рабочего монопольного режима в предшествующий режим индикации.
- «Switch_To_Single_Mode» – переход в рабочий монопольный режим работы ЖКИ.
- «Write_Display» – вывод строки на ЖКИ в монопольном режиме индикации.

Подробное описание библиотеки приведено в документе «Библиотеки Ind_Mode и Work_Mode» (см. Приложение Н, [7]).

9 Обновление встроенного ПО микроконтроллера и Target-файлов

Компания ОВЕН совершенствует производимые контроллеры и их программное обеспечение, в том числе – встроенное ПО микроконтроллера и Target-файлы.

Обновленное ПО микроконтроллера и Target-файлы следует устанавливать на используемый ПЛК только в том случае, если наблюдаются сбои в работе действующего ПО. Если ПЛК работает без сбоев, то производить обновление программного обеспечения не рекомендуется.

Обновленное ПО и соответствующие ему версии Target-файлов можно скачать с сайта

Внимание!
Обновление ПО микроконтроллера и Target-файл ПЛК должны иметь соответствующие друг другу версии.

9.1 Определение актуальной версии ПО микроконтроллера

Определение актуальной версии ПО микроконтроллера выполняется просмотром значения параметра «Versions» (версии) конфигурации ПЛК (см. п. 7.3.6).

9.2 Обновление ПО микроконтроллера

Для обновления ПО микроконтроллера на новую версию предназначен интерфейс DBGU.

Для соединения с ПК следует использовать специальную плату-переходник, а также кабель КС1, либо КС2, соединяющий плату-переходник с портом RS-232 ПК (напрямую, либо через адаптер USB/RS-232). Кабель КС2 и плата-переходник приобретаются отдельно.

Внимание!

1) Порт DBGU предназначен исключительно для сервисных целей. Запрещается подключать к нему любые устройства на этапе наладки и эксплуатации, за исключением платы для обновления ПО микроконтроллера при необходимости выполнения его обновления.
2) При обновлении ПО микроконтроллера сбрасывается заданное значение времени, и его необходимо переустановить.
3) При обновлении ПО микроконтроллера сбрасываются значения всех конфигурационных параметров.
4) После обновления ПО микроконтроллера следует выполнить сброс EEPROM и сконфигурировать прибор заново.
5) Значения параметров, сохраненные программой тиражирования из старого ПО микроконтроллера, записать (перенести) в новое ПО микроконтроллера будет невозможно. При необходимости использования значений параметров из старого ПО микроконтроллера, следует сохранить прежнюю конфигурацию при помощи программы тиражирования заново.

9.2.1 Обновление ПО микроконтроллера с использованием утилиты «Перепрошивка ПЛК»

Для обновления ПО микроконтроллера с использованием утилиты «Перепрошивка ПЛК» следует:

1) В программе-навигаторе дистрибутивного диска нажатием соответст-

вующей кнопки (см. рисунок 9.1) – запустить на выполнение программу установки (инсталлятор) программы обновления ПО микроконтроллера требуемой модификации ПЛК (или в папке «Patching» дистрибутивного диска выбрать и запустить файл требуемой утилиты «Перепрошивка ПЛК <Версия>.exe»).

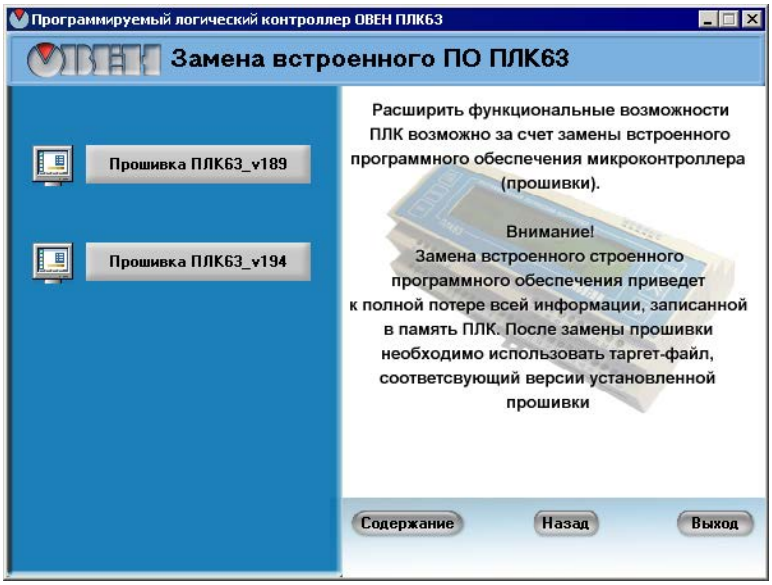


Рисунок 9.1 – Окно программы-навигатора дистрибутивного диска

2) Запустить инсталлятор программы смены прошивки (ПО микроконтроллера) – например, щелчком на ярлыке программы ( ).
В открывающихся окнах программы – выполнять действия, описание которых приведено в окне. Например, см. рисунок 9.2.

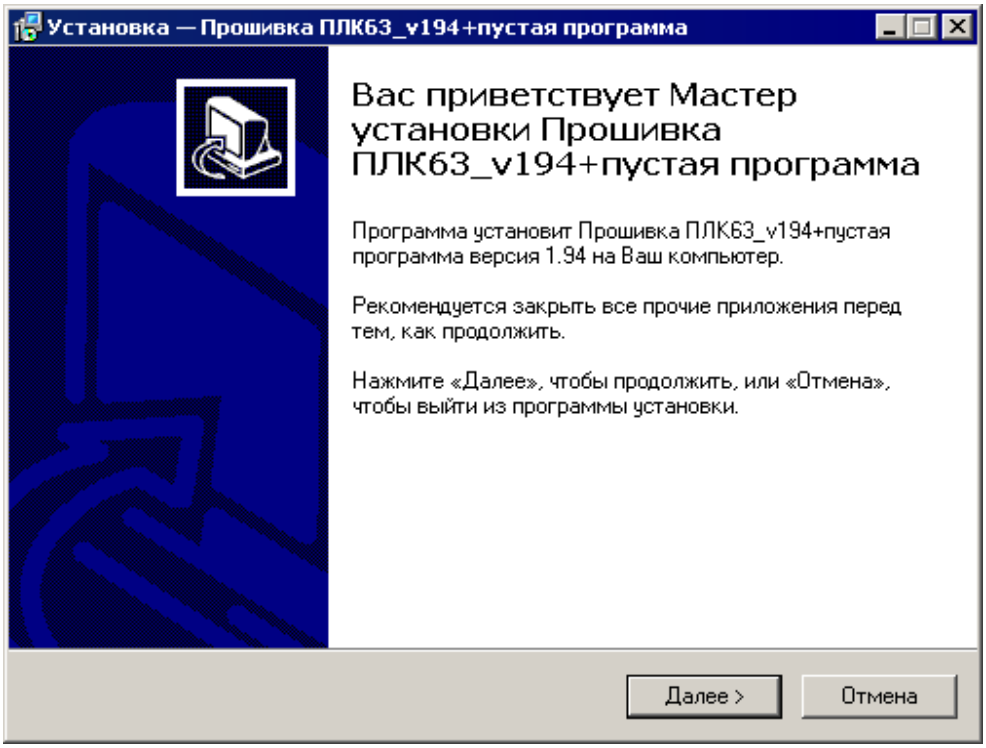


Рисунок 9.2 – Окно мастера установки программы смены прошивки (ПО микроконтроллера)

3) После завершения работы программы - инсталлятора на ПК инсталлируется программа смены прошивки (ПО микроконтроллера). На экране монитора ПК размещается ярлык запуска программы:

4) Запустить программу смены прошивки (ПО микроконтроллера). Выполняя инструкции, размещенные в окнах программы (см, например, рисунок 9.3) – выполнить процедуру смены прошивки (ПО микроконтроллера). ПО микроконтроллера будет обновлено.

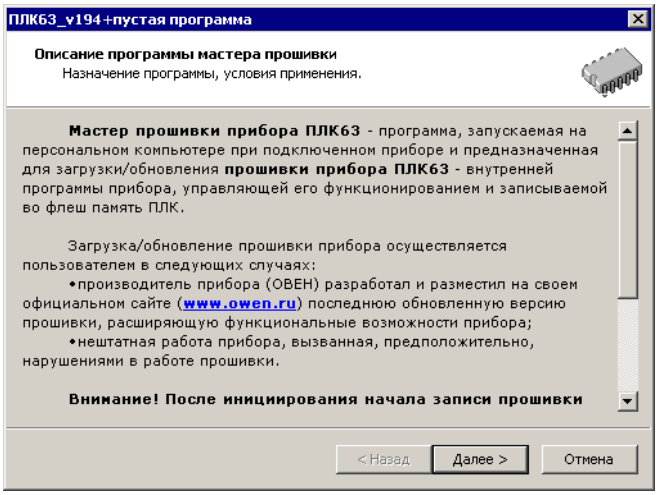


Рисунок 9.3 – Окно программы смены прошивки (ПО микроконтроллера)

Далее следует перейти к установке target-файла.

9.3 Обновление Target-файла

Для обновления Target-файла ПЛК следует:

1) Скопировать на жесткий диск ПК требуемый Target-файл для конкретного ПЛК. При выборе Target-файла следует обратить внимание на то, что имя Target-файла не полностью совпадает с наименованием контроллера: в наименовании контроллера использована кириллица (например, ПЛК63), а в названии Target-файла – латиница (например, PLC63). Для каждой модификации ПЛК в поставку включен соответствующий Target-файл. Так, для ПЛК63 на дистрибутивном диске размещены Target-файлы PLC63-l и PLC63- M.
2) Извлечь Target-файл (и входящие в его комплект файлы) из архива.
3) Удалить старую версию файла.
4) Установить требуемую версию Target-файла.
Процедуры удаления (деинсталляции) и установки (инсталляции) Target-файлов описаны в разделе 3.2 .

Приложение А. Интерфейс ПО CoDeSys

А.1 Элементы управления ПО CoDeSys

Элементы управления ПО CoDeSys приведены в таблице А.1.

Таблица А.1 – Элементы управления программой

Команда меню	Кнопка панели инструментов	Горячие клавиши	Описание команды
1	2	3	4
Файл (File)			
Создать (New)			Создает новый проект. Новый проект получает имя «Untitled».
Создать по шаблону (New from template)			Открывает окно выбора файла, в котором следует выбрать требуемый файл проекта (*.pro), который послужит шаблоном нового проекта. Новый проект получает имя «Untitled».
Открыть (Open)		<Ctrl+O>	Открывает ранее сохраненный проект. Если в момент вызова этой команды какой-то проект уже открыт и в него были внесены изменения, то программа предложит сохранить этот проект
Закрыть (Close)			Закрывает открытый в данный момент проект. Если с момента открытия в проект были внесены изменения, то программа предложит сохранить этот проект
Сохранить (Save)		<Ctrl+S>	Сохраняет файл проекта
Сохранить как… (Save as…)			Сохраняет проект или библиотеку под новым именем. При этом исходный файл не изменяется
Сохранить/отправить архив (Save/Mail Archieve…)			Создает архив проекта. Все файлы, которые используются проектом CoDeSys, сохраняются и сжимаются в файл с расширением *.zip, который удобно хранить и пересылать по электронной почте.
Печать (Print)		<Ctrl+P>	Печатает содержание активного окна
Параметры печати (Printer Setup…)			Открывает окно настройки печати
Выход (Exit)			Закрывает CoDeSys. Если в момент вызова этой команды открыт проект, то программа предложит его сохранить
Правка (Edit)			
Отменить (Undo)		<Ctrl+Z>	Отменяет последнее изменение, сделанное в открытом редакторе или в Организаторе объектов. Используя эту команду, вы можете отменить все изменения, выполненные после открытия окна
Вернуть (Redo)		<Ctrl+Y>	Возвращает последнее изменение, отмененное в открытом редакторе или в Организаторе объектов командой Undo
Cut – Вырезать		<Ctrl+X>	Перемещает выделенный элемент в буфер. При этом выделенный элемент удаляется из окна редактора
Copy – Копировать		<Ctrl+C>	Копирует выделенный элемент в буфер, содержимое окна редактора при этом не изменяется

Таблица А.1 – Продолжение

1	2	3	4
Paste – Вставить		<Ctrl+V>	Вставляет содержимое буфера, начиная с текущей позиции курсора в окне редактора. В графических редакторах команда выполняется только если содержимое буфера соответствует выбранному элементу
Delete – Удалить		<Delete>	Удаляет выбранную область, содержимое буфера при этом не изменяется
Find… – Найти		<Ctrl+F>	Находит введенный текст в активном окне редактора. Открывает диалог поиска
Find next – Найти далее		<F3>	Начинает поиск введенного текста с текущей позиции и далее
Replace – Найти и заменить			Находит заданный текст и заменяет его на введенный. Вызов команды открывает диалог поиска и замены выбранного текста
Input Assistant – Ассистент ввода			Открывает диалог выбора элемента, который можно ввести в текущей позиции. В левом столбце следует выбрать категорию элементов, в правом – требуемый элемент, а затем нажать OK
Auto Declare – Автоматическое объявление переменных			Открывает диалог объявления переменных
Next Error – Следующая ошибка			Показывает следующую ошибку, если проект скомпилирован с ошибками. При этом открывает соответствующий редактор в том месте, где допущена ошибка, а в окне сообщений отображается краткое описание ошибки
Previous Error – Предыдущая ошибка			Показывает предыдущую ошибку, если проект скомпилирован с ошибками. При этом открывает соответствующий редактор в том месте, где допущена ошибка, а в окне сообщений отображается краткое описание ошибки
Macros – Макросы			Показывает список всех определенных в проекте макрокоманд (макросов). При выполнении макроса открывается окно «Process Macro», в котором выводится имя макроса и имя активной команды
Project – Проект			
Build – Компилировать измененные POU			Компилирует только POU, которые были изменены
Rebuild all – Компилировать весь проект			Компилирует весь проект, даже если он не был изменен
Clear all – Очистить все			Стирает всю информацию о предыдущей компиляции и загрузке проекта в контроллер
Load Download-Information – Загрузить информацию о загрузке кода			Загружает информацию о загрузке кода в контроллер, если она была сохранена в директории, отличной от той, в которой находится проект
Translate into another language – Перевести текст проекта на другой язык			Переводит текст проекта на другой национальный язык (используется вспомогательный текстовый файл, созданный в CoDeSys и переведенный в текстовом редакторе на требуемый язык)
Document – Создать документ для печати			Создает версию проекта для печати

Таблица А.1 – Продолжение

1	2	3	4
Export… – Экспортировать проект			Экспортирует проект из одного инструмента МЭК программирования в другой. Можно экспортировать POU, типы данных, визуализации, описания подключенных к проекту библиотек (но не сами библиотеки) и другие ресурсы
Import – Импортировать данные из файла в проект			Импортирует в проект данные из выбранного файла
Siemens Imports – Импортировать из файлов Siemens			Импортирует переменные и POU из файлов Siemens-STEP5 и STEP7
Merge – Слияние проектов			Сливает два проекта
Compare – Сравнить			Сравнивает два проекта или разные версии одного и того же проекта
Project info – Информация о проекте			Сохраняет дополнительную информацию о проекте
Global Search – Глобальный поиск		<Ctrl+F>	Находит заданный текст в POU, типах данных или разделе глобальных переменных проекта
Global Replace – Заменить текст			Находит заданный текст в POU, типах данных или в глобальных переменных проекта и заменяет его другим
Viev Instance – Показать экземпляры функционального блока			Показывает экземпляры выбранного в организаторе объектов функционального блока. Отображает список всех экземпляров выбранного функционального блока и его реализация
Show Call Tree – Показать дерево вызовов объекта			Показывает дерево вызовов выбранного объекта в новом окне (проект должен быть скомпилирован)
Show Cross Reference – Показать адрес переменной			Открывает диалог, в котором отображены адрес, место расположения (POU, номер строки) переменной (проект должен быть скомпилирован)
Check – Семантический контроль			Команды этого меню используются для дополнительного семантического контроля (проект должен быть скомпилирован без ошибок)
Add Action – Добавить действие			Создает действие, связанное с блоком, выделенным в Организаторе объектов. При этом следует задать имя действия и язык, на котором оно будет описано
User Group Passwords – Пароли групп пользователей			Устанавливает права доступа к объекту для различных групп пользователей
Меню Insert – Вставка			
Declaration keywords – Список ключевых слов			Выводит список ключевых слов для быстрого ввода ключевых слов, допускаемых в разделе объявлений POU. После выбора ключевого слова из списка, оно будет вставлено в текущую позицию курсора
Types – Список типов			Выводит список доступных типов для их быстрого ввода
New declaration – Добавить новую переменную			Добавляет новую переменную в таблицу редактора объявлений
Extras – Опции			Команды данного пункта Меню могут меняться в зависимости от режима работы

Таблица А.1 – Продолжение

1	2	3	4
Online – Подключение к контроллеру			
Login – Подключиться к контроллеру			Устанавливает соединение ПО CoDeSys с контроллером (или запускает программу эмуляции) и включает режим «Online»
Logout – Отключить соединение с контроллером			Разрывает соединение с контроллером или заканчивает работу программы (в режиме эмуляции). Включает режим «Offline»
Download – Загрузить код проекта в контроллер			Загружает код проекта в контроллер
Run – Запустить программу в контроллере/режим эмуляции			Запускает программу на выполнение в контроллере или в режиме эмуляции
Stop – Остановить выполнение программы			Останавливает программу при ее выполнении в контроллере или в режиме эмуляции
Reset – Сброс			Заново инициализирует все переменные, за исключением VAR RETAIN. Если определены начальные значения переменных, они будут присвоены (включая VAR PERSISTENT). Прочие переменные приобретут стандартные значения по умолчанию (например, 0 для целых типов). Данный сброс аналогичен выключению и включению питания ПЛК, при работающей программе
Reset (cold) – Холодный сброс			Холодный сброс. Выполняет те же действия, что и при команде «Reset», и дополнительно выполняет инициализацию энергонезависимой области памяти RETAIN
Reset (original) – Заводской сброс			Выполняет «Reset Cold». Происходит инициализация области PERSISTENT и удаление программы пользователя. Восстанавливаются заводские настройки контроллера
Toggle Breakpoint – Установить точку останова			Устанавливает точку останова в текущей позиции активного окна. Если в этой позиции уже стоит точка останова, то она будет удалена
Breakpoint Dialog – Открыть диалог управления точками останова			Открывает диалог управления точками останова в проекте; в нем указаны все заданные точки останова
Step over – Выполнить одну инструкцию программы			Выполняет одну инструкцию программы. Если это инструкция вызова POU, то данный POU выполняется целиком, затем программа останавливается
Step in – Выполнить программу по шагам			Выполняет программу по шагам, с заходом в вызываемые блоки. Вызываемые POU открываются в отдельных окнах
Single Cycle – Выполнить один цикл			Выполняет один рабочий цикл контроллера. Команду можно повторять многократно при отслеживании работы программы по рабочим циклам
Write values – Записать значение в переменную			Перед началом рабочего цикла присваивает переменной (или нескольким переменным) заранее введенные значения

Таблица А.1 – Продолжение

1	2	3	4
Force values – Зафиксировать значение переменной			Фиксирует значения одной или нескольких переменных. Запись заданного значения осуществляется в начале и в конце каждого управляющего цикла: 1. Чтение входов, 2. Фиксация переменных, 3. Выполнение кода программы, 4. Фиксация переменных, 5. Запись выходов
Release force – Отменить фиксацию значений переменных			Отменяет фиксацию значений переменных
Write/Force-Dialog – Показать список записываемых и фиксируемых переменных			Открывает окно, содержащее таблицы записываемых (Writelist) и фиксируемых (Forcelist) переменных. В левом столбце таблиц отображаются имена переменных, в правом – их установленные значения
Show Call Stack – Показать список вызванных POU			Показывает список вызванных POU, когда программа остановлена в точке останова
Display Flow control – Показать контроль потока исполнения			Включает режим контроля потока исполнения. Если данная возможность поддерживается целевой платформой, то каждая строка или цепь программы, которая была выполнена в контроллере в предыдущем управляющем цикле, будет выделена
Simulation – Включить режим эмуляции			«Старт». Включает режим эмуляции, программа будет выполнена в ПК. Если режим эмуляции выключен, программа будет запущена в контроллере
			«Стоп». Останавливает программу при ее выполнении в контроллере или в режиме эмуляции
Communication Parameters – Параметры соединения			Выводит диалог настройки параметров связи ПК и ПЛК (если используется OPC или DDE сервер, то эти параметры можно настроить из их конфигурации)
Sourcecode download – Загрузить в контроллер исходные тексты проекта			Загружает исходные тексты проекта в контроллер (именно исходные тексты проекта – не код проекта, который создается при компиляции)
Create bootproject – Автоматически загружать код проекта при перезапуске ПЛК			Делает код проекта автоматически загружаемым при перезапуске контроллера: проект будет выполняться автоматически при перезапуске ПЛК
Write file to PLC – Записать файл в контроллер			Записывает в ПЛК выбранный файл (любого типа); размер файла ограничивается размером карты памяти контроллера
Read file from PLC – Считать файл из контроллера			Считывает ранее сохраненный в контроллере файл и сохраняет его в указанную директорию на ПК
Window – Работа с окнами			
Tile Horizontal – Упорядочить окна по горизонтали			Упорядочивает размещение окон по горизонтали так, чтобы они не перекрывали друг друга и полностью занимали рабочую область
Tile Vertical – Упорядочить окна по вертикали			Упорядочивает размещение окон по вертикали так, чтобы они не перекрывали друг друга и полностью занимали рабочую область

Таблица А.1 – Окончание

1	2	3	4
Cascade – Упорядочить окна каскадом			Упорядочивает окна каскадом – каждое следующее поверх остальных
Arrange Symbols – Выстроить свернутые окна			Выстраивает свернутые окна в ряд в нижней части Рабочего окна
Close All – Закрыть все окна			Закрывает все окна
Messages – Открыть окно сообщений			Открывает окно сообщений, которое содержит информацию о предыдущей компиляции, проверке или сравнении
Library Manager – Открыть менеджер библиотек			Открывает окно менеджера библиотек.
Log – Открыть бортжурнал			Открывает «бортжурнал» – детальный протокол последовательности действий, которые были выполнены во время «Online»-сессии. Бортжурнал записывается в двоичный файл формата *.log
Help – Помощь			
Contents – Содержание справочной документации			Открывает окно системы оперативной помощи
Search – Осуществить поиск			Переход к контекстному поиску по текстам оперативной помощи
About – Показать информацию о программе			Открывает окно с информацией о программе CoDeSys

А.2 Основные режимы (Редакторы) ПО CoDeSys

CoDeSys предоставляет встроенные специализированные редакторы для всех пяти языков МЭК 61131-3 и дополнительный CFC редактор:

- Список Инструкций (IL);
- Функциональные блоковые диаграммы (FBD);
- Релейно-контактные схемы (LD);
- Структурированный текст (ST);
- Последовательные функциональные схемы (SFC):
  - мониторинг времени исполнения шагов;
  - автоматический анализатор причин ошибок;
  - набор управляющих флагов: сброс, разрешение мониторинга, фиксация переходов и т.д.
- Непрерывные функциональные диаграммы (CFC):
  - автоматическая расстановка и соединение;
  - макро опция для структурирования больших диаграмм.

Два специальных редактора управляют прикладной средой исполнения:

- Конфигуратор задач задает:
  - циклические задачи и задачи, исполняемые по событиям;
  - параметры сторожевого таймера;
  - настройку событий.
- Конфигуратор ввода-вывода обеспечивает:
  - Profibus конфигурирование на основе GSD файлов;
  - CANopen конфигурирование на основе EDS файлов;
  - ASI конфигурирование.

Приложение Б. Перечень параметров конфигурации ПЛК63

Перечень параметров конфигурации ПЛК приведен в таблице Б.1.

Таблица Б.1 – Перечень параметров конфигурации ПЛК

Путь и Название параметра1	Адрес modbus (dec)	Адрес mod-bus2	Имя в протоколе ОВЕН	Тип	Минимум	Максимум	Заводск. знач.	Комментарий
1	2	3	4	5	6	7	8	9
Конфигурация | Дискретные Вх. | Сост.Дискр. Вх.	288	4096	r.Cn	byte	0000000 0b	1111111 1b	--	Оперативный параметр. Тип отображения – бинарный. Левый разряд соответствует Входу С1.
Конфигурация | Дискретные Вх. | Логика Дискр.Вх	289	-	inv.D	byte	0000000 0b	1111111 1b	0000000 0b	Тип отображения – бинарный. Левый разряд соответствует Входу С1. Единица в соответствующем дискретному входу разряде позволяет воспринимать замкнутый контакт как 0, разомкнутый – как 1, ноль в соответствующем дискретному входу разряде позволяет воспринимать замкнутый контакт как 1, разомкнутый – как 0.
Конфигурация | Дискретные Вх. | Пост.Ф.Дреб Конт	290	-	Tin.F	byte	0	255	10	Постоянная времени низкочастотного фильтра. Задается в мс.
Конфигурация | ВУ | Сост.ВУ{N}3	{0, 1} … {10, 11}	{4184, 4185} … {4194, 4195}	r.oe	Float	0.000	1.000	--	Оперативный параметр, доступный на запись, задаётся в долях единицы с точностью 0,001. Определяет уровень выходного сигнала.

[1] В Конфигураторе параметры каждого из выходов объединены в отдельную ветку, в приборе же они идут сплошным списком.
[2] копия области %I, %Q (dec).
[3] Здесь и ниже {N} принимает значения 1…6 для выходов 1…6 соответственно.

Таблица Б.1 – Продолжение

1	2	3	4	5	6	7	8	9
Конфигурация | ВУ | Период ШИМ ВУ{N}	291…296	-	thpd	Word	2	65000	1000	Для дискретных ВУ* задаёт период выходного ШИМ-сигнала. Задаётся в мс. Для аналоговых ВУ не используется.
Конфигурация | ВУ | Мин.имп.ШИМ ВУ{N}	297…302	-	t.L	Word	10	5000	50	Для дискретных ВУ задаёт минимальную длительность импульса ШИМ-сигнала. Если Сост.ВУ{N}*Период ШИМ.ВУ{N} < < Мин.имп.ШИМ{N}, то Сост.ВУ{N} = 0. Задаётся в единицах мс. Для аналоговых ВУ не используется.
Конфигурация | ВУ | Безоп.сост.{N}	{12,13} … {22,23}	-	O.ALr	Float	0.000	1.000	0.000	Задаётся в долях единицы с точностью 0,001. Определяет состояние ВУ (аналогового или дискретного), когда основная программа не функционирует (загрузка прибора, калибровка ЦАП, «зависание» и т.д.)
Аналоговые Вх. | Коррекция ХС	303	-	Cj-.c	Enum			Включить	Включает коррекцию холодного спая для термопар.

Таблица Б.1 – Продолжение

1	2	3	4	5	6	7	8	9
Аналоговые Вх. | Вход {N}4	{24,25} … {38,39}	{4098, 4099} {4102, 4103} {4106, 4107} {4110, 4111} {4114, 4115} {4118, 4119} {4122, 4123} {4126, 4127}	read	float				Оперативный параметр, доступный только для чтения. Параметр показывает измеренное соответствующим входом значение.
Аналоговые Вх. | Время измерений на Входе {N}	-	{4100, 4101} {4104, 4105} {4108, 4109} {4112, 4113} {4116, 4117} {4120, 4121} {4124, 4125} {4128, 4129}	-	-	-	-	-	-
Аналоговые Вх. | Тип датчика {N}	304…311	-	In-t	enum	Нет датчика	Рез.0..2 кОм	Нет датчика	Определяет тип подключённого к соответствующему входу датчика
Аналоговые Вх. | Пост.Фильтра {N}	312…319	-	In.fd	short	0	1800	5	Определяет постоянную времени НЧ-фильтра
Аналоговые Вх. | Полоса Фильтра{N}	{40,41} … {54,55}	-	In.fg	float	0	9990	10	Определяет уровень пикового фильтра
ВУ МР1 | Сост.ВУ МР1	320	4196	r.oe.s	byte				Оперативный параметр, определяющий состояние подключённого МР1. Тип отображения – бинарный.

[4] Здесь и ниже {N} принимает значения 1…8 для входов 1…8 соответственно.

Таблица Б.1 – Продолжение

1	2	3	4	5	6	7	8	9
Доп.пар-ры | Ярк.подсв.ЖКИ	321	4198	Ind.i	byte	0	50	45	Яркость подсветки ЖКИ. Задается в условных единицах от 0 до 50. 0 соответствует полностью выключенной подсветке, 50 – максимальной яркости.
Доп.пар-ры | Контраст ЖКИ	322	4199	Ind.c	byte	5	32	25	Определяет контрастность отображаемой на ЖКИ информации. Для получения максимальной контрастности значение подбирается индивидуально для каждого прибора в зависимости от температуры, старения и др. параметров.
Доп.пар-ры | Сост.клавиатуры	323	4130	kbrd	byte				Оперативный параметр, соответствующий сумме кодов нажатых клавиш (см. п. 7.3.5).
Доп.пар-ры | Звук кнопок	324		beep	bool	Выключить	Включить	Включить	
Доп.пар-ры | Время и Дата	{56,57}	4200, 4201	rtc	DateTime				Оперативный, поясные дата и время контроллера без учёта летнего/зимнего времени с точностью до секунды.
ВерсииПрошивок | Имя устройства	{58…65}	{4202 … 4209}	Dev	string				Имя устройства. Не изменяемое значение.
ВерсииПрошивок | Версия прошивки	{66…73}	{4132 … 4139}	ver	String				Версия прошивки[5]
ВерсииПрошивок | ВерсияПрограммы	{74…81}	{4210 … 4217}	p.ver	String				Версия прошивки программы CodeSys
Настр.RS-485 | Скорость	325		bPS	enum			115200	Параметры данной ветви не влияют на параметры связи по RS-232

[5] в формате: Версия_прошивки_ядра_ЦП(string); Версия_прошивки_вспомогательного_процессора(hex); Версия_прошивки_измерительного_процессора(hex)

Таблица Б.1 – Окончание

1	2	3	4	5	6	7	8	9
Настр.RS-485 | Длина слова	326		Len	enum	5	8	8	
Настр.RS-485 | Четность	327		prty	enum			No Parity	
Настр.RS-485 | Стоп биты	328		sbit	enum	1	2	1	
Настр.RS-485 | Длина адреса	329		a.len	bool	8	11	8	
Настр.RS-485 | Адрес прибора	330		Addr	short	0	2047	16	
Настр.RS-485 | Задержка ответа	331		Rs.dl	byte	0	50	5	
Невидимое дерево (невидимо на ЖКИ прибора и по сети, но параметры доступны по протоколам «ОВЕН» и «Modbus»)								
Код Посл.Ош.	10303		r.src	short				Параметр ядра, тип retain, код последней ошибки прибора.

Приложение В. Список EnumP

Список EnumP приведен в таблице В.1.

Таблица В.1 – Список EnumP

№№ п/п	Enum*	№№ п/п	Enum*	№№ п/п	Enum*	№№ п/п	Enum*
1	2	3	4	5	6	7	8
0	"0"	60	"ТС 1000Н 1.617"	109	"июль"	160	""
1	"1"	61	"ТП ТХК L"	110	"август"	161	"1"
2	"2"	62	"ТП ТХА К"	111	"сентябрь"	162	"1,5"
3	"3"	63	"Ток 4..20 мА"	112	"октябрь"	163	"2"
4	"4"	64	"Ток 0..20 мА"	113	"ноябрь"	164	""
5	"5"	65	"Ток 0..5 мА"	114	"декабрь"	165	"Нет АНР"
6	"6"	66	"Напряж. 0..1В"	115	""	166	"АНР ТобР"
7	"7"	67	"Рез.40..900 Ом"	116	"нет"	167	"АНР ТобД"
8	"8"	68	"Рез.0,04..2 кОм"	117	"день"	168	"АНР Тпр+"
9	"9"	69	" НЕТ ДАТЧИ-КА"	118	"ночь"	169	"АНР Тпр-"
10	"10"	70	"115200"	119	""	170	"Выкл."
11	""	71	"57600"	120	"нет "	171	"Вкл."
12	"%"	72	"38400"	121	"утро"	172	"Обрыв"
13	"мкА"	73	"28800"	122	"вечер"	173	"К.З."
14	"мА"	74	"19200"	123	""	174	"Ноль"
15	"А"	75	"14400"	124	"нет"	175	"Фаза"
16	"мкВ"	76	"9600"	125	"утро"	176	"Земля"
17	"мВ"	77	"4800"	126	"день"	177	"Авария"
18	"В"	78	"2400"	127	"вечер"	178	"Работа"
19	"Вт"	79	"1200"	128	"ночь"	179	"Останов"
20	"бит"	80	""	129	""	180	"Прогрев"
21	"байт"	81	"no"	130	"нет"	181	Выдержка",
22	"мкС"	82	"even"	131	"весна"	182	"Охлаждение"
23	"мСек"	83	"odd"	132	"осень"	183	"Перегрев"
24	"Сек"	84	"7 бит"	133	""	184	"Норма"
25	"мин"	85	"8 бит"	134	"нет"	185	"Переохлажде-ние"
26	" час "	86	"11 бит"	135	"зима"	186	"Наладка"
27	""	87	""	136	"лето"	187	"Ручной"
28	"С`"	88	"Рег Тприт +"	137	"нет "	188	"Полуавтомат"
29	"Па"	89	"Рег Тприт -"	138	"зима"	189	"ПИД"
30	"кПа"	90	"Рег обр Р"	139	"весна"	190	"ON/OFF"
31	""	91	"Рег обр Д"	140	"лето"	191	"Задвижка"
32	"No"	92	""	141	"осень"	192	"Останов"
33	"Yes"	93	"нет"	142	"нет"	193	"Запуск"
34	"Off"	94	"понедельник"	143	"зима"	194	"Деж.Лето"
35	"On"	95	"вторник"	144	"лето"	195	"Охлажден"
36	"False"	96	"среда"	145	""	196	"Вентиляц"
37	"True"	97	"четверг"	146	""	197	"Деж.Зима"
38	"Нет"	98	"пятница"	147	""	198	"Прогрев"
39	"Да"	99	"суббота"	148	""	199	"Нагрев"
40	"Отклонить"	100	"воскресенье"	149	""	200	"Обратная"
41	"Принять"	101	""	150	""	201	"Замёрз"

Таблица В.1 – окончание

1	2	3	4	5	6	7	8
42	"Выключить"	102	"нет"	151	"РЕЛЕ"	202	"АНР Тпр Л"
43	"Включить"	103	"январь"	152	"ЦАП"	203	"АНР Тпр З"
44	"Запретить"	104	"февраль"	153	""	204	"АНР Тобр Д"
45	"Разрешить"	105	"март"	154	"Even"	205	"Фреоновый"
46	"Вкл. на время"	106	"апрель"	155	"Odd"	206	"Водяной"
47	" НЕТ ДАТЧИКА"	107	"май"	156	"Space"	207	"Отключён"
48	" ТС 50М 1.426"	108	"июнь"	157	"Mark"	208	"Водяной"
49	" ТС 50М 1.428"			158	"No Parity"	209	"Электрический"
50	" ТС 50П 1.385"			159	"Multy Drop"	210	"Отключён"
51	" ТС 50П 1.391"						
52	" ТС 100М 1.426"						
53	" ТС 100М 1.428"						
54	" ТС 100П 1.385"						
55	" ТС 100П 1.391"						
56	" ТС 500П 1.385"						
57	" ТС 500П 1.391"						
58	"ТС 1000П 1.385"						
59	"ТС 1000П 1.391"						
* – отображается без кавычек							

Приложение Г. Перечень поддерживаемых датчиков

Перечень поддерживаемых датчиков приведен в таблице Г.1.

Таблица Г.1 – Используемые на входе первичные преобразователи (датчики)

Наименование	Диапазон измерений, оС	Значение единицы младшего разряда, оС	Предел основной приведенной погрешности, %
Термометры сопротивления по ГОСТ Р 8.625-2006 или термопреобразователи сопротивления по ГОСТ 6651-94			
Pt 50 (α=0,00385 °C^{-1})	–200…+750	0,1	±0,25
50 П (α=0,00391 °C^{-1})	–200…+750	0,1	
Cu 50 (α=0,00426 °C^{-1})	–50…+200	0,1	
50 М (α=0,00428 °C^{-1})	–190…+200	0,1	
Pt 100 (α=0,00385 °C^{-1})	–200…+750	0,1	
100 П (α=0,00391 °C^{-1})	–200…+750	0,1	
Cu 100 (α=0,00426 °C^{-1})	–50…+200	0,1	
100 М (α=0,00428 °C^{-1})	–190…+200	0,1	
Pt 500 (α=0,00385 °C^{-1})	–200…+650	0,1	
500 П (α=0,00391 °C^{-1})	–200…+650	0,1	
Pt 1000 (α=0,00385 °C^{-1})	–200...+650	0,1	
1000 П (α=0,00391 °C^{-1})	–200...+650	0,1	
Ni 1000 (α=0,00617 °C^{-1})	–60...+180	0,1	
Термоэлектрические преобразователи по ГОСТ Р 8.585-2001			
ТХК (L)	–200…+800	0,1	±0,5
ТХА (K)	–200…+1300	0,1	(±0,25) 2)
Универсальные датчики			
– резистивный (40... 900 Ом)	0...100 %	0,1 %	±0,25
– резистивный (0,04... 2 кОм)	0...100 %	0,1 %	
– токовый 0... 20 мА	0...100 %	0,1 %	
– токовый 4... 20 мА	0...100 %	0,1 %	
– токовый 0... 5 мА	0...100 %	0,1 %	
– напряжения 0…1 В	0...100 %	0,1 %	

Примечания.

1) α – температурный коэффициент термометра сопротивления – отношение разницы сопротивлений датчика, измеренных при температуре 100 и 0 °С, к его сопротивлению, измеренному при 0 °С (R_0), деленное на 100 °С и округленное до пятого знака после запятой.
2) Основная приведенная погрешность без КХС.
3) Для работы с прибором могут быть использованы только изолированные термоэлектрические преобразователи с незаземленными рабочими спаями.

Приложение Е. Использование OPC-сервера

Приложение предназначено для ознакомления пользователя с технологией подключения ПЛК к ПК через OPC-серверы – разработанные компанией 3S-Software и технологией подключения приборов, разработанных компанией ОВЕН, к ПК через OPC-драйверы.

Для работы оборудования с современными SCADA системами, поддерживающими спецификацию OPC DA, требуются OPC-драйверы (OPC–сервер) для приборов, реализующие спецификацию Data Access (DA).

Прочесть и записать данные может пользовательская программа на языке, полноценно поддерживающем COM технологию Microsoft (Visual Basic, C++, Java, Delphi и т.д.). Получение данных возможно также из приложений, поддерживающих доступ к COM объектам. Например, приложений пакета MS Office, что позволяет пользователю получить, например, в таблице Excel набор технологических параметров, изменяющихся в реальном масштабе времени.

Е.1 Использование OPC-сервера 3S-Software

OPC-сервер, разработанный компанией 3S-Software, предназначен для подключения ПЛК к системам SCADA. OPC-сервер соответствует спецификации OPC DA 2.0, в частности, просмотр списка имен переменных подключенного ПЛК.

Для подключения ПЛК к ПК следует:

1) Выбором команды **Пуск | Программы | 3S Software | Communication | CoDeSys OPC Configurator** – запустить ПО «OPC Configurator».

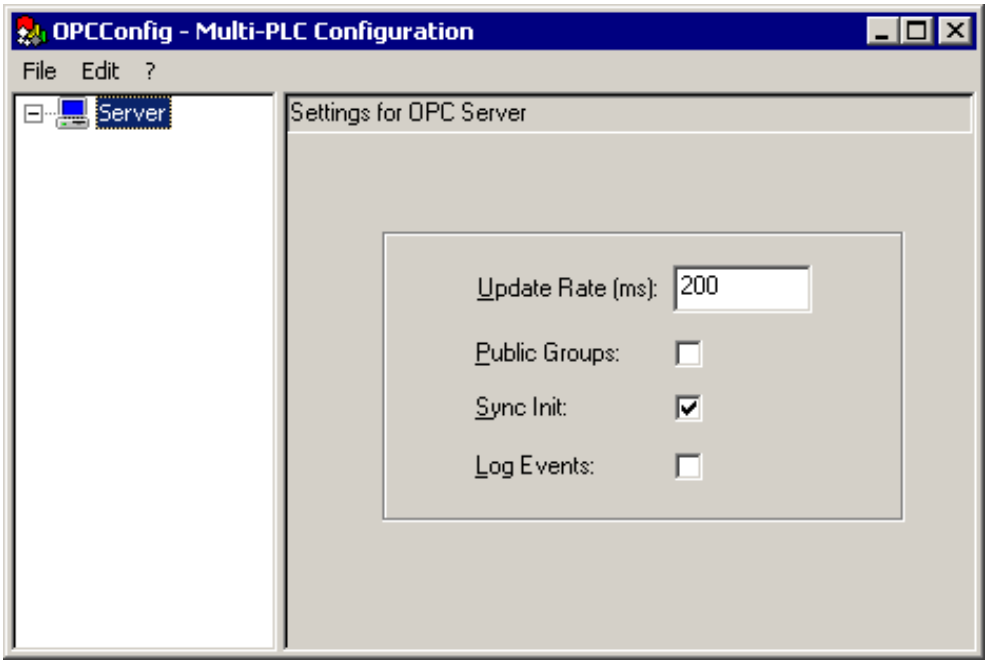


Рисунок Е.1 – Окно OPC Configurator

2) В открывшемся окне «**OPC Config – Multi-PLC Configuration»** (рисунок Е.1) в левой части окна отображается иерархический список, исходно содержащий одну строку: «Server».
Выделить строку «Server» и в поле параметров (в правой части окна) – установить время обновления данных (в миллисекундах), введя требуемое значение в поле «**Update Rate (ms)».**

3) В контекстном меню строки «Server» выбрать команду «**Append**

PLC».
В иерархический список в левой части окна добавляются строки «PLC1» и «**Connection**».

4) В иерархическом списке (в левой части окна) – выбрать пункт «Connection».
В поле параметров (в правой части окна) – нажать кнопку «Edit»

5) В открывшемся окне «Communication Parameters» – установить параметры подключения ПЛК, см. рисунок Е.2.

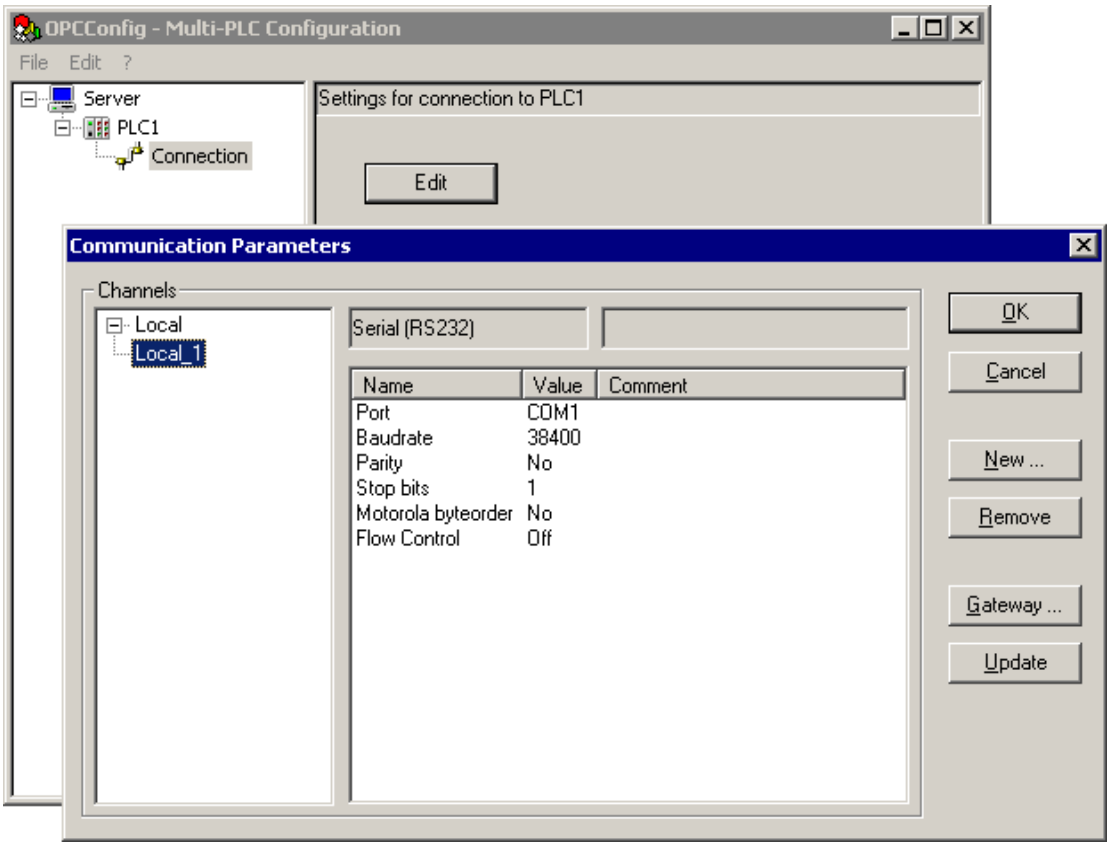


Рисунок Е.2 – Окно «Communication Parameters»

6) Нажать кнопку «**ОК**» окна «**Communication Parameters**», в открывшемся окне запроса подтверждения операции («Save changes in Multi-PLC Configuration») – нажать кнопку и»Да» для подтверждения произведенных настроек.

После этого ПО «**OPC-сервер»** сконфигурировано и готово к работе под управлением **SCADA-системы**.

Е.2.1 Установка и использование OPC-драйверов фирмы ОВЕН

OPC-драйверы, разработанные компанией ОВЕН, предназначены для подключения приборов фирмы ОВЕН к системам SCADA.

OPC-драйверы, разработанные компанией ОВЕН, работают с ПЛК63 и ПЛК73 только по протоколу Modbus.

Для установки модулей OWEN-RS232 и OWEN-RS485 требуется:

1) Запустить программу-инсталлятор (файл OwenOPC-setup.exe).
2) В открывшемся окне программы-инсталлятора – нажать кнопку «Далее». В последовательно открывающихся окнах мастера установки – выполнять инструкции, отображаемые в окне (см. рисунок Е.3).

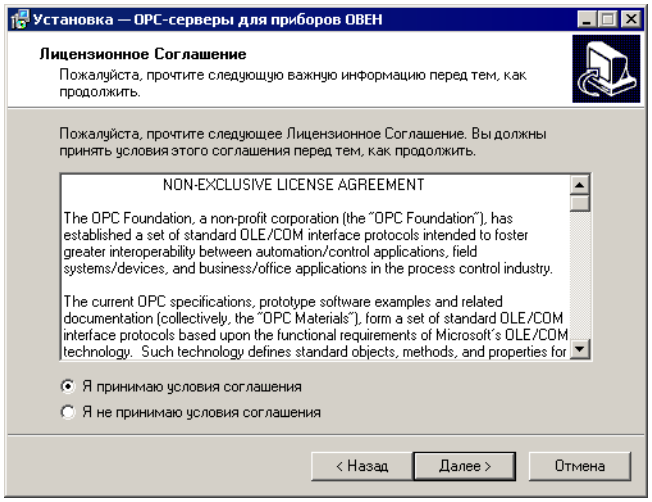


Рисунок Е.3 – Окно 1 программы – инсталлятора OPC-сервера для приборов ОВЕН

Е.2.1 Использование OPC-драйверов «ОВЕН»

Для подключения прибора следует применять драйвер «OPC – сервер для приборов MODBUS».

Для этого:

1) Запустить программу-драйвер: выбрать команду «Пуск | Программы | Owen | OPC drivers | OPC-сервер для приборов с интерфейсом RS-485 и протоколом Modbus».
2) В открывшемся окне – задать конфигурацию порта (см. рисунок Е.4).

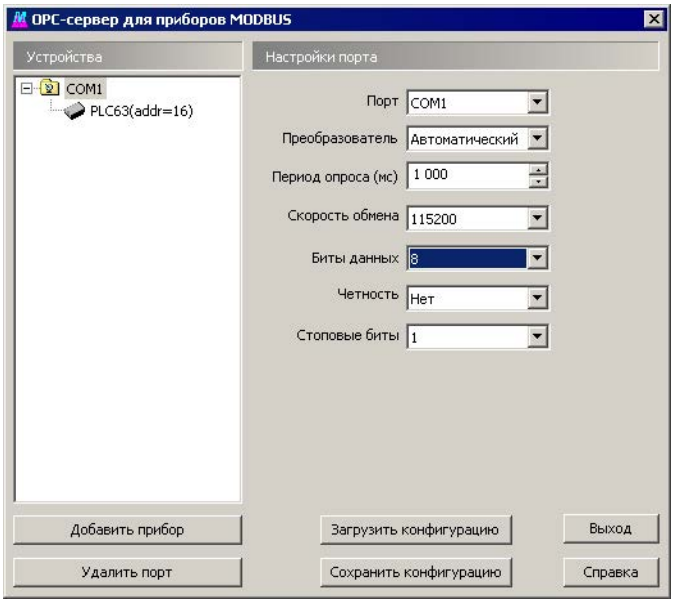


Рисунок Е.4 – Окно «OPC-сервер для приборов Modbus». Конфигурирование порта

3) Нажать кнопку «Добавить прибор» или выбрать команду «Добавить прибор» контекстного меню области «Устройства».
   В открывшемся окне – задать конфигурацию прибора (см. рисунок Е.5).

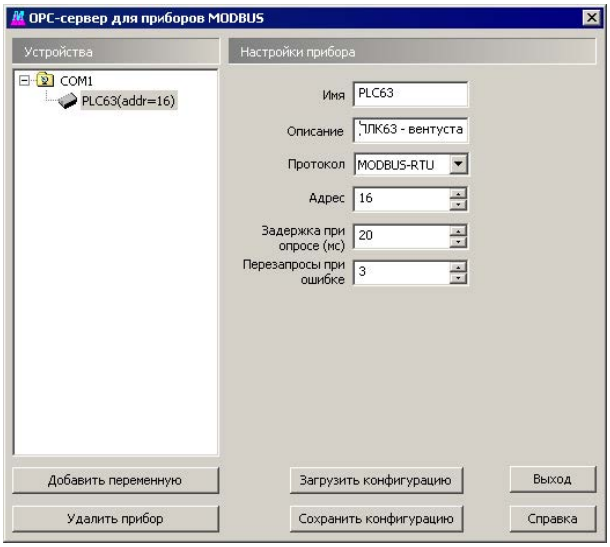


Рисунок Е.5 – Окно «OPC-сервер для приборов Modbus». Конфигурирование прибора

4) Нажать кнопку «Добавить переменную» или выбрать команду «Добавить переменную» контекстного меню области «Устройства».
В открывшемся окне – задать параметры переменной (см. рисунок Е.6). Параметры переменной можно взять из конфигурации ПЛК.
Задавая параметры переменной, следует обратить внимание на то, что адрес переменной в OPC – задается в шестнадцатиричной системе, а в конфигурации ПЛК (PLC Configuration) – в десятичной.

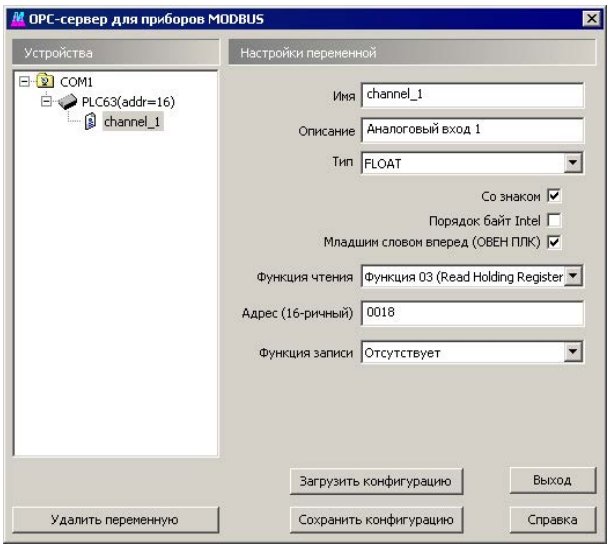


Рисунок Е.6 – Окно «OPC-сервер для приборов Modbus». Параметры переменной

5) Для сохранения заданной конфигурации – нажать кнопку «Сохранить конфигурацию».
6) При возникновении вопросов по конфигурированию драйвера OPC следует обратиться к справочной информации; для вызова справочной системы следует нажать кнопку «Справка».

Приложение Ж. Отображаемые на ЖКИ символы и их коды

При необходимости функции «ShowDint», «ShowReal» и «ShowString» библиотеки Work_Mode позволяют вывести на заданную позицию (или позиции) ЖКИ требуемые специализированные символы (см. п. 8.4.1.1).

Доступные для отображения символы и их коды приведены в таблице Ж.1.

В верхней строке таблицы приведены значения старшего бита кода символа, в первом левом столбце приведены значения младшего бита кода. В клетках таблицы приведены символы, отображаемые на ЖКИ при заданном значении кода. В рамке приведено такое изображение символа, каково оно на ЖКИ; справа от него приведено буквенно-цифровое обозначение символа (если таковое существует).

Например, строка 1, столбец С. В ячейке на их пересечении отображена буква Б. Это значит, что символу с кодом С1 соответствует на дисплее символ «Б».

Таблица Ж.1 – Отображаемые на ЖКИ символы и их коды

Младший бит кода символа	Старший бит кода символа															
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0		0	@	P	`	p					А	Р	а	р
1	!	1	A	Q	a	q					Б	С	б	с
2	"	2	B	R	b	r					В	Т	в	т
3	#	3	C	S	c	s					Г	У	г	у
4	$	4	D	T	d	t					Д	Ф	д	ф
5	%	5	E	U	e	u					Е	Х	е	х
6	&	6	F	V	f	v					Ж	Ц	ж	ц
7	'	7	G	W	g	w					З	Ч	з	ч
8	(	8	H	X	h	x			Ё	ё	И	Ш	и	ш
9	)	9	I	Y	i	y					Й	Щ	й	щ
A	*	:	J	Z	j	z					К	Ъ	к	ъ
B	+	;	K	[	k						Л	Ы	л	ы
C	,	<	L	¥	l			¼			М	Ь	м	ь
D	—	=	M	]	m				§		Н	Э	н	э
E	.	>	N	^	n			½			О	Ю	о	ю
F	/	?	O	_	o			¾			П	Я	п	я

Приложение Н. Нормативно-справочная документация

Перечень нормативно-справочной и эксплуатационной документации, использованной в данном документе, приведен в таблице Н.1.

Таблица Н.1 – Нормативно-справочная и эксплуатационная документация

№	Наименование	Наименование файла и размещение его на дистрибутивном диске	Аннотация
1	«Первые шаги в CoDeSys»	First Steps with CoDeSys RU.pdf	
2	«Руководство пользователя по программированию ПЛК в CoDeSys 2.3»	CoDeSys_V23_RU.pdf	
4	«Визуализация CoDeSys. Дополнение к руководству пользователя по программированию ПЛК в CoDeSys 2.3»	CoDeSys_Visu_V23_RU.pdf	
5	Руководство по эксплуатации ПЛК63		
6	Руководство по эксплуатации ПЛК73		
7	Библиотеки Ind_Mode и Work_Mode		
8	Библиотека ModBus.lib		
9	Библиотека owen_fb.lib		
10	Библиотека OwenNet.lib		
11	Библиотека pid_reg2.lib		
12	Библиотека ARM7_specific.lib		
13	Библиотека sound.lib		

Лист регистрации изменений

Версия документа	Дата выпуска	Содержание изменений
1.0	15.01.2010	Включает содержание следующих документов из дистрибутивного пакета: 1) Конфигурирование области ввода-вывода ПЛК63 v.8.pdf 2) Ограничения ПЛК63.pdf 3) Порядок программирования ПЛК63_РП.pdf 4) Работа с ЖКИ описание библиотек ФБ ПЛК63_v7.pdf 5) OPC-серверы
1.1	20.03.2010	Сокращены описания библиотек Ind_Mode.lib и Work_Mode.lib. Дана ссылка на документ [9] Исключен Ethernet и TCP/IP
	28.04.2010	Внесены исправления по замечаниям А.Кулешова
1.2	31.05.2010	Внесены исправления по замечаниям и примерам А.Гайворонского
1.3	07.06.2010	Добавлены приложение Ж и п. 8.4.1.1 (отображение символов на ЖКИ)
1.3.1	07.06.2010	Внесены исправления по замечаниям А.Гайворонского
1.3.2	09.06.2010	Дополнены и исправлены ссылки, исправлен формат.
1.4	09.06.2010	Приложение Д (Коды ошибок) перенесено в РЭ