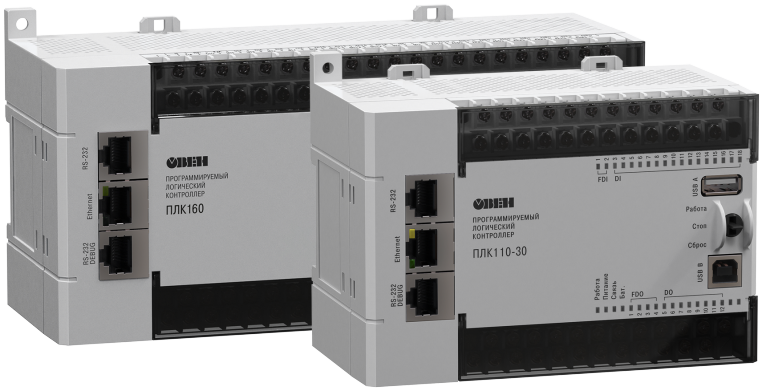
ПЛК1xx [M02]

Программирование в CODESYS



Руководство пользователя

03.2022
версия 1.33

Содержание

Введение

В настоящем руководстве изложены основы процедуры создания пользовательской программы и настройки для программируемых логических контроллеров ПЛК110 [M02] и ПЛК160 [M02] (далее по тексту именуемых «контроллер» или «ПЛК») с помощью ПО CODESYS v.2.3 (далее по тексту – CODESYS).

Полная информация о создании пользовательской программы содержится в документе «Руководство пользователя по программированию ПЛК в CoDeSys V2.3», доступном на сайте owen.ru в разделе Codesys V2.

При первом прочтении настоящего руководства рекомендуется сначала ознакомиться с разделом 6 и предыдущими. В дальнейшем рекомендуется обращаться к последующим разделам документа.

Информация по установке, эксплуатации, техническому обслуживанию и устранению ошибок работы контроллеров содержится в руководствах по эксплуатации на соответствующие ПЛК, доступные на сайте owen.ru.

Персонал, программирующий и настраивающий ПЛК, обязан владеть:

- приемами работы с графическим интерфейсом операционной системы и ПО;
- методикой программирования ПЛК с использованием CODESYS в объеме, изложенном в документе «Руководство пользователя по программированию ПЛК в CoDeSys V2.3»;
- методикой эксплуатации ПЛК в объеме, изложенном в руководстве по эксплуатации на соответствующий ПЛК.

Предупреждающие сообщения

В данном руководстве применяются следующие предупреждения:

ВНИМАНИЕ
Ключевое слово ВНИМАНИЕ сообщает о критически важной информации, на которую рекомендуется обратить особое внимание.

ПРЕДУПРЕЖДЕНИЕ
Ключевое слово ПРЕДУПРЕЖДЕНИЕ сообщает о важной информации, на которую рекомендуется обратить внимание.

ПРИМЕЧАНИЕ
Ключевое слово ПРИМЕЧАНИЕ обращает внимание на полезные советы и рекомендации, а также информацию для эффективной и безаварийной работы оборудования.

Используемые термины и аббревиатуры

CODESYS (Controller Development System) – программное обеспечение, специализированная среда программирования логических контроллеров. Торговой марка компании 3S-Smart Software Solutions GmbH.

OPC (OLE for Process Control) – открытый для использования набор спецификаций, разработанный организацией OPC Foundation на основе технологий Microsoft COM/DCOM.

OPC DA – спецификация Data Access (DA) OPC, которая позволяет читать и записывать данные в прибор, организовывать подписку на данные и передавать клиенту уведомление об обновлении данных.

Retain-память – энергонезависимая память для хранения значений Retain-переменных пользовательской программы.

Retain-переменная – переменная пользовательской программы, значение которой сохраняется в случае выключения питания контроллера.

SCADA (Supervisory Control And Data Acquisition) – программное обеспечение для получения и отображения данных в удобном для пользователя виде, с возможностью управления.

ЛКМ/ПКМ – левая/правая кнопка мыши.

ПК – персональный компьютер.

ПЛК – программируемый логический контроллер.

ПО – программное обеспечение.

Проект – пользовательская программа ПЛК, разрабатываемая в CODESYS. После отладки и загрузки в контроллер обеспечивает правильную работу контроллера.

ОЗУ – оперативное запоминающее устройство.

Файл настроек целевой платформы (target-файл) – файл, поставляемый производителем ПЛК и описывающий аппаратные и программные особенности конкретного ПЛК. Файл обеспечивает корректное взаимодействие CODESYS и ПЛК.

ШИМ – широтно-импульсная модуляция.

ПРИМЕЧАНИЕ
Терминология, используемая в интерфейсах и документации CODESYS, специфична и не всегда соответствует требованиям стандартов ЕСПД. Например, режимы редактирования текстов программ именуются «редакторы». Режимы редактирования объектов ПО, отнесенных в интерфейсе CODESYS к «ресурсам», именуются «ресурсами» (например, режим редактирования конфигурации ПЛК – «ресурсом Конфигурация ПЛК») и т. п. Описания режимов работы ПО проиллюстрированы и достаточно подробны, чтобы различие в терминологии не могло повлиять на понимание текста.

1 Требования к ПК

Минимальные системные требования для ПК:

- Pentium IV, 2 ГГц;
- 512 Мб ОЗУ (рекомендуется 1024 Мб);
- 500 Мб жесткий диск;
- CD-ROM привод;
- интерфейсы RS-232, Ethernet или USB для подключения ПЛК.
- ОС Windows XP, 7 (Service Pack 1 или выше), 8, 10 (32/64 Bit).

Для программирования ПЛК подключается к ПК с помощью кабеля КС14 из комплекта поставки ПЛК (про подключение с помощью других интерфейсов см. раздел 6.6).

Подключение к ПК описывается в *руководстве по эксплуатации* на соответствующий контроллер.

2 Ограничение контроллера по размеру 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 и программными ограничениями контроллера. Тип лицензии указывается в маркировке конкретного ПЛК в последнем знаке обозначения:

- **L** – объем памяти ввода-вывода контроллера ограничен 360 байтами*: 122 байта отводятся для памяти ввода (%I), 234 байта отводятся для памяти вывода (%Q) и оставшиеся 4 байта – под специальную память (%M).
- **M** – ограничений со стороны лицензии в размере памяти нет, но есть программное ограничение в 100 Кбайт. По умолчанию суммарный объем памяти ввода (%I) и вывода (%Q) установлен равным 16 Кбайт. Этого достаточно для большинства задач, но этот объем может быть увеличен до 32 Кбайт (на вкладке «Распределение памяти» окна «Настройки целевой платформы» в строках «Входы» и «Выходы», см. рисунок 6.4).

ПРИМЕЧАНИЕ

* Ограничение до 360 байт распространяется только на размер памяти области ввода-вывода, количество внутренних переменных программы контроллера ограничивается только количеством свободной оперативной памяти.

Пример

Контроллер **ПЛК110-24.60.Р-L [M02]** имеет ограничение объема памяти ввода-вывода размером 360 байт.

Контроллер **ПЛК110-24.60.Р-M [M02]** имеет ограничение объема памяти ввода-вывода 100 Кбайт.

Для расчета необходимого объема памяти ввода-вывода и выбора требуемого типа лицензии можно воспользоваться методикой, изложенной в разделе 6.7.

Задание конфигурации памяти ввода-вывода описано в разделе 10.1.

3 Порядок программирования

ПРИМЕЧАНИЕ
ПЛК рекомендуется запрограммировать до монтажа контроллера на объекте, но можно и после.

Порядок программирования и настройки ПЛК:

1. Установка CODESYS (см. раздел 4).
2. Выбор контроллера и установка требуемого файла настроек целевой платформы – target-файла (см. раздел 5).
3. Создание и отладка пользовательского проекта в CODESYS.
4. Установка связи ПК с контроллером. Во время установки связи CODESYS автоматически компилирует пользовательский проект и предлагает загрузку скомпилированного кода в ОЗУ контроллера (см. раздел 6.6).
5. Запуск выполнения пользовательского проекта, проверка его работоспособности и, в случае необходимости, отладка (см. раздел 6.11).
6. Если пользовательский проект работает корректно, то сохранение в энергонезависимой Flash-памяти контроллера для последующей загрузки и выполнения при включении питания ПЛК. В противном случае возврат к шагу 5 (в процессе отладки операции могут выполняться многократно).

4 Установка CODESYS

ПРИМЕЧАНИЕ
CODESYS распространяется бесплатно и не требует лицензирования (за исключением отдельных необязательных приложений).

Для установки CODESYS на ПК следует:

1. Скачать установочный exe-файл CODESYS со страницы CODESYS V2 на сайте owen.ru.
2. Запустить установочный exe-файл CODESYS.
3. Следовать инструкциям мастера установки.

ПРИМЕЧАНИЕ
Во время установки CODESYS следует обратить внимание, что язык работы ПО в процессе установки выбирается дважды: при первом выборе русский язык отсутствует в списке доступных языков, при втором – присутствует.

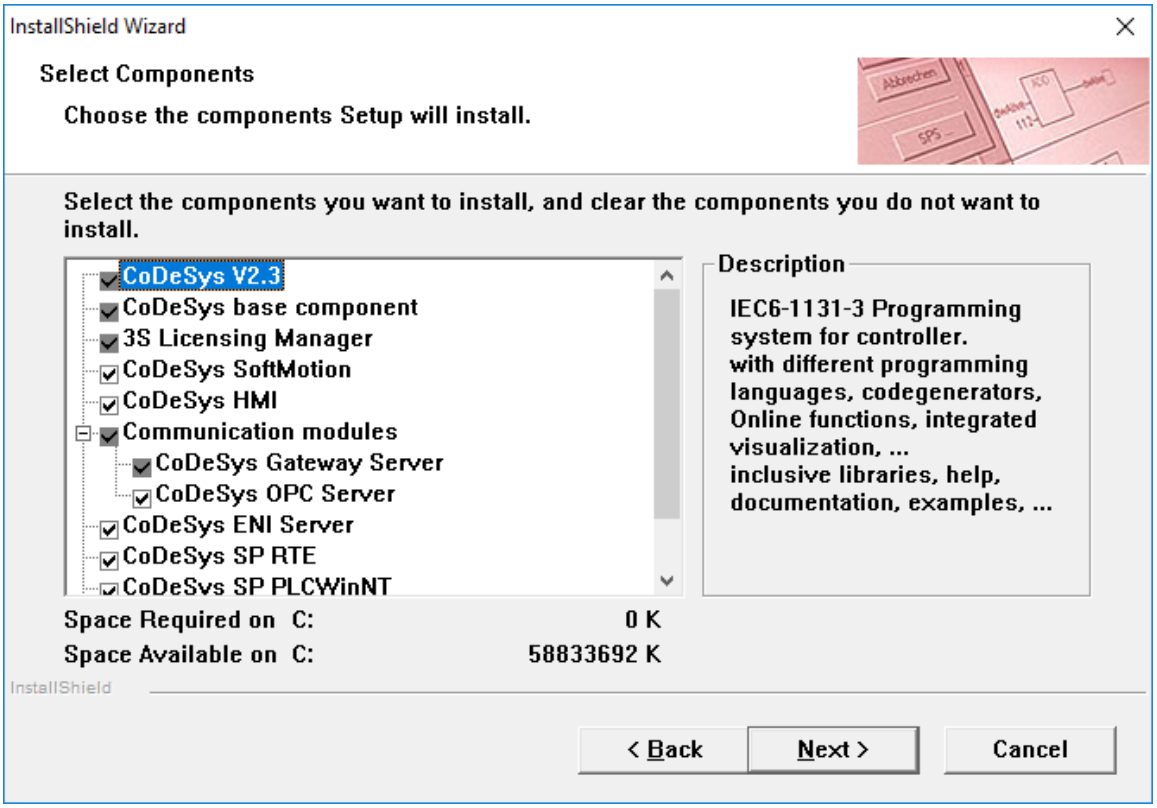


Рисунок 4.1 – Окно установки CODESYS

5 Установка настроек целевой платформы (target-файла)

Исходная информация о конфигурации ПЛК содержится в предварительных настройках целевой платформы (target-файле) контроллера. Настройки целевой платформы поставляются в виде набора файлов, основным (указываемым пользователем в процессе установки настроек) среди которых является target-файл, имеющий расширение **.tnf*, (**T**arget **Inf**ormation).

Target-файл для контроллеров можно скачать на странице CODESYS V2 на сайте owen.ru.

Target-файл содержит информацию о ресурсах конкретного ПЛК (о количестве и типах входов и выходов, интерфейсов, памяти, дополнительных устройств и т. д.), с которыми работает CODESYS. Чтобы контроллер стал доступен для разработки пользовательских программ в конкретной системе с установленным CODESYS, требуется в этой системе установить target-файл.

ПРИМЕЧАНИЕ
Компания «ОВЕН» совершенствует производимые контроллеры и ПО и периодически предлагает пользователю обновленные версии встроенного ПО микроконтроллера и target-файлов. Подробнее об обновлении встроенного ПО микроконтроллера и target-файлов см. раздел 15.

Target-файлы для разных версий встроенного ПО одной и той же модели контроллера могут быть установлены в один и тот же экземпляр CODESYS. Названия target-файлов всегда различаются, например, указанными в них номерами версий. Пользовательская программа, созданная с использованием target-файла для встроенного ПО одной версии может оказаться несовместимым со встроенным ПО другой версии. Для переноса проекта между контроллерами с различными версиями встроенного ПО см. приложение Перенос проекта между несовместимыми версиями встроенного ПО контроллера .

ПРИМЕЧАНИЕ
Следует обратить внимание, что имя target-файла не полностью совпадает с наименованием контроллера: в наименовании контроллера использована кириллица (например, ПЛК110), в названии target-файла – латиница (например, PLC110).

5.1 Способ 1

Настройки целевой платформы (target-файл) устанавливаются с помощью утилиты «InstallTarget». Утилита представляет собой компонент CODESYS и устанавливается на ПК совместно с CODESYS.

Для установки target-файла следует:

1. На ПК выбрать команду **Пуск** → **Программы** → **3S Software** → **InstallTarget**. В поле «Installed Targets» (Установленные файлы) отображается перечень ранее установленных target-файлов.
2. В открывшемся окне утилиты (см. рисунок 5.1) нажать кнопку «Open…» (Открыть).

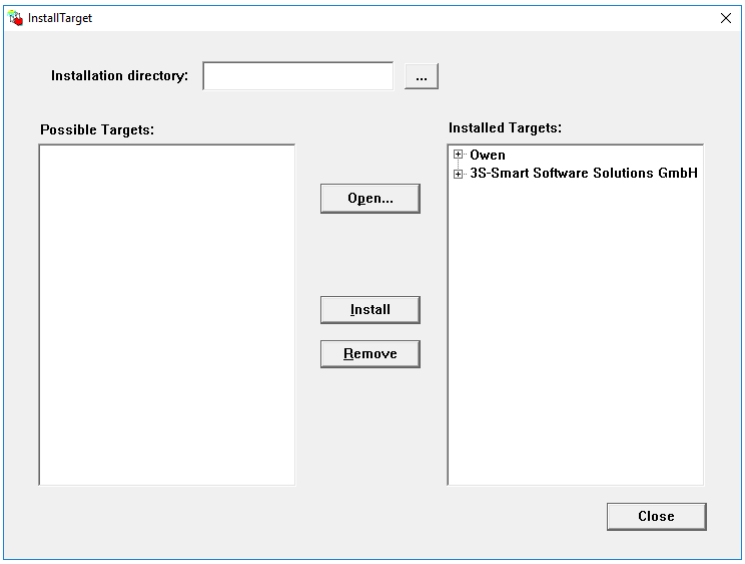


Рисунок 5.1 – Окно утилиты «InstallTarget»

3. В открывшемся окне выбора файла указать путь к target-файлу для требуемого контроллера. В поле «Installation directory» (Путь к файлу) отобразится выбранный путь к папке, в поле «Possible Targets» (Доступные файлы) отобразится список доступных target-файлов.
4. Выделить требуемый target-файл и нажать кнопку «Install» (Установить). Target-файл будет инсталлирован на ПК и отобразится в поле «Installed Targets».

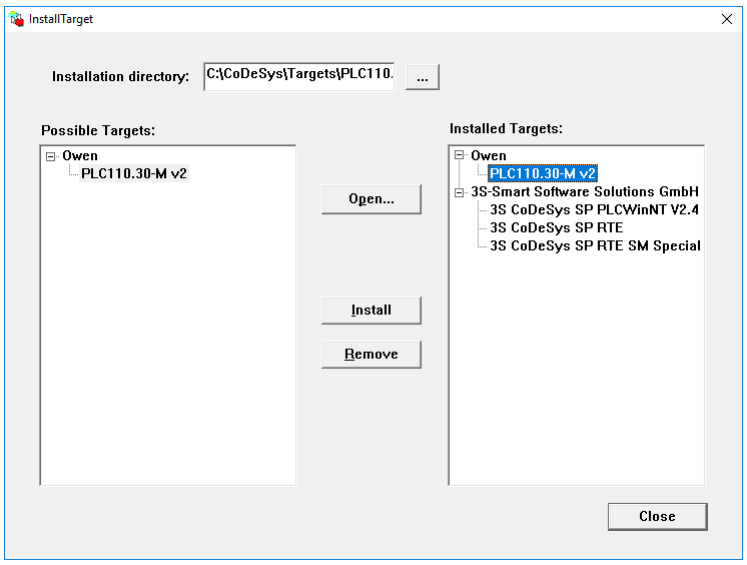


Рисунок 5.2 – Установленный target-файл

ПРИМЕЧАНИЕ
В случае необходимости (например, при ошибке в выборе файла) установленный target-файл можно удалить. Для деинсталляции следует выделить target-файл в поле «Installed Targets» и нажать кнопку «Remove» (Удалить). Файл будет удален и перестанет отображаться в списке в поле «Installed Targets».

5. После завершения инсталляции требуемого target-файла нажать кнопку «Close» (Закрыть). Окно утилиты «InstallTarget» закроется.

5.2 Способ 2

Target-файл устанавливается с помощью специализированной утилиты «InstallTarget. bat», которая входит в архив файлов настроек целевой платформы.

Для установки target-файла следует:

1. Открыть директорию архива файлов настроек целевой платформы.
2. Запустить файл «InstallTarget.bat».
3. Дождатся завершения установки.
4. Перезагрузить CODESYS.

6 Работа с CODESYS

Для запуска CODESYS следует вызвать команду **Пуск** → **Программы** → **3S Software** → **CoDeSys V2.3**. Откроется главное окно CODESYS.

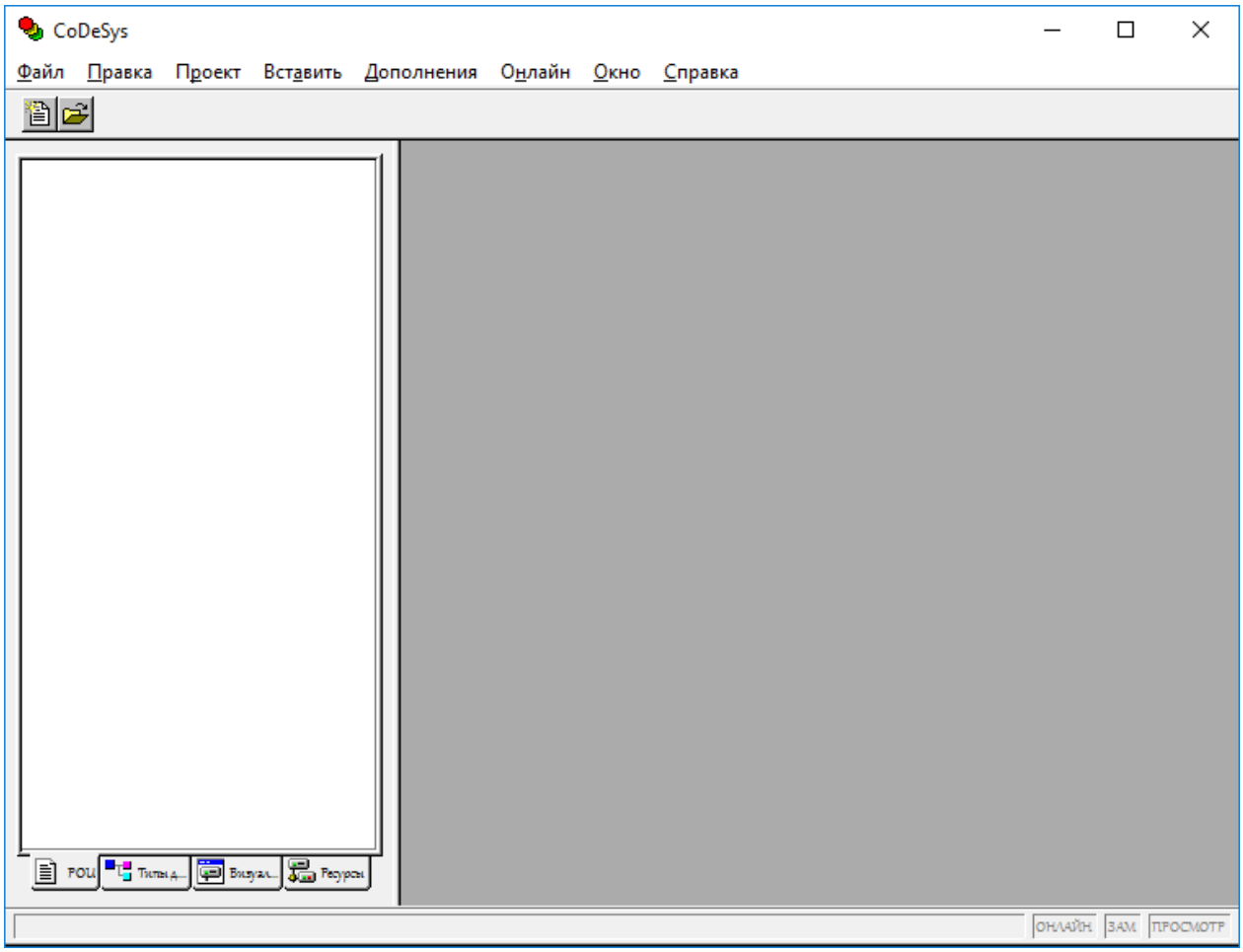


Рисунок 6.1 – Главное окно CODESYS

6.1 Создание проекта

Для создания нового проекта (пользовательской программы) следует:

1. В главном окне ПО CODESYS вызвать команду **Файл** → **Создать** в главного меню или нажать кнопку  на панели инструментов.
2. В открывшемся окне «Настройки целевой платформы» (Target Settings) нажатием на кнопку у правого края поля «Конфигурация» (Configuration) раскрыть список предварительно установленных на ПК target-файлов (см. раздел 5). В списке выделить требуемый файл и щелкнуть на его названии ЛКМ.

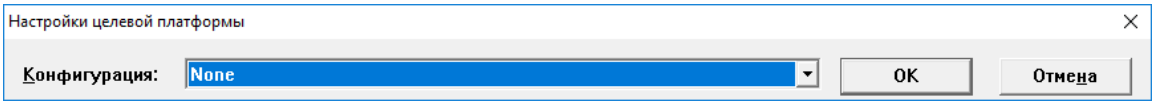


Рисунок 6.2 – Окно «Настройки целевой платформы» (Target Setting)

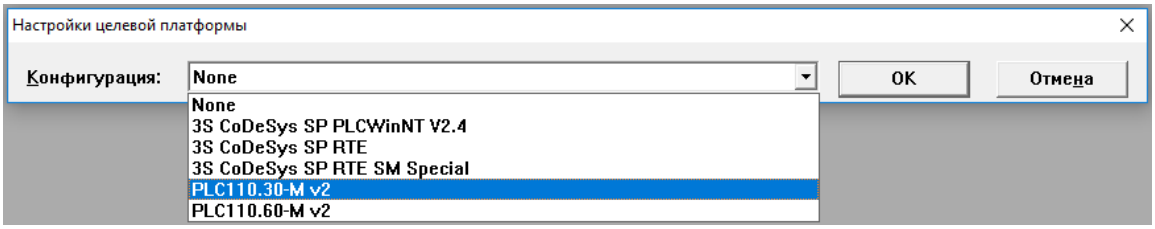


Рисунок 6.3 – Выбор конфигурации

3. В открывшихся вкладках окна «Настройки целевой платформы» (Target Setting) отображаются установленные производителем значения параметров целевой платформы. Как правило, установленные производителем значения параметров не требуют изменения. Исключение могут составить размеры сохраняемой при отключении питания retain-памяти и памяти входов-выходов. Объем retain-памяти по умолчанию установлен «16#4000», что соответствует 16 кбайт.

ПРИМЕЧАНИЕ
В предыдущей версии ПО контроллера для увеличения размера retain-памяти требовалось перейти на вкладку «Распределение памяти» (Memory Layout) окна «Настройки целевой платформы» (Target Setting) и исправить значение в строке Retain («Энергонез.»): значение «16#1000» заменить на «16#4000», в настоящей же версии это значение по умолчанию установлено «16#4000».

4. Для увеличения размера памяти входов-выходов (%I и %Q) следует перейти на вкладку «Распределение памяти» (Memory Layout) окна «Настройки целевой платформы» (Target Setting) и исправить значение в строках «Входы»/«Выходы»: значение «16#1FFF» заменить на «16#2FFF».

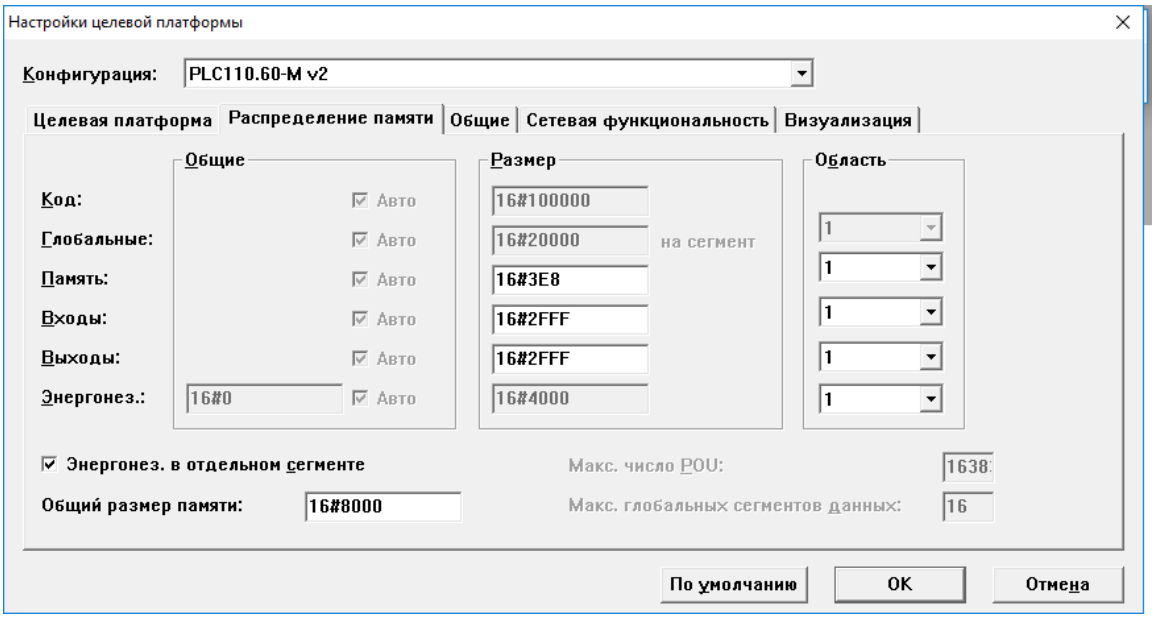


Рисунок 6.4 – Увеличение размера памяти входов-выходов на вкладке «Распределение памяти» (Memory Layout)

5. Нажать кнопку «ОК» окна «Настройки целевой платформы» (Target Setting).

6. В открывшемся окне «Новый программный компонент» (New POU) 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.

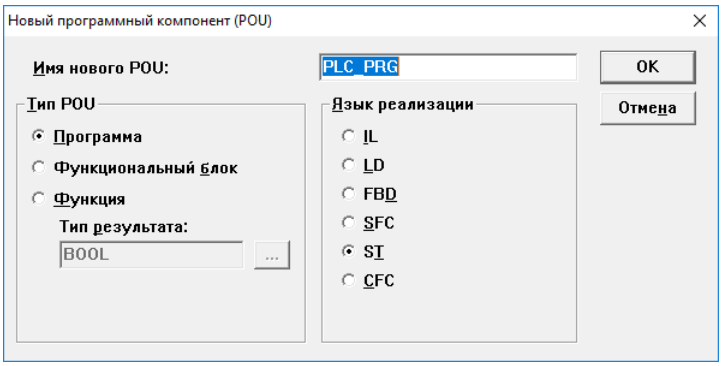


Рисунок 6.5 – Окно «Новый программный компонент» (New POU)

7. В группе переключателей «Язык реализации» (Language of the POU) следует выбрать требуемый язык программирования (о языках программирования см. раздел 6.3). В правой верхней области главного окна программы откроется окно редактора, в котором создается программа, исполняемая контроллером. В зависимости от выбранного языка программирования это окно выглядит по-разному (на рисунке ниже пример для языка программирования LD). В верхней части окна отображается область объявления переменных – «Редактор объявлений», в нижней – область редактора собственно программы.

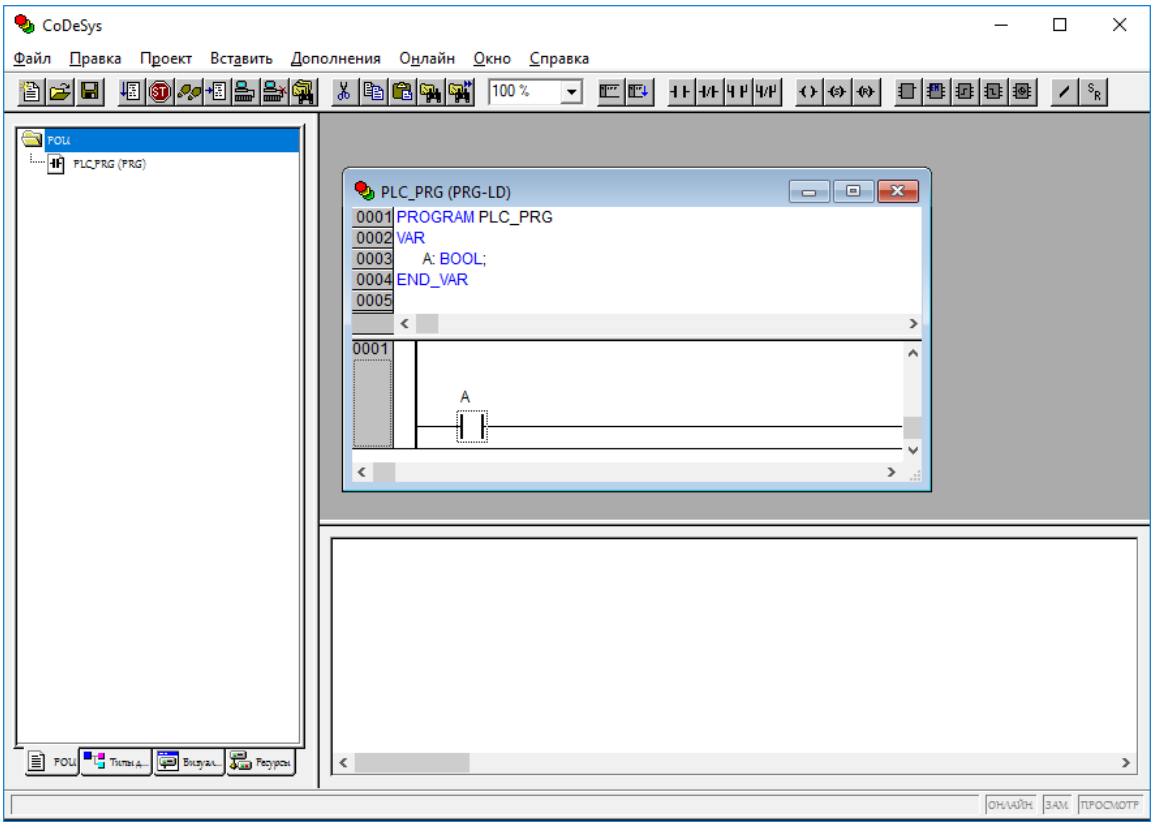


Рисунок 6.6 – Главное окно проекта

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

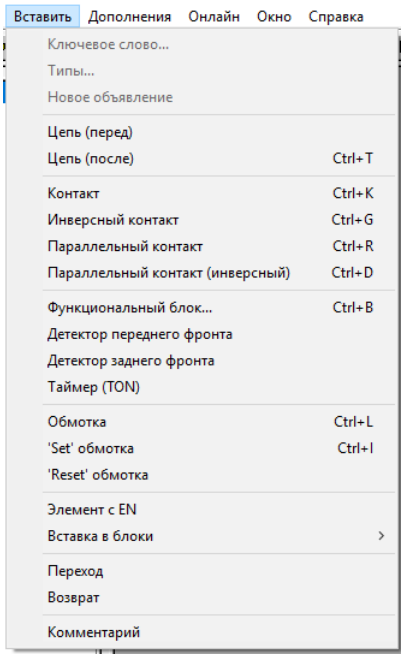


Рисунок 6.7 – Контекстное меню области редактирования программы на языке программирования LD

6.2 Основные элементы интерфейса

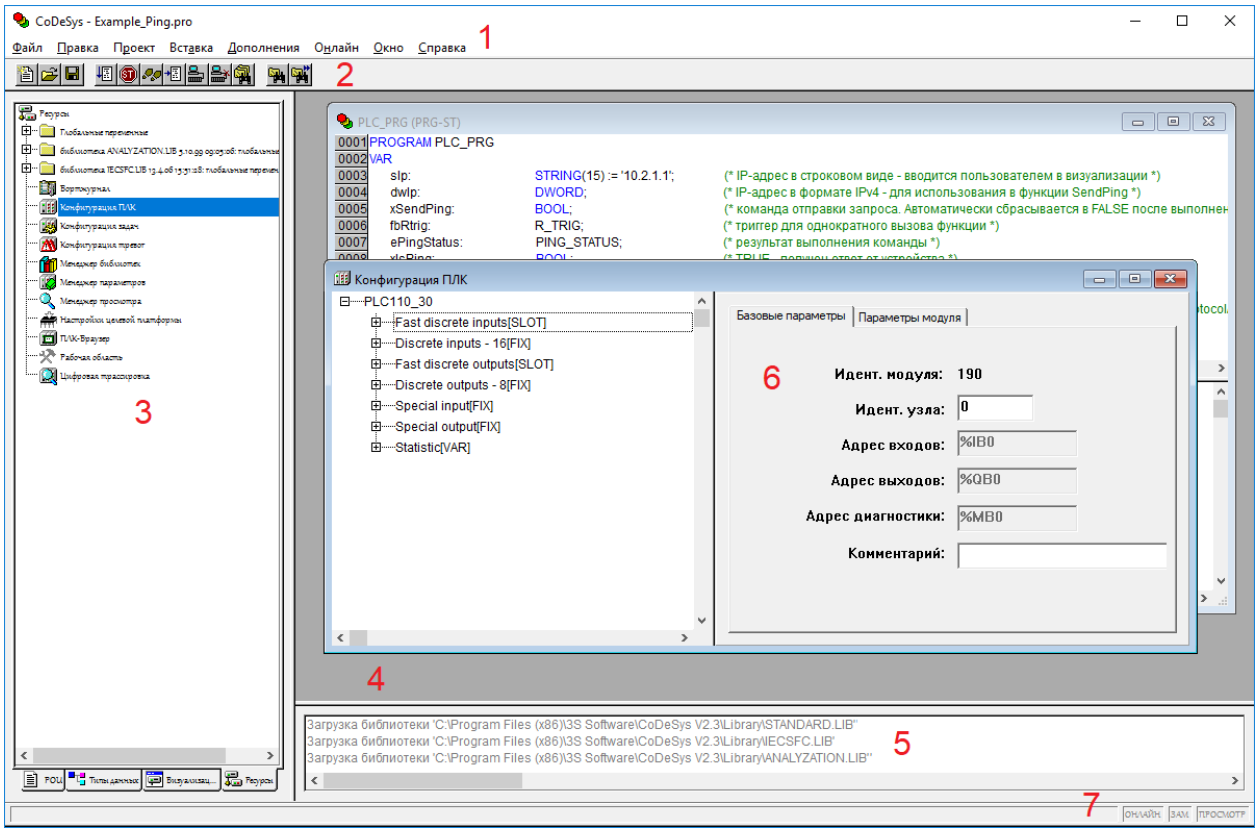


Рисунок 6.8 – Основные элементы интерфейса

Главное окно CODESYS содержит следующие элементы:

1. **Главное меню** – перечень доступных групп команд ПО. В различных режимах работы группы команд главного меню дополняются специализированными командами.

2. **Панель инструментов** – кнопки, дублирующие часто используемые команды программы. В различных режимах работы панель инструментов дополняется специализированными панелями.

   ПРИМЕЧАНИЕ
   Значение команд главного меню и панели инструментов, а также горячие клавиши для вызова см. в разделе 6.2.3.

3. **Организатор объектов** – переключатель групп режимов работы ПО (см. раздел 6.2.1).
4. **Рабочая область** – служит для отображения окон режимов работы, кода и элементов проекта.
5. **Окно (область) сообщений** – служит для отображения сообщений компилятора, результатов поиска и списка перекрестных ссылок.
6. **Окно режима работы** – служит для изменения параметров проекта.
7. **Строка статуса** – содержит информацию о текущем состоянии проекта (см. раздел 6.2.2).

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

6.2.1 Организатор объектов

Организатор объектов расположен в левой части главного окна программы (см. рисунок 6.8) и предназначен для вызова режимов работы ПО. Организатор объектов включает вкладки:

- «POU»;
- «Типы данных»;
- «Визуализации»;
- «Ресурсы».

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

Для перехода на требуемую вкладку следует щелкнуть ЛКМ на наименовании требуемой вкладки (в нижней части организатора объектов). Для перехода к требуемому объекту в пределах выбранной вкладки следует щелкнуть ЛКМ 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 ЛКМ на наименовании требуемого объекта или, выбрав наименование требуемого объекта, нажать клавишу **Enter**. Окно режима (или окно одного из объектов режима) откроется в рабочей области главного окна.

«POU»

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 поддерживаемых CODESYS языков программирования.

«Типы данных»

На вкладке «Типы данных» отображается иерархический список типов данных, используемых в проекте. Кроме стандартных типов данных можно использовать определяемые пользователем типы данных: структуры, перечисления и ссылки.

«Визуализации»

На вкладке «Визуализации» отображается иерархический список элементов визуализации пользовательской программы – **графических представлений** объекта управления. Визуализация непосредственно связана с созданной в CODESYS пользовательской программой контроллера. **Редактор визуализации** CODESYS предоставляет набор готовых графических элементов, которые можно соответствующим образом связать с переменными проекта.

В режиме «Online» представление элементов на экране изменяется в зависимости от значений переменных (см. раздел 7.5.7). Визуализация может исполняться в системе программирования, в отдельном приложении «CODESYS HMI».

«Ресурсы»

На вкладке «Ресурсы» отображается иерархический список ресурсов – объектов CODESYS, обеспечивающих конфигурацию проекта:

- глобальные переменные, используемые во всем проекте;
- менеджер библиотек для подключения необходимых библиотек к проекту;
- журнал записи действий во время исполнения;
- конфигурация тревог для конфигурирования обработки тревог в проекте;
- конфигурация ПЛК для конфигурирования аппаратуры контроллера;
- конфигурация задач для управления задачами;
- менеджер для просмотра и заказа наборов значений переменных;
- настройки целевой платформы.

Двойное нажатие ЛКМ на требуемой записи в списке «Ресурсы» приводит к открытию в рабочей области окна выбранного режима («ресурса»).

6.2.2 Строка статуса

В строке статуса отображаются:

- при выборе пункта меню – его описание;
- во время работы 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 название элемента (например, Rectangle);
- во время работы 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 пользовательская программа:
  - **SIM** – в режиме эмуляции;
  - **RUN** – пользовательская программа запущена;
  - **BP** – установлена точка останова;
  - **FORCE** – фиксируются переменные.

6.2.3 Команды главного меню

Таблица 6.1 – Элементы управления главного меню

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
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 команды какой-то проект уже открыт и в него были внесены изменения, то CODESYS предложит сохранить открытый проект
Закрыть (Close)			Закрывает открытый в данный момент проект. Если с момента открытия в проект были внесены изменения, то CODESYS предложит сохранить открытый проект
Сохранить (Save)		<Ctrl + S>	Сохраняет файл проекта
Сохранить как… (Save as…)			Сохраняет проект или библиотеку под новым именем. Исходный файл не изменяется

Продолжение таблицы 6.1

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
Сохранить/отправить архив (Save/Mail Archive…)			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)		<Alt + F4>	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, можно отменить все изменения, выполненные после открытия окна
Вернуть (Redo)		<Ctrl + Y>	Возвращает последнее изменение, отмененное в открытом редакторе или в организаторе объектов командой Undo
Вырезать (Cut)		<Ctrl + X>	Перемещает выделенный элемент в буфер. Выделенный элемент удаляется из окна редактора
Копировать (Copy)		<Ctrl + C>	Копирует выделенный элемент в буфер, содержимое окна редактора не изменяется
Вставить (Paste)		<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
Удалить (Delete)		<Delete>	Удаляет выбранную область, содержимое буфера не изменяется
Найти (Find)		<Ctrl + F>	Находит введенный текст в активном окне редактора. Открывает диалоговое окно поиска
Найти далее (Find next)		<F3>	Начинает поиск введенного текста с текущей позиции и далее
Найти и заменить (Replace)		<Ctrl + H>	Находит заданный текст и заменяет его на введенный. Вызов команды открывает диалоговое окно поиска и замены выбранного текста
Ассистент ввода (Input Assistant)		<F2>	Открывает диалоговое окно выбора элемента, который можно ввести в текущей позиции. В левом столбце следует выбрать категорию элементов, в правом – требуемый элемент, а затем нажать «OK»
Автоматическое объявление переменных (Auto Declare)		<Shift + F2>	Открывает диалог объявления переменных

Продолжение таблицы 6.1

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
Следующая ошибка (Next Error)		<F4>	Показывает следующую ошибку, если проект скомпилирован с ошибками. Открывается соответствующий редактор в том месте, где допущена ошибка, а в окне сообщений отображается краткое описание ошибки
Предыдущая ошибка (Previous Error)		<Shift + F4>	Показывает предыдущую ошибку, если проект скомпилирован с ошибками. Открывается соответствующий редактор в том месте, где допущена ошибка, а в окне сообщений отображается краткое описание ошибки
Макросы (Macros)			Показывает список всех определенных в проекте макрокоманд (макросов). Во время выполнения макроса открывается окно «Process Macro», в котором выводится имя макроса и имя активной команды
Проект (Project)			
Компилировать (Build)		<F11>	Компилирует только POU, которые были изменены
Компилировать все (Rebuild all)			Компилирует весь проект, даже если он не был изменен
Очистить все (Clear all)			Стирает всю информацию о предыдущей компиляции и загрузке проекта в контроллер
Загрузить информацию о загрузке кода (Load Download-Information)			Загружает информацию о загрузке кода в контроллер, если она была сохранена в директории, отличной от той, в которой находится проект
Перевод на другой язык (Translate into another language)			Переводит текст проекта на другой язык (используется вспомогательный текстовый файл, созданный в CODESYS и переведенный в текстовом редакторе на требуемый язык)
Документ… (Document)			Создает версию проекта для печати
Экспорт… (Export…)			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
Импорт… (Import…)			Импортирует в проект данные из выбранного файла
Siemens импорт (Siemens Imports)			Импортирует переменные и POU из файлов Siemens-STEP5 и Siemens-STEP7
Объединить… (Merge)			Объединяет два проекта
Сравнить… (Compare)			Сравнивает два проекта или разные версии одного и того же проекта
Информация о проекте (Project info)			Сохраняет дополнительную информацию о проекте
Глобальный поиск (Global Search)		<Ctrl + F>	Находит заданный текст в POU, типах данных или разделе глобальных переменных проекта

Продолжение таблицы 6.1

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
Глобальная замена (Global Replace)			Находит заданный текст в POU, типах данных или в глобальных переменных проекта и заменяет его другим
Просмотр экземпляра (View Instance)			Показывает экземпляры выбранного в организаторе объектов функционального блока. Отображает список всех экземпляров выбранного функционального блока и его реализаций
Показать дерево вызовов (Show Call Tree)			Показывает дерево вызовов выбранного объекта в новом окне (проект должен быть скомпилирован)
Показать перекрестные ссылки (Show Cross Reference)			Открывает диалоговое окно, в котором отображены адрес, место расположения (POU, номер строки) переменной (проект должен быть скомпилирован)
Контроль (Check)			Команды этого меню используются для дополнительного семантического контроля (проект должен быть скомпилирован без ошибок)
Добавить действие (Add Action)			Создает действие, связанное с блоком, выделенным в организаторе объектов. Требуется задать имя действия и язык, на котором оно будет описано
Пароли Группы пользователей… (User Group Passwords)			Устанавливает права доступа к объекту для различных групп пользователей

Продолжение таблицы 6.1

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
Вставить (Insert)			
Ключевое слово… (Declaration keywords)			Вывод списка 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
Тип… (Types)			Вывод списка доступных типов для их быстрого ввода
Новое объявление (New declaration)			Добавляет новую переменную в таблицу редактора объявлений
Дополнения (Extras)			
Команды данного пункта меню могут меняться в зависимости от режима работы			
Онлайн (Online)			
Подключение (Login)			Устанавливает соединение CODESYS с контроллером (или запускает программу эмуляции) и включает режим «Online»
Отключение (Logout)			Разрывает соединение с контроллером или заканчивает работу программы (в режиме эмуляции). Включает режим «Offline»
Загрузка (Download)			Загружает код проекта в контроллер
Старт (Run)		<F5>	Запускает программу на выполнение в контроллере или в режиме эмуляции
Стоп (Stop)		<Shift + F8>	Останавливает программу при ее выполнении в контроллере или в режиме эмуляции
Сброс (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
Сброс (холодный) Reset (cold)			Холодный сброс. Выполняет те же действия, что и при команде «Сброс», и дополнительно выполняет инициализацию энергонезависимой области памяти RETAIN
Сброс (заводской) Reset (original)			Выполняет «Сброс (холодный)», а также инициализация области PERSISTENT и удаление программы пользователя. Восстанавливаются заводские настройки контроллера
Переключить точку останова (Toggle Breakpoint)		<F9>	Устанавливает точку останова в текущей позиции активного окна. Если в этой позиции уже стоит точка останова, то она будет удалена
Диалог точек останова (Breakpoint Dialog)			Открывает диалоговое окно управления точками останова в проекте, в нем указаны все заданные точки останова

Продолжение таблицы 6.1

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
Шаг поверху (Step over)		<F10>	Выполняет одну инструкцию программы. Если это инструкция вызова POU, то данный POU выполняется целиком, затем программа останавливается
Шаг детальный (Step in)		<F8>	Выполняет программу по шагам, с заходом в вызываемые блоки. Вызываемые POU открываются в отдельных окнах
Один цикл (Single Cycle)		<Ctrl + F5>	Выполняет один рабочий цикл контроллера. Команду можно повторять многократно для отслеживания работы программы по рабочим циклам
Записать значения (Write values)		<Ctrl + F7>	Перед началом рабочего цикла присваивает переменной (или нескольким переменным) заранее введенные значения
Фиксировать значения (Force values)		<F7>	Фиксирует значения одной или нескольких переменных. Заданное значение записывается в начале и в конце каждого управляющего цикла: 1. Чтение входов. 2. Фиксация переменных. 3. Выполнение кода программы. 4. Фиксация переменных. 5. Запись выходов.
Освободить фиксацию (Release force)		<Shift + F7>	Отменяет фиксацию значений переменных
Диалог Запись/ Фиксация (Write/Force-Dialog)		<Ctrl + Shift + F7>	Открывает окно, содержащее таблицы записываемых (Writelist) и фиксируемых (Forcelist) переменных. В левом столбце таблиц отображаются имена переменных, в правом – их установленные значения
Показать стек вызовов… (Show Call Stack)			Показывает список вызванных POU, когда программа остановлена в точке останова
Отображать поток выполнения (Display Flow control)			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
Режим эмуляции (Simulation)			«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
Параметры связи (Communication Parameters)			Выводит диалоговое окно настройки параметров связи ПК и ПЛК (если используется OPC или DDE сервер, то эти параметры можно настроить из их конфигурации)

Продолжение таблицы 6.1

Команда меню	Кнопка панели инструмен-тов	Горячие клавиши	Описание команды
Загрузка исходных текстов (Sourcecode download)			Загружает исходные тексты проекта в контроллер (именно исходные тексты проекта – не код проекта, который создается при компиляции)
Создание загрузочного проекта (Create bootproject)			Делает код проекта автоматически загружаемым при перезапуске контроллера: проект будет выполняться автоматически при перезапуске ПЛК
Записать файл в ПЛК (Write file to PLC)			Записывает в ПЛК выбранный файл (любого типа), размер файла ограничивается размером карты памяти контроллера
Читать файл из ПЛК (Read file from PLC)			Считывает ранее сохраненный в контроллере файл и сохраняет его в указанную директорию на ПК
Окно (Window)			
По вертикали (Tile Vertical)			Упорядочивает размещение окон по вертикали так, чтобы они не перекрывали друг друга и полностью занимали рабочую область
По горизонтали (Tile Horizontal)			Упорядочивает размещение окон по горизонтали так, чтобы они не перекрывали друг друга и полностью занимали рабочую область
Каскадом (Cascade)			Упорядочивает окна каскадом – каждое следующее поверх остальных
Выровнять свернутые (Arrange Symbols)			Выстраивает свернутые окна в ряд в нижней части Рабочего окна
Закрыть все (Close All)			Закрывает все окна
Сообщения (Messages)		<Shift + Escape>	Открывает окно сообщений, которое содержит информацию о предыдущей компиляции, проверке или сравнении проекта
Менеджер библиотек (Library Manager)			Открывает окно менеджера библиотек
Бортжурнал (Log)			Открывает «бортжурнал» – детальный протокол последовательности действий, которые были выполнены во время «Online»-сессии. Бортжурнал записывается в двоичный файл формата **.log*
Справка (Help)			
Содержание… (Contents)			Открывает окно системы оперативной помощи
Поиск… (Search)			Переход к контекстному поиску по текстам оперативной помощи
О программе… (About)			Открывает окно с информацией о программе CODESYS

6.2.4 Основные режимы (редакторы)

CODESYS предоставляет встроенные специализированные редакторы для всех пяти языков стандарта 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;
  - макроопция для структурирования больших диаграмм.

Специальные редакторы отвечают за прикладные функции рабочей среды:

- 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.

6.3 Языки программирования

В соответствии с требованиями стандарта МЭК 61131-3 CODESYS поддерживает следующие языки программирования:

- IL;
- ST;
- FBD;
- LD;
- SFC.

CODESYS также поддерживает «Язык непрерывных функциональных схем» (**CFC**), который отличается от FBD тем, что блоки и соединители в CFC располагаются свободно, разрешены циклы и свободные соединения.

Подробное описание языков программирования приведено в документе «Руководство пользователя по программированию ПЛК в CoDeSys V2.3» на сайте owen.ru.

IL (список инструкций)

IL – текстовый язык, схожий с ассемблером STEP5 компании SIEMENS. Все операции языка производятся через аккумулятор, язык IL 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.

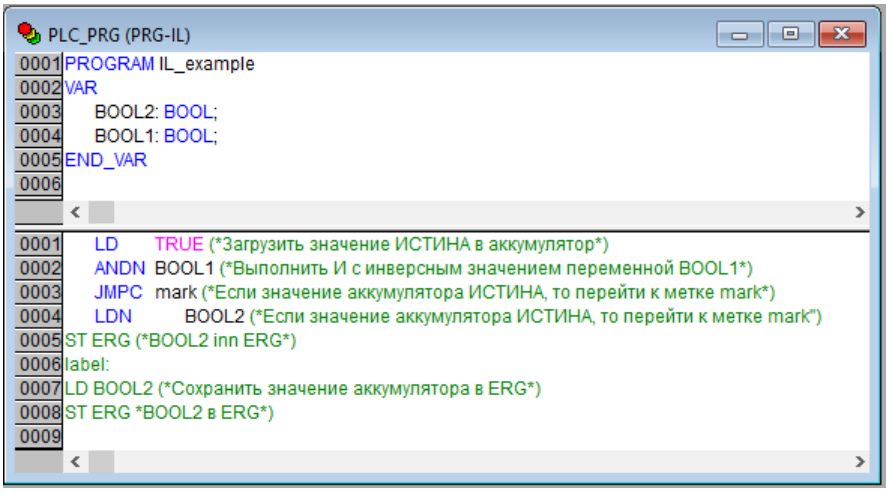


Рисунок 6.9 – Пример программы на языке IL

ST (структурированный текст)

ST – текстовый язык высокого уровня, схожий с языком Pascal. ST оптимален для программирования циклов и условий и состоит из набора инструкций, которые могут использоваться в условных операторах (IF…THEN…ELSE) и в циклах (WHILE…DO).

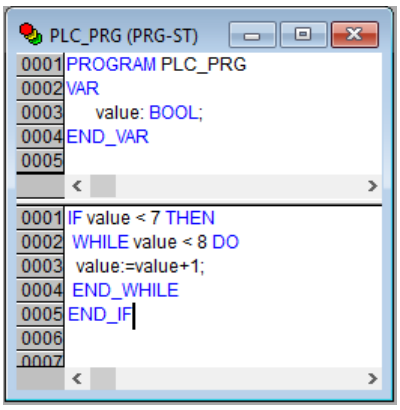


Рисунок 6.10 – Пример программы на языке ST

FBD (функциональные блоковые диаграммы)

FBD – графический язык программирования, который работает со схемами, состоящими из блоков и операндов – с последовательностью цепей. Каждая цепь содержит логическое или арифметическое выражение, вызов функционального блока, переход или инструкцию возврата.

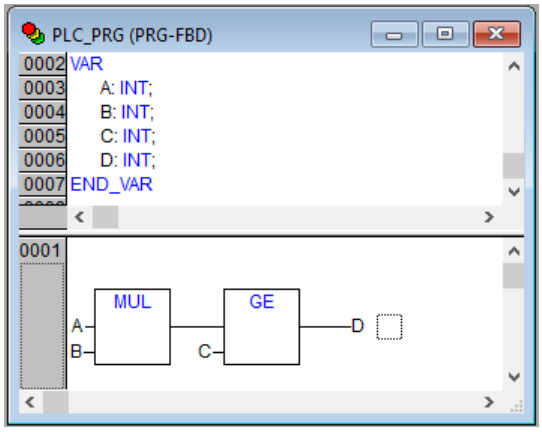


Рисунок 6.11 – Пример программы на языке FBD

LD (релейные диаграммы)

LD – 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. Язык сложен в использовании для работы с аналоговыми типами данных.

LD лучше всего подходит для построения логических переключателей и используется для программирования большинства ПЛК. Допускается 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 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 «OFF» (Выключено).

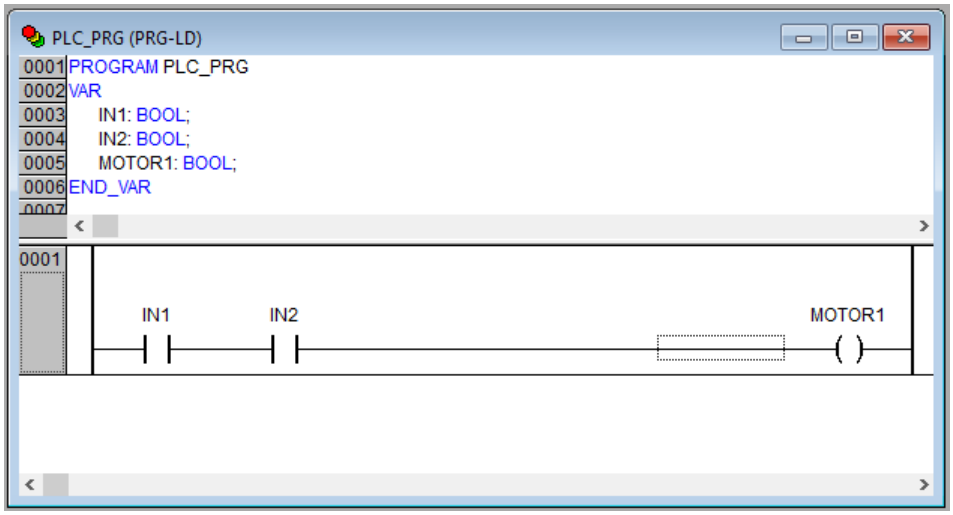


Рисунок 6.12 – Пример программы на языке LD

SFC (последовательные функциональные схемы)

SFC – графический язык, который используется для структурирования приложений и состоит из шагов и переходов. Действия выполняются внутри шагов. SFC не конвертируется в другие языки.

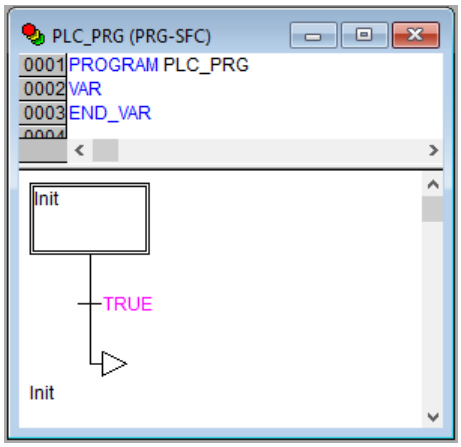


Рисунок 6.13 – Пример программы на языке SFC

CFC (непрерывные функциональные схемы)

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

ПРИМЕЧАНИЕ
Свобода размещения компонентов и соединений приводит к тому, что требуется задавать порядок выполнения программы. Группа команд «Порядок» контекстного меню позволяет отобразить порядковые номера (по очередности выполнения) элементов программы и изменить этот порядок в случае необходимости. Порядковые номера элементов отображаются в затемненном поле у правого верхнего угла каждого элемента.

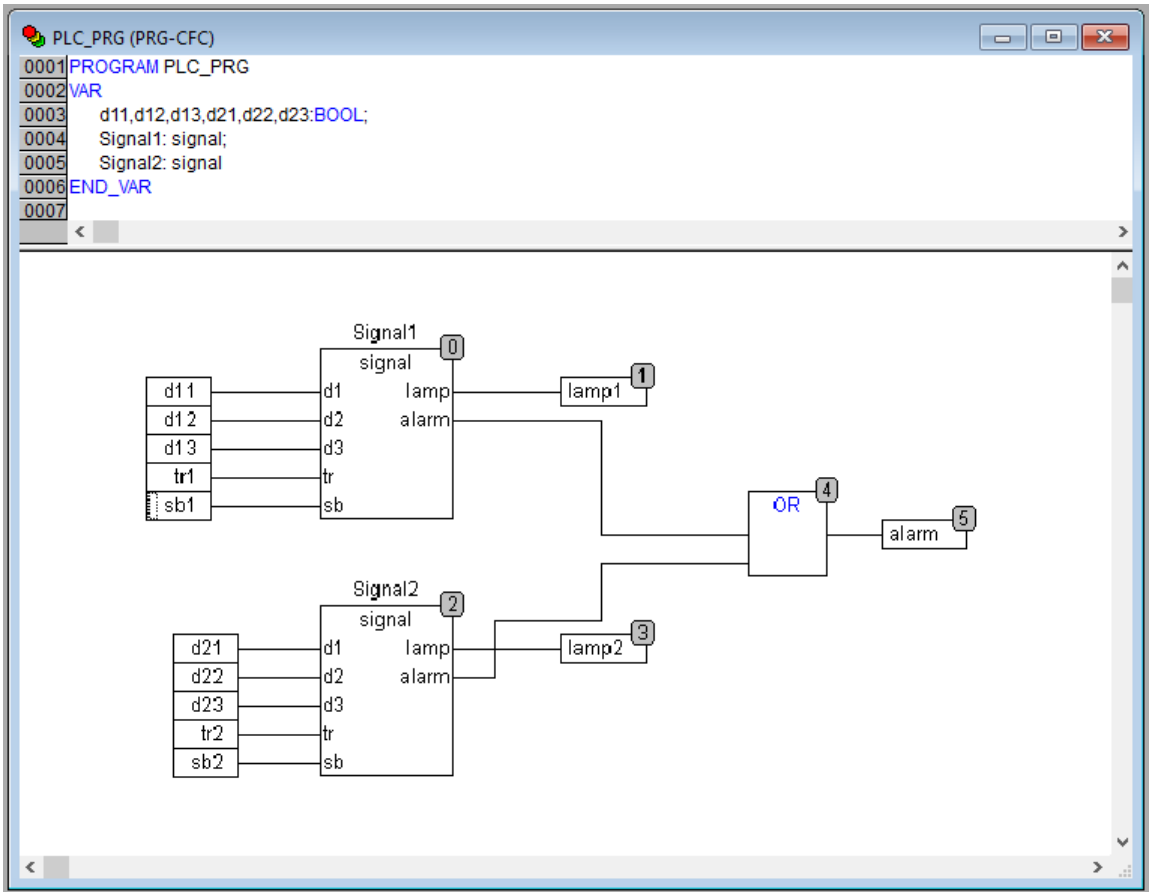


Рисунок 6.14 – Пример программы на языке CFC

6.4 Редактор

Все режимы редактирования (далее – «редакторы») программных компонентов POU содержат раздел объявлений (верхняя часть окна) и область кода (нижняя часть окна), см. рисунок 6.15.

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 откроется окно редактора. Тип окна зависит от выбранного языка программирования (см. раздел 6.3). В каждом окне редактора становятся доступны команды контекстного меню, содержащие основные операции, доступные в выбранном языке. Панель инструментов главного окна дополняется панелью, кнопки которой аналогично командам контекстного меню вызывают основные операции, доступные в выбранном языке. Окно редактора открывается также при добавлении программного компонента.

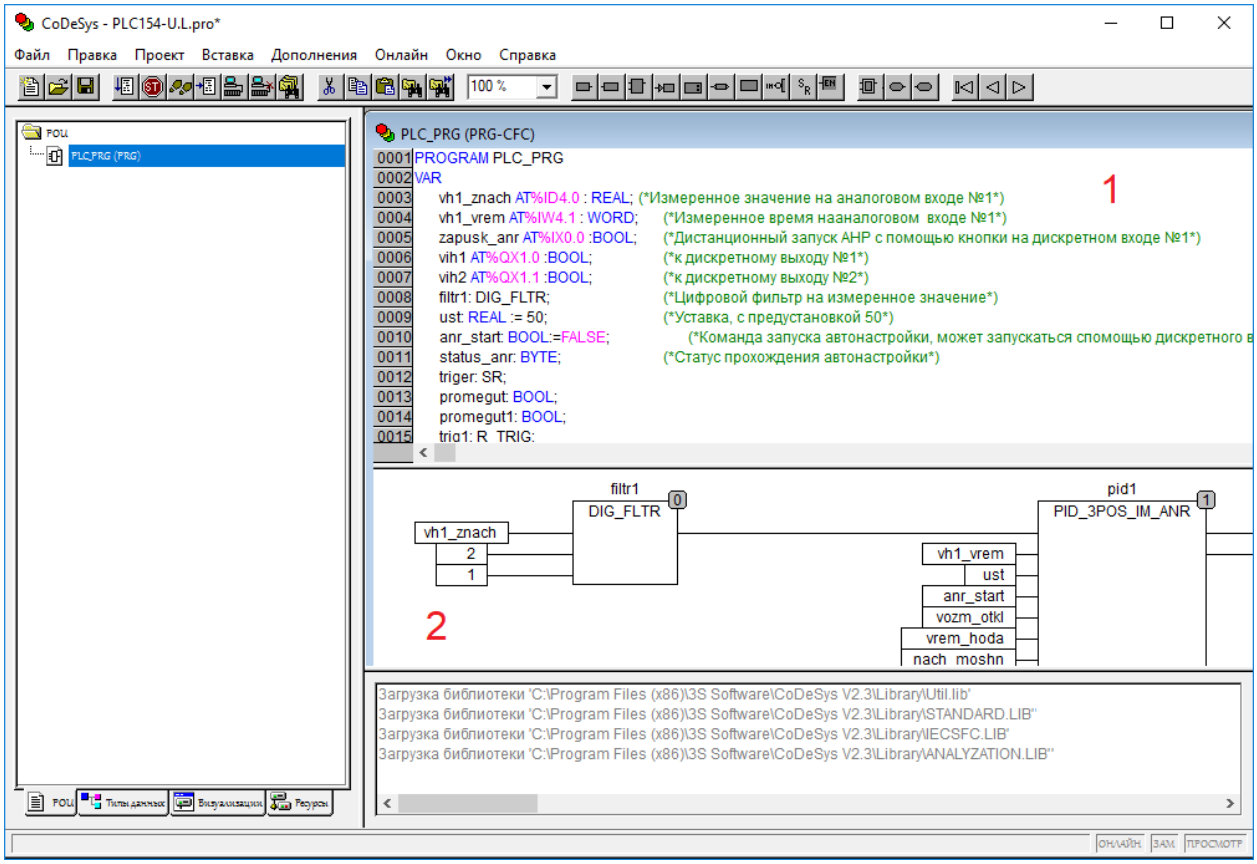


Рисунок 6.15 – Пример окна редактирования: 1 – раздел объявлений; 2 – область кода

6.5 Типы данных

Тип данных определяет род информации и методы ее обработки и хранения, количество выделяемой памяти. Программист может использовать элементарные (базовые) типы данных (подробнее см. раздел 7.3.1) или создавать собственные (пользовательские) типы на их основе (см. раздел 7.3.2).

6.6 Установка связи с ПЛК

Пользовательскую программу можно загрузить в контроллер только после установки связи контроллера с ПК.

Поддерживаемые интерфейсы связи контроллера и ПК для загрузки пользовательской программы:

- Ethernet;
- RS-232;
- USB.

ПРИМЕЧАНИЕ
Интерфейс RS-485 предназначен только для обмена данными, но не для загрузки.

ПРИМЕЧАНИЕ
В некоторых случаях может потребоваться подключение ПК разработчика к ПЛК, подключенного к другому ПК, соединенных локальной сетью, либо сетью Интернет. Вариант с локальной сетью описан в данном руководстве, вариант соединения с использованием Интернета по своим настройкам немногим отличается от соединения по локальной сети.

Настройка связи требуется однократно для определенного интерфейса. Во время отладки настройка связи может потребоваться только в случае перехода на другой интерфейс связи.

6.6.1 Настройка интерфейсов связи

Для настройки соединения с контроллером следует выбрать в главном меню **Онлайн → Параметры связи…**. Откроется окно настройки интерфейсов связи с контроллером.

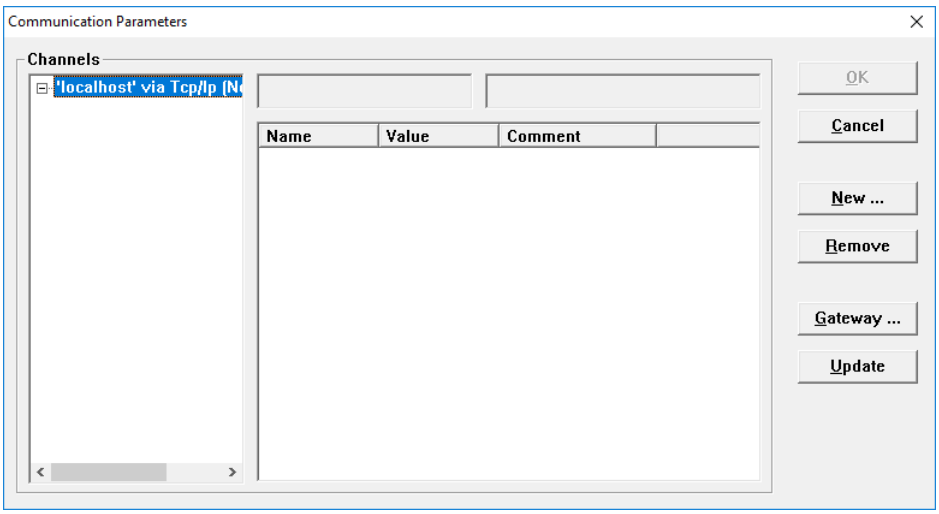


Рисунок 6.16 – Окно настройки интерфейсов связи

В левой части окна в иерархическом списке (Channels) перечислены все созданные каналы связи.

В средней части окна отображается таблица настроек канала связи, выделенного в иерархическом списке. Таблица состоит из колонок:

- **Название параметра** (Name);
- **Значение** (Value);
- **Комментарий** (Comment).

Каждый вид подключения имеет свой набор параметров: адрес или номер устройства, если в данный канал включено несколько устройств, а также свойства канала связи, например, скорость передачи данных, наличие защиты от помех, временные задержки, и т. п. Над таблицей расположены две панели, которые отображают текст без возможности редактирования: в левой панели – название вида (протокола) связи, используемого в текущем канале.

В правой части окна расположены кнопки для работы со списком каналов связи:

- **OK/Cancel** – применить/отменить изменения, внесенные в список каналов связи;
- **New/Remove** – создать новую и удалить текущую запись о канале связи;
- **Gateway** – вызов окна переключения локальных/удаленных подключений (см. рисунок 6.17);
- **Update** – обновление данных.

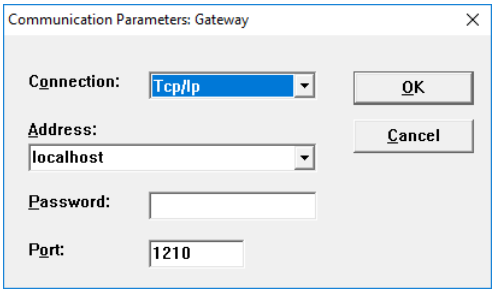


Рисунок 6.17 – Окно команды Gateway

В окне Gateway переключатель **Connection** (Соединение) устанавливает соединение при подключении:

- «Local» – локальное;
- «Tcp/Ip» – сетевое с использованием протокола TCP/IP.

Для локального соединения требуется установить только параметры порта. В случае подключения по сети с использованием протокола TCP/IP, в текущем диалоговом окне становятся доступными дополнительные настройки:

- **сетевой адрес компьютера (Address)** с подключенным к нему контроллером (IP-адрес, имя компьютера, в случае использования в сети протокола DNS либо слово «localhost», если требуется установить соединение с контроллером, подключенным по локальной сети);
- **пароль (Password)** для доступа к защищенному соединению;
- **порт (Port)** указывается для введенного сетевого адреса (по умолчанию используется 1210).

Смена подключения с локального на удаленное с использованием протокола TCP/IP (и наоборот) приводит к очистке списка подключений в таблице настроек канала связи.

6.6.1.1 RS-232

Интерфейс RS-232 (последовательный порт) в промышленной автоматике является распространенным интерфейсом взаимодействия микропроцессорных устройств. В ПК такие интерфейсы обозначались, как COM-порты (порты COM1, COM2, … COMn). В современных конфигурациях ПК порты RS-232 нередко отсутствуют и для подключения различных внешних устройств используются более высокоскоростные USB-порты, обладающие, однако, меньшими возможностями в плане длины соединяющих проводов (до пяти метров).

Для настройки интерфейса RS-232 следует:

1. Выбрать **Онлайн** → **Параметры связи…** в главном меню. Откроется окно настройки интерфейсов связи с контроллером.
2. Нажать кнопку **New**. Откроется окно «Communication parameters: New Channel» – задания нового канала связи.
3. В окне задать имя нового соединения (например, Owen) и выбрать из перечня интерфейс соединения **Serial (RS232)** для связи по интерфейсу Debug RS-232 или USB Device.
4. В случае выбора соединения **Serial (RS232)** в настройках параметров следует задать:
   - COM-порт (параметр Port), по которому ПЛК подключается к ПК;
   - скорость соединения (параметр Baudrate) – 115200 бит/с;
   - бит четности (параметр Parity) – No;
   - параметры Gateway → Connection – «local».

Работа с USB-портом ничем не отличается от работы с обычным последовательным портом, увидеть номер данного порта можно с помощью диспетчера устройств Windows. По умолчанию скорость передачи составляет 115200 бит/с, размер передаваемого слова 8 бит, проверка четности выключена, количество стоповых битов равно 1.

Драйвер виртуального COM-порта можно скачать на сайте owen.ru на странице контроллера. Для установки драйвера следует подключить включенный ПЛК к USB-порту ПК стандартным кабелем типа A-B (в комплект поставки не входит). После отключения питания или перезагрузки ПЛК для установки связи может потребоваться повторное отключение и подключение кабеля USB-порта для повторной инициализации драйвера.

6.6.1.2 Ethernet

Для установки соединения между ПЛК и ПК по интерфейсу Ethernet требуется выполнение следующих условий:

- IP-адреса должны выбираться из значений, допустимых для частного пользования*;
- IP-адрес ПК должен быть статическим;
- IP-адреса и маска подсети ПК и контроллера устанавливаются так, чтобы они находились в одной IP-подсети.

ПРИМЕЧАНИЕ
* Согласно правилам распределения IP-адресов, некоторые адреса могут использоваться свободно всеми желающими при организации подсетей с протоколом TCP/IP: адреса 10.x.y.z с маской 255.0.0.0, адреса 172.16.x.y…172.31.x.y с маской 255.240.0.0 и адреса 192.168.x. y с маской 255.255.0.0, где x, y, z – целые числа в диапазоне от 0 до 255.

IP-адрес контроллера можно изменить с помощью команды «SetIP», подаваемой в режиме «ПЛК-Браузер (PLC-Browser)» (подробнее о работе в режиме «ПЛК Браузер (PLC-Browser)» см. раздел 12). Для изменения IP-адреса связь с контроллером должна быть предварительно установлена через интерфейс Debug RS-232 или USB Device.

Дополнительный IP-адрес ПК задается в свойствах протокола TCP/IP в настройках сетевого окружения Windows. При изготовлении IP-адрес контроллера – **10.0.6.10**, поэтому ПК следует присвоить дополнительный IP-адрес в подсети 10.0.6, отличный от адреса 10.0.6.10, маска подсети задается равной 255.255.0.0. Подробно процесс присвоения дополнительного IP-адреса для ПК приведен в видеоинструкции Подключение ПЛК к ПК по Ethernet.

6.6.1.3 TCP/IP

Для установки связи с помощью протокола TCP/IP по интерфейсу Ethernet следует:

1. Проверить состояние сетевого подключения Ethernet на ПК.

2. Соединить кросс-кабелем Ethernet контроллер и ПК. Если соединение установлено успешно, на экране ПК появится соответствующее сообщение, индикаторы Ethernet соединения на ПЛК начнут мигать.

Для начала работы с контроллером, подключенным по протоколу TCP/IP, следует настроить подключение контроллера к ПК:

1. Перенастроить сетевое соединение подключенного контроллера: ввести статический IP-адрес и маску подсети (см. раздел 6.6.1.2).
2. Выбрать **Онлайн → Параметры связи…** в главном меню. Откроется окно настройки интерфейсов связи с контроллером.
3. Нажать кнопку **New**. Откроется окно «Communication parameters: New Channel» – задания нового канала связи.
4. В окне задать имя нового соединения (например, TCP_IP_new) и выбрать из перечня интерфейс соединения: «Tcp/Ip (Level 2)» для связи по протоколу TCP/IP.

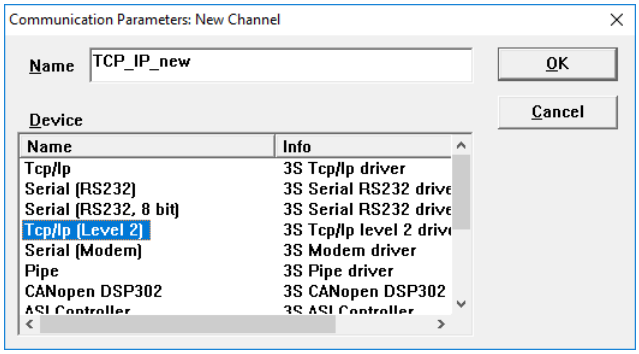


Рисунок 6.18 – Создание нового TCP/IP соединения

5. В параметре **Address** указать значение IP-адреса контроллера и сохранить новый канал связи.

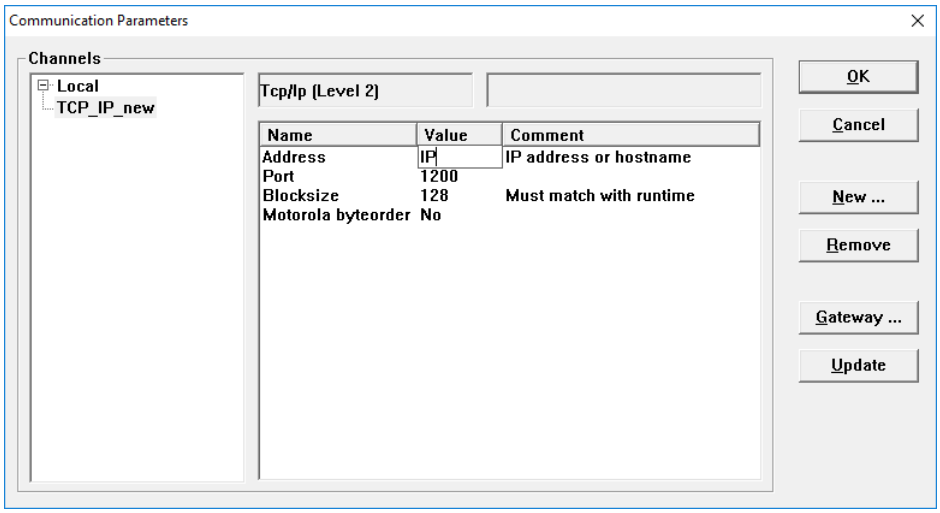


Рисунок 6.19 – Ввод IP-адреса

6. Нажать на кнопку **Gateway** и в параметре **Connection** поставить «Tcp\Ip», нажать кнопку OK.

После задания параметров соединения по интерфейсу Ethernet по протоколу TCP/IP следует перезагрузить контроллер отключением питания контроллера и повторным его включением через пять или более секунд.

6.6.2 Установка связи с контроллером

Для установки связи с контроллером требуется любая пользовательская программа (подойдет простейшая).

Простейшей пользовательской программой на языке ST является символ «;» (точка с запятой) – её будет достаточно для проверки связи с контроллером. Простейшие пользовательские программы на других языках приведены на рисунке ниже.

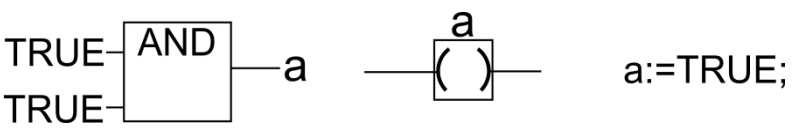


Рисунок 6.20 – Примеры простейших пользовательских программ на языках FBD, LD и ST

Для установки связи с контроллером следует:

1. Выбрать **Онлайн** → **Подключение** в главном меню. Предварительно должен быть снят флаг перед строкой меню **Онлайн** → **Режим эмуляции** (установка и снятие флага производится последовательными щелчками ЛКМ на строке). Перед установкой связи CODESYS скомпилирует пользовательскую программу, и в случае наличия в ней ошибок – прервет установку связи.
2. После установки связи CODESYS предложит загрузить или обновить код пользовательской программы в оперативной памяти контроллера (см. рисунок 6.21).

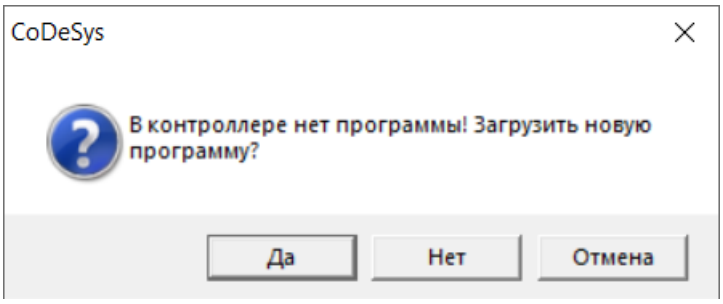


Рисунок 6.21 – Окно предложения загрузки программы

ПРИМЕЧАНИЕ
ПЛК110 поддерживает режим «ONLINE CHANGE», позволяющий обновить выполняющуюся в контроллере программу без прерывания ее работы (см. рисунок 6.22).

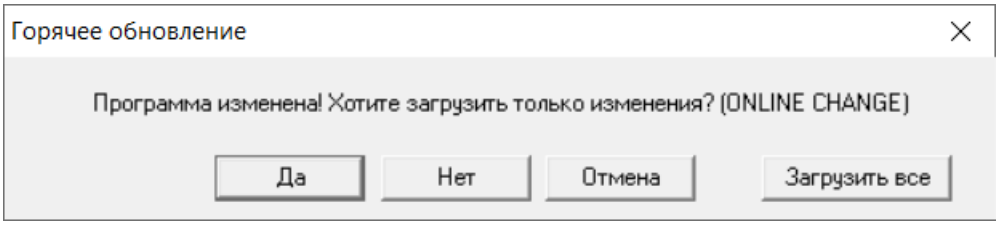


Рисунок 6.22 – Окно предложения обновления программы в режиме «ONLINE CHANGE»

6.6.3 Установка драйвера подключения ПЛК по USB Device

Для связи контроллера с ПК следует:

1. Скачать драйвер с сайта owen.ru из раздела CODESYS V2.

   ПРИМЕЧАНИЕ
   Для работы драйвера требуется ОС Windows 7/8/10.

2. Запустить установщик драйвера и следовать инструкциям мастера установки.
3. Соединить кабелем «USB A – B» разъем «USB B» на лицевой панели контроллера и любой свободный USB-порт ПК.
4. Подать питание на контроллер и на ПК. Если драйвер успешно установился, то в диспетчере устройств на ПК появляется новый порт «PLC110 USB Virtual Serial Port», таким образом подключение ПЛК к ПК распознается как добавление нового COM-порта к ПК с присвоением порту индивидуального порядкового номера (например, COM4).
5. Далее подключение к контроллеру осуществляется так же, как если бы контроллер подключался через физический COM-порт ПК.

В случае успешной установки драйвера и подключения контроллера с помощью порта «USB B» окно диспетчера устройств Windows отобразит новое устройство так, как показано на рисунке ниже.

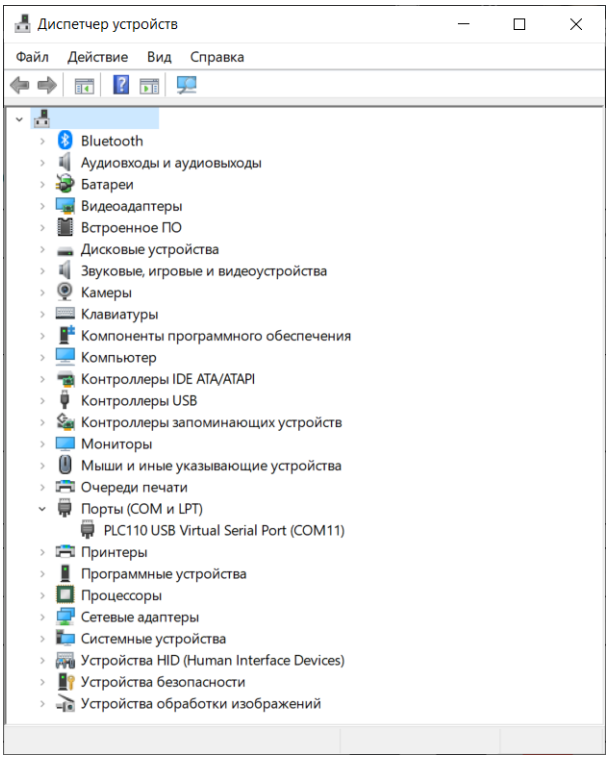


Рисунок 6.23 – Диспетчер устройств Windows после успешной установки драйвера

После установки можно сразу приступать к работе с CODESYS. Во время настройки подключения следует указывать номер COM-порта, соответствующий отображенному в диспетчере устройств (в рассмотренном примере, на рисунке выше это порт COM4).

6.7 Конфигурирование памяти ввода-вывода ПЛК

Перед написанием пользовательской программы рекомендуется настроить конфигурацию 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. Устройства обмениваются данными с пользовательской программой через специальную область памяти ввода-вывода. Конфигурация памяти ввода-вывода ПЛК задается в окне режима «Конфигурация ПЛК», которое вызывается на вкладке «Ресурсы» организатора объектов.

Размер памяти ввода-вывода определяется типом лицензии CODESYS контроллера ПЛК (см. раздел).

Подробное описание конфигурирования памяти ввода-вывода приведено в разделе 10.1.

Для подсчета объема памяти ввода-вывода, требуемой для работы с приборами компании «ОВЕН», следует воспользоваться данными из таблицы ниже.

Таблица 6.2 – Потребности ПЛК в памяти ввода-вывода, требуемой для работы с некоторыми приборами компании ОВЕН

Прибор	Объем требуемой памяти для протоколов передачи данных, байт					
	ОВЕН		Modbus		DCON	
	%I	%Q	%I	%Q	%I	%Q
ТРМ2хх (один аналоговый вход)	–	4	–	–	–	–
ТРМ151, ТРМ148, ТРМ133 (один аналоговый вход)	–	4	–	–	–	–
СП3хх (одна переменная на чтение с ПЛК)	–	–	–	2	–	–
СП3хх (одна переменная на запись с ПЛК)	–	–	–	2	–	–
МВ110-224.2А (один аналоговый вход)	4	–	4	–	4	–

Продолжение таблицы 6.2

Прибор	Объем требуемой памяти для протоколов передачи данных, байт					
	ОВЕН		Modbus		DCON	
	%I	%Q	%I	%Q	%I	%Q
МВ110-224.8А (один аналоговый вход)	4	–	4	–	4	–
МУ110-224.8И (один аналоговый выход)	–	2	–	2	–	–
МУ110-224.6У (шесть аналоговых выходов)	–	2	–	2	–	–

Рекомендации по расчету требуемой памяти ввода-вывода:

- в случае использования приборов других производителей, работающих по протоколам Modbus или DCON, следует по руководствам по эксплуатации на эти приборы определить сколько байт данных содержат команды, посылаемые по сети. Для работы с приборами ввода требуемое количество байт надо прибавить к размеру области %I, для работы с приборами вывода требуемое количество байт надо прибавить к размеру области %Q;
- 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;
- для приборов и операторских панелей, работающих по протоколу Modbus, одно значение параметра передается, как минимум, в двухбайтном регистре (даже если параметр – однобайтовый);
- в случае использования модуля архивации на каждую архивируемую переменную следует зарезервировать в памяти %Q место, равное размеру этой переменной;
- в случае использования модулей в режиме 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 в режиме Master одного сетевого протокола дополнительно требует от 4 до 8 байт;
- после подсчета необходимого размера областей памяти %I и %Q следует провести проверку достаточности объема доступной памяти каждого типа. Следует учитывать, что часть памяти занимается собственными входами и выходами. Если расчет показал, что резерва памяти нет, то следует использовать контроллер с лицензией «М», так как из-за принятого в CODESYS способа выравнивания адресов переменных в памяти ввода-вывода может возникнуть дополнительный расход памяти. Алгоритм выравнивания описан в разделе 10.7.3.1, но учитывать особенности выравнивания во время расчета потребности в памяти ввода-вывода не рекомендуется из-за сложности расчета.

6.8 Визуализация

CODESYS позволяет создавать окна-визуализации, в которых можно располагать визуальные элементы для графического отображения данных из пользовательской программы. Данные в визуализацию передаются из контроллера, при установленной с ним связи (подробнее см. раздел 6.6).

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

Если уровень заполнения емкости жидкостью доступен в пользовательской программе в виде значения 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. Также можно разместить кнопки запуска и остановки пользовательской программы.

Окно визуализации можно создать во вкладке «Визуализации» организатора объектов, нажав кнопку «Добавить объект» в контекстном меню строки «Визуализации».

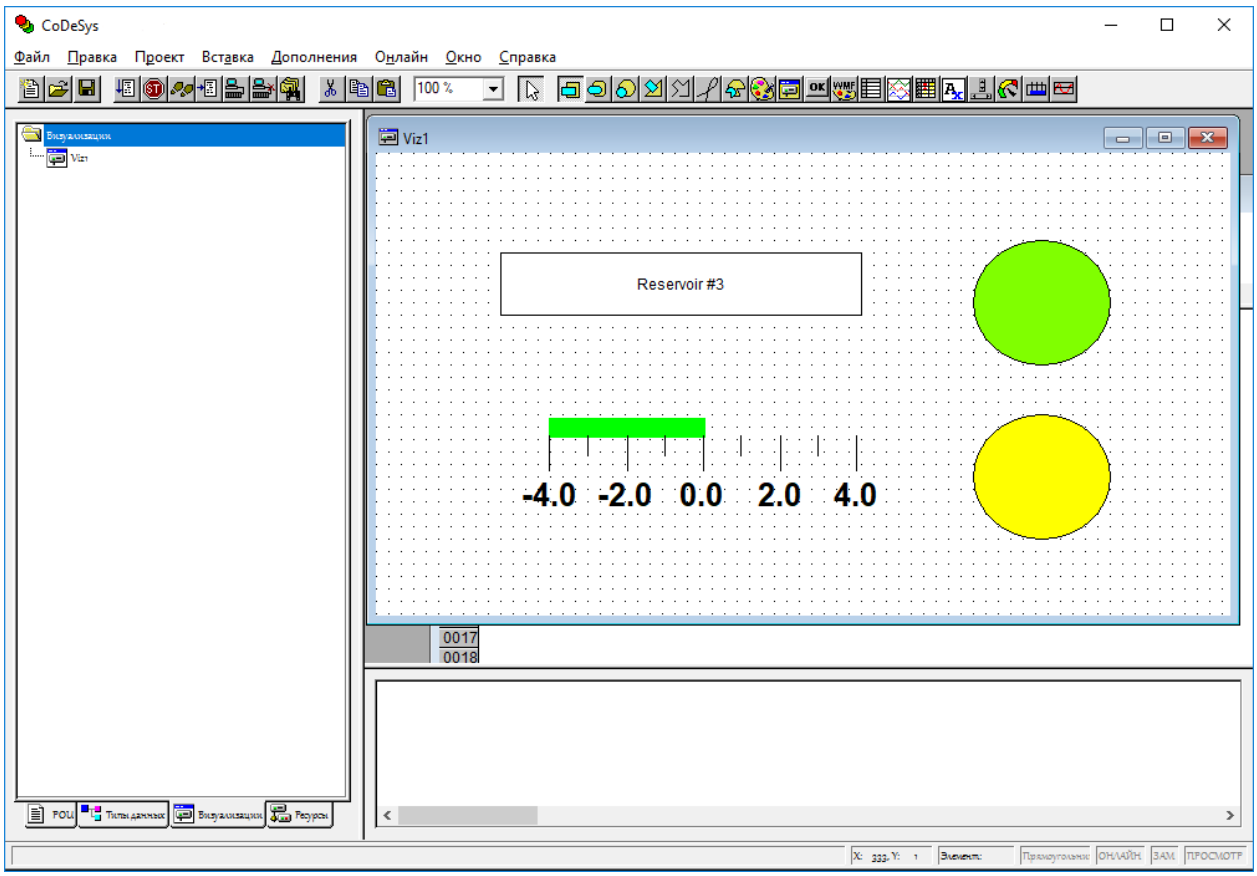


Рисунок 6.24 – Окно визуализации проекта

6.9 Сохранение проекта

Созданный проект можно сохранить в виде файла на жестком диске ПК для дальнейшей работы. Первоначально проект сохраняется вызовом команды **Файл** → **Сохранить как…** в главном меню, далее возможно сохранение вызовом команды **Файл** → **Сохранить** в главном меню или нажатием кнопки **Сохранить** () на панели инструментов.

ВНИМАНИЕ
Проект может быть также сохранен на встроенный в контроллер Flash-диск, что позволяет хранить проект непосредственно в контроллере и снижает вероятность его потери. Для загрузки проекта на встроенный Flash-диск контроллера следует после установки связи с контроллером (подробнее об установке связи см. раздел 6.6) выбрать команду **Онлайн** → **Загрузки исходных текстов** главного меню.

Проект может быть сохранен совместно с конфигурацией, т. е. со структурой, описанной в target-файле, загруженном при вызове проекта. Сохранение с конфигурацией 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. Рекомендуется сохранять проект в таком режиме, если предполагается возможность редактирования проекта по прошествии значительного времени с момента его создания. Режим сохранения проекта совместно с конфигурацией включается в окне «Конфигурация ПЛК» на вкладке «Ресурсы» организатора объектов, установкой флажка переключателя «Сохранять конф. файлы в проекте».

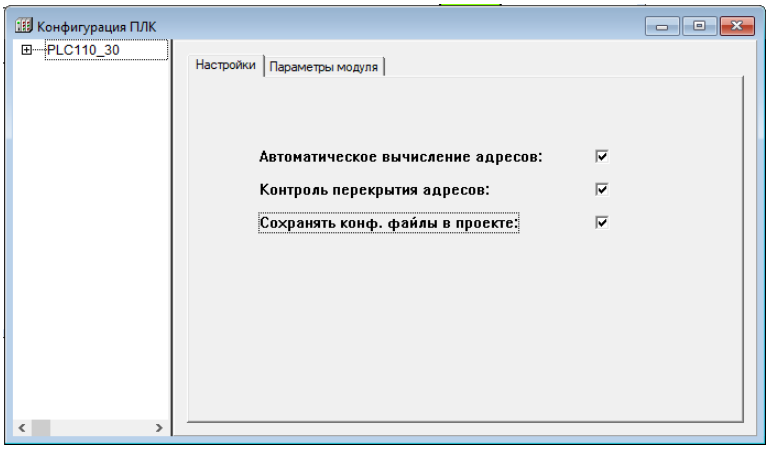


Рисунок 6.25 – Установка режима сохранения конфигурации в файл проекта в окне «Конфигурация ПЛК»

6.10 Индикация состояния контроллера

Во время работы с контроллером (программирование, отладка, работа с Flash-памятью) следует отслеживать его состояние по светодиодным индикаторам на передней панели корпуса контроллера.

Таблица 6.3 – Индикация контроллера

Индикатор	Состояние индикатора	Описание
«Работа»	Светится	Ядро ОС загружено успешно, пользовательская программа загрузилась и запустилась
	Не светится	Пользовательская программа не работает, остановлена или не загружена
	Слабо светится	Ядро ОС еще не загрузилось после включения питания прибора
	Мигает раз в 500 мс	Ядро ОС повреждено (не совпадает контрольная сумма)
	Мигает раз в 200 мс	Перегрузка центрального процессора
«Питание»	Светится	Наличие питания у контроллера
	Не светится	Отсутствие питания у контроллера
«Связь»	Светится	Наличие связи с CODESYS
	Не светится	Отсутствие связи с CODESYS
«Бат.»	Светится	Требуется замена батарейки
	Не светится	Замена батарейки не требуется

Также на передней панели контроллера расположены светодиодные индикаторы состояний дискретных входов и выходов. Каждый индикатор соответствует одному дискретному (как быстрому, так и обычному) входу или выходу. Во время работы контроллера индикатор светится в случае значения логической единицы на соответствующем ему входе или выходе и не светится, когда на входе или выходе значение логического нуля. Уровни напряжений, соответствующие логическим нулю и единице, приведены в *руководстве по эксплуатации* контроллера.

6.11 Запуск пользовательской программы (отладка)

Для отладочного запуска загруженной пользовательской программы следует выбрать команду **Онлайн** → **Старт** в главном меню.

ПРИМЕЧАНИЕ
Установка трехпозиционного переключателя «Работа – Стоп – Сброс» в положение «Работа» или «Стоп» на подключенном контроллере не приведет к запуску или остановке загруженной программы.

Запуск текущей пользовательской программы, записанной в ПЗУ, после выключения контроллера или сброс контроллера из CODESYS (сброс текущей пользовательской программы, записанной в ОЗУ, выбором команд **Онлайн** → **Сброс (горячий)** или **Онлайн** → **Сброс (горячий)** происходит,

если трехпозиционный переключатель находится в положении «Работа». Если трехпозиционный переключатель находится в положении «Стоп», то запуск или сброс не происходят.

ПРЕДУПРЕЖДЕНИЕ
Положение трехпозиционного переключателя анализируется контроллером при подаче питания или после сброса с помощью трехпозиционного переключателя.

Таблица 6.4 – Положения переключателя

Положение переключателя	Состояние прибора	Описание
Перевод в верхнее положение «Работа»	Во включенном состоянии	Пользовательская программа не запустится
	До включения	Со стартом контроллера запустится пользовательская программа, если она настроена на автозапуск*
Перевод в среднее положение «Стоп»	Во включенном состоянии	Пользовательская программа не будет остановлена
	До включения	Во время старта прибора пользовательская программа не будет загружена в ОЗУ контроллера и запущена**
Перевод в нижнее (нефиксируемое) положение «Сброс»	Во включенном состоянии	Через 6 секунд удержания произойдет перезагрузка прибора
	В отключенном состоянии	Ничего не произойдет

ПРИМЕЧАНИЕ
* Для настройки пользовательская программы на автозапуск следует **заранее** подключить контроллер к CODESYS и создать загрузочный проект (**Онлайн → Создание загрузочного проекта**).
** В случае попытки подключения к контроллеру из CODESYS будет выдаваться сообщение об отсутствии пользовательской программы.

После запуска загруженной пользовательской 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.

6.12 Сохранение пользовательской программы в памяти контроллера

После написания и отладки пользовательскую программы можно записать во внутреннюю Flash-память контроллера командой **Онлайн → Создание загрузочного проекта** в главном меню. Пользовательская программа сохраняется в памяти контроллера и после отключения питания или перезагрузки контроллера будет автоматически запускаться после перезагрузки или включения питания.

ВНИМАНИЕ
Ресурс встроенной Flash-памяти контроллера ограничен (около 50 000 циклов записи), поэтому не рекомендуется во время отладки пользовательской программы каждый раз записывать ее во Flash-память контроллера.

Если контроллер циклически перезагружается из-за ошибок в пользовательской программе, сохраненной во Flash-памяти, или некорректной записи программы во Flash-память, следует до или сразу после перезагрузки контроллера установить трехпозиционный переключатель в положение «Стоп». Пользовательская программа из Flash-памяти не будет автоматически запущена, что даст возможность подключиться к контроллеру и загрузить в него корректно работающую пользовательскую программу.

7 Разработка пользовательской программы

Приемы работы и примеры разработки пользовательских программ представлены в документе «Первые шаги с CoDeSys V2.3» на сайте owen.ru.

7.1 Программные компоненты (POU)

Пользовательская программа создается в CODESYS на любом из доступных языков программирования и может состоять из одного или нескольких программных компонентов (POU).

К программным компонентам (POU) относятся:

- программы;
- функции;
- функциональные блоки.

Отдельные POU могут включать действия (подпрограммы). Каждый программный компонент состоит из раздела объявлений и кода. Для написания всего кода POU используется только один язык программирования.

CODESYS поддерживает все описанные стандартом МЭК компоненты. Для использования стандартных элементов достаточно включить в проект библиотеку *standard. lib* (подробнее о библиотеках см. раздел 7.4).

Программные компоненты могут вызывать другие программные компоненты, но рекурсии недопустимы.

Главная программа, выполняемая циклически, должна называться **PLC_PRG**. В проекте могут быть определены несколько задач с различными условиями выполнения. Работа с задачами описана в разделе «Конфигуратор задач (Task Configuration)» документа «*Руководство пользователя по программированию ПЛК в CoDeSys V2.3*».

ВНИМАНИЕ
Если в окне «Конфигурация задач (Task Configuration)» определена последовательность выполнения задач (см. раздел 7.5), то проект может не содержать PLC_PRG. Нельзя удалять или переименовывать POU PLC_PRG в однозадачном проекте – если не используется «Конфигурация задач (Task Configuration)». В однозадачном проекте PLC_PRG является главной программой.

7.1.1 Программа

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, экземпляров программы не существует. Программа является глобальной во всем проекте.

Программу нельзя вызывать из функции. 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, что является главным различием между программой и функциональным блоком, в котором изменяются только значения переменных данного экземпляра функционального блока.

Список объявлений программы (в разделе объявлений окна редактирования) начинается с ключевого слова PROGRAM и следующего за ним имени программы.

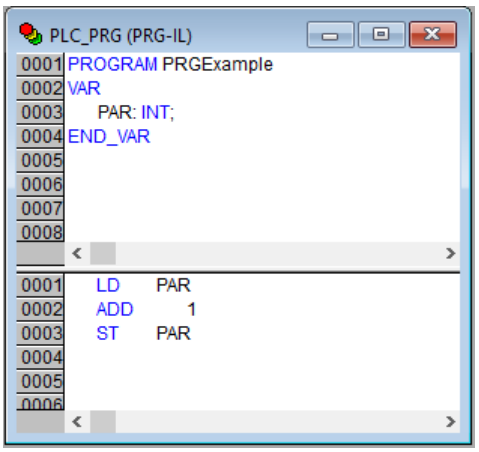


Рисунок 7.1 – Пример записи программы на языке IL

7.1.2 Функция

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 следует указать тип возвращаемого значения – после имени функции написать двоеточие и тип (см. рекомендации по наименованию в приложении документа «*Руководство пользователя по программированию ПЛК в CoDeSys V2.3*»). Правильно объявленная функция выглядит следующим образом:

```
FUNCTION Fct: INT;
```

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

Пример

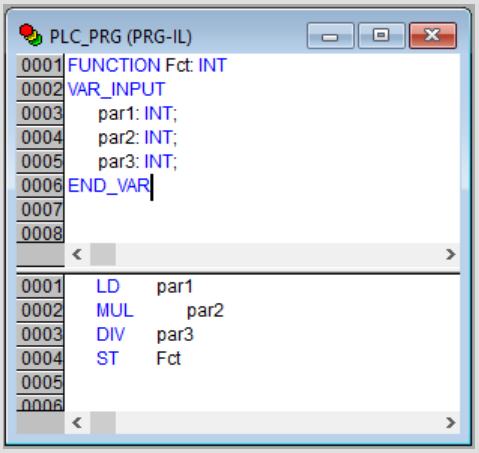


Рисунок 7.2 – Пример записи функции на языке IL

В функции, написанной на языке IL, используется три входных переменных (**par1**– **par3**) целочисленного типа (INT, диапазон изменения от –32768 до 32767) и возвращается результат деления произведения первых двух на третью. Список объявлений функции в разделе объявлений начинается с ключевого слова FUNCTION и следующего за ним имени функции, за которым, отделенное двоеточием, указывается название типа возвращаемого значения.

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

В языке SFC функция вызывается только из шага или перехода.

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

7.1.3 Функциональный блок

Функциональный блок – это программный компонент (POU), который принимает и возвращает произвольное число значений. В отличие от функции, функциональный блок не формирует возвращаемое значение. Список всех объявлений функционального блока в разделе объявлений начинается с ключевого слова FUNCTION_BLOCK и следующего за ним имени блока.

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

Пример

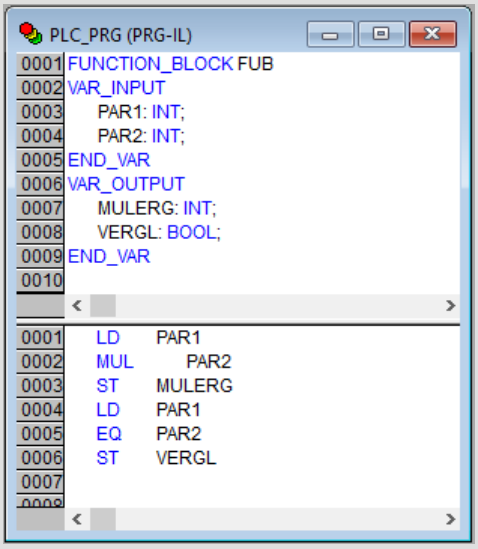


Рисунок 7.3 – Пример записи функционального блока на языке IL

В примере функциональный блок, написанный на языке IL, имеет две входных и две выходных переменных. Значение выходной переменной MULERG равно произведению значений двух входных переменных, значение VERGL определяется в результате сравнения значений входных переменных.

7.2 Переменные

Программные компоненты (POU) проекта обрабатывают переменные – величины, значения которых могут меняться в ходе выполнения пользовательской программы (в частных случаях переменные, обрабатываемые пользовательской 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.

7.2.1 Типы переменных

Переменные в CODESYS могут принадлежать к нескольким типам.

Переменные могут быть:

- **локальные** – используются только в рамках текущего программного компонента и задаются в разделе объявлений (см. раздел 6.1);
- **глобальные** – используются в рамках всего проекта (во всех программных компонентах, входящих в его состав) и задаются в разделе объявлений, вызываемом выбором объекта «Глобальные переменные» (Global Variables) на вкладке «Ресурсы» организатора объектов.

Переменные также могут относиться к **входным** или **выходным**, а также одновременно к **входным и выходным**.

Если необходимо сохранять значения переменных, то их можно объявить как **перманентные переменные** – такие переменные сохраняют свои значения при определенных сбоях в системе. Перманентные переменные бывают сохраняемые и постоянные.

Сохраняемые переменные обозначаются во время объявления ключевым словом RETAIN. RETAIN-переменные сохраняют свои значения, даже если произошла авария питания (выключение и

включение) контроллера, что равносильно команде «Сброс» (**Онлайн** → **Сброс**). Значения RETAIN-переменных сохраняются в энергонезависимой памяти контроллера.

ПРИМЕЧАНИЕ
Переменные, объявленные в окне «Конфигурация ПЛК» в подэлементе «ModBus (slave)», являются сохраняемыми RETAIN-переменными.

Пример

Сохраняемая RETAIN-переменная

```
VAR RETAIN
    rTemperature: REAL; (* Сохраняемая RETAIN-переменная *)
END_VAR
```

Контроллеры с версией ПО 1.0.x поддерживают следующие режимы работы с энергонезависимой памятью:

- **запись по событию** (используется по умолчанию) – RETAIN-переменные записываются автоматически по сигналу об отключении питания контроллера. Для записи используется накопленная энергия конденсаторов источника питания ПЛК;

  ВНИМАНИЕ
  RETAIN-переменные не сохраняются в следующих случаях:
  – срабатывание сторожевого таймера (WatchDog);
  – перезагрузка контроллера по команде из пользовательской программы или ПЛК браузера («PLC-Browser»).

  В энергонезависимой памяти контроллера остаются значения, записанные ранее.
- **циклическая запись** – RETAIN-переменные записываются циклично. Период устанавливается пользователем в пределах от 1 до 1000 секунд. Для надежности запись ведется поочередно в две копии RETAIN-переменных. В режиме циклической записи рекомендуется использовать цикл ПЛК не менее 10 мс, так как запись RETAIN-переменных вызывает дополнительную нагрузку на процессор контроллера.

Узнать или изменить активный режим работы с энергонезависимой памятью можно в ПЛК-Браузере («PLC-Browser») с помощью специальных команд:

- **SetupRetainMode** – просмотр активного режима работы с RETAIN-переменными;
- **SetCyclicMode XXX** – выбор режима циклической записи, где XXX – значение периода в секундах от 1 до 1000;
- **SetCyclicMode 0** – отключение режима циклической записи и переход к режиму записи по сигналу о пропадании питании контроллера.

ПРИМЕЧАНИЕ
Для применения настроек режима работы с энергонезависимой памятью следует перезагрузить ПЛК по питанию или командой **rebooT**.

Подробная информация и примеры проектов доступны в описании библиотеки RetainControlLib в разделе CODESYS V2 на сайте owen.ru.

Постоянные переменные обозначаются ключевым словом PERSISTENT. В отличие от сохраняемых переменных постоянные переменные сохраняют свои значения только в случае загрузки кода новой пользовательской программы, но не в случае выключения питания или сброса. Значения постоянных переменных размещаются вне энергонезависимого ОЗУ.

Подробнее о типах переменных см. *Руководство пользователя по программированию ПЛК в CoDeSys V2.3*.

7.2.2 Объявление переменных

Для использования в POU переменная должна быть **объявлена**. Объявление переменных производится в разделе объявлений (см. раздел 6.1). Синтаксис, используемый при объявлении переменных, соответствует стандарту МЭК 61131-3.

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

В разделе объявлений зарезервированные слова, типы данных и сами переменные автоматически подсвечиваются разными цветами.

Наиболее важные команды можно найти в контекстном меню, которое появляется по щелчку ПКМ или по нажатию сочетания клавиш **Ctrl + F10**.

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

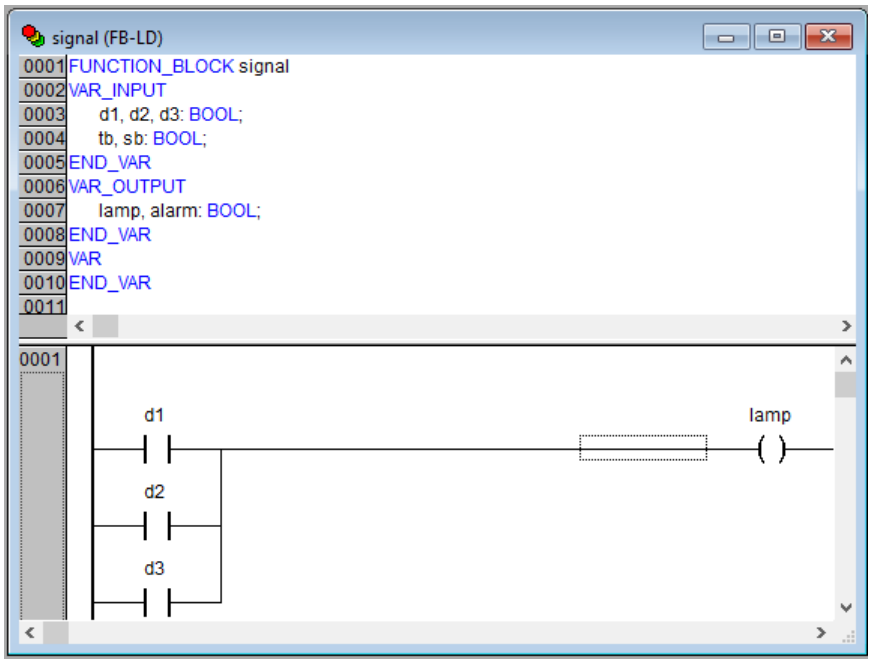


Рисунок 7.4 – Окно раздела объявлений (верхняя часть окна редактора POU)

7.2.3 Методы объявления переменных

В CODESYS применяются следующие методы объявления переменных:

- текстовый;
- табличный;
- автоматический.

Текстовый

Синтаксис объявления переменных текстовым методом (наименования заключены в квадратные скобки):

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

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 важен знак подчеркивания: переменные A_BCD и AB_CD считаются разными. Идентификатор не должен содержать подряд более одного символа подчеркивания. Длина идентификатора не ограничена, все символы являются значимыми.

Все переменные и типы данных можно инициализировать с помощью оператора «=». 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**.

ВНИМАНИЕ
Не рекомендуется использовать прямую адресацию при помощи ключевого слова **AT**. Для прямой адресации компилятор не проверяет за пользователем область памяти, на которую он ссылается при объявлении переменной. Переменным, размещаемым в области конфигурации ПЛК, следует присваивать имена непосредственно в области конфигурации. Дополнительное объявление переменных, объявленных в области конфигурации, не требуется.

Табличное

Табличный способ объявления переменных позволяет ускорить процедуру объявления переменных. Для вызова окна табличного объявления следует выбрать команду **Объявления в форме таблицы** контекстного меню окна раздела объявлений. 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.

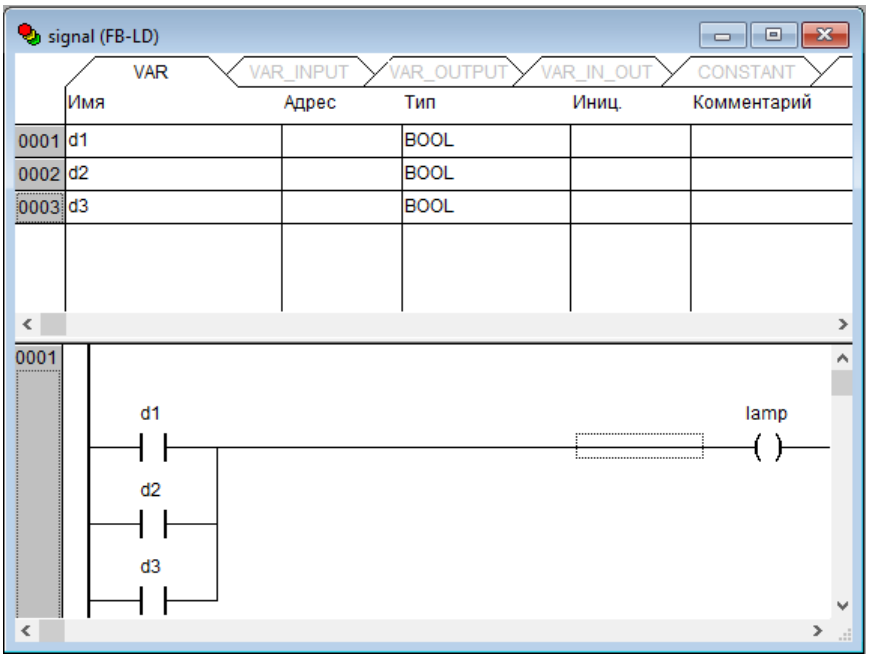


Рисунок 7.5 – Окно раздела объявлений в табличном представлении

Автоматическое

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 **Авто объявление…** в контекстном меню окна раздела объявлений.

В открывшемся окне задается имя добавляемой переменной (в поле «Имя»). В других полях окна значения задаются выбором из раскрывающегося списка или списков, отображаемых в специальных окнах.

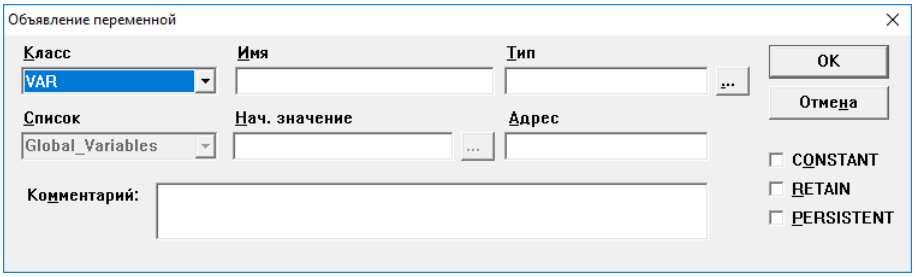


Рисунок 7.6 – Окно автоматического объявления переменной

Например, требуемый тип переменной можно выбрать в окне «Ассистент ввода», которое открывается по нажатию кнопки с тремя точками, размещенной у правого края поля «Тип».

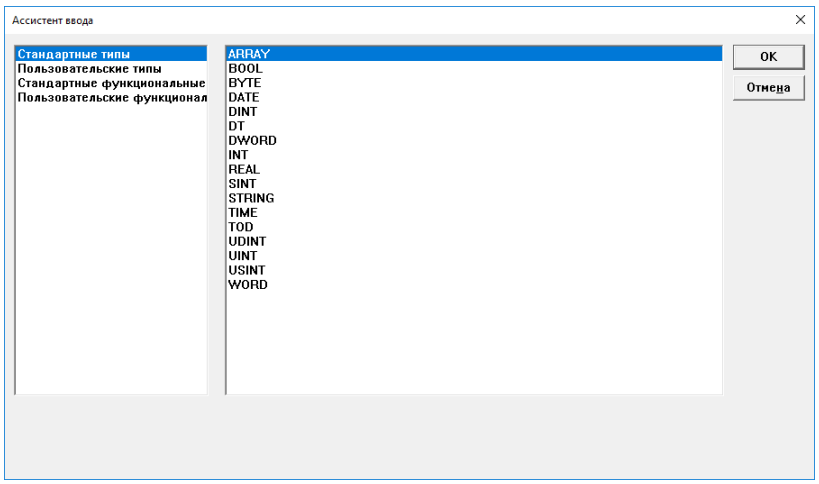


Рисунок 7.7 – Окно ассистента ввода типа переменной

7.3 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.

7.3.1 Базовые типы данных

Логический (BOOL)

BOOL – логический тип данных, который может принимать два значения – ИСТИНА (TRUE) или ЛОЖЬ (FALSE). Логический тип данных занимает 8 бит памяти (если не задан прямой битовый адрес).

Целочисленный

BYTE, **WORD**, **DWORD**, **SINT**, **USINT**, **INT**, **UINT**, **DINT** и **UDINT** – целочисленные типы данных, которые отличаются диапазонами сохраняемых данных и требованиями к памяти. Характеристики целочисленных типов данных приведен в таблице ниже.

Таблица 7.1 – 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

Рациональный

REAL – данные в формате с плавающей запятой, используются для сохранения рациональных чисел. Для рационального типа требуется 32 бита памяти.

Диапазон значений рационального типа от [1.175494351e-38] до [3.402823466e+38].

Строки

Строковый тип **STRING** представляет строки символов. Максимальный размер строки определяет количество резервируемой памяти и указывается во время объявления 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
Объявление строки размером до 35 символов:
```
str:STRING(35):='Просто строка';
```

Время и дата

Форматы данных времени и даты:

- **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**.

7.3.2 Пользовательские типы данных

Кроме стандартных типов данных в проекте можно использовать определяемые пользователем сложные типы данных: массивы, перечисления, структуры и некоторые другие (см. раздел 8).

7.4 Подключение дополнительных программных модулей

Дополнительные программные модули (библиотеки) подключаются в окне «Менеджер библиотек» (см. рисунок 7.8).

Окно «Менеджер библиотек» вызывается командой **Окно** → **Менеджер библиотек** или выбором пункта «Менеджер библиотек» в дереве на вкладке «Ресурсы» организатора объектов.

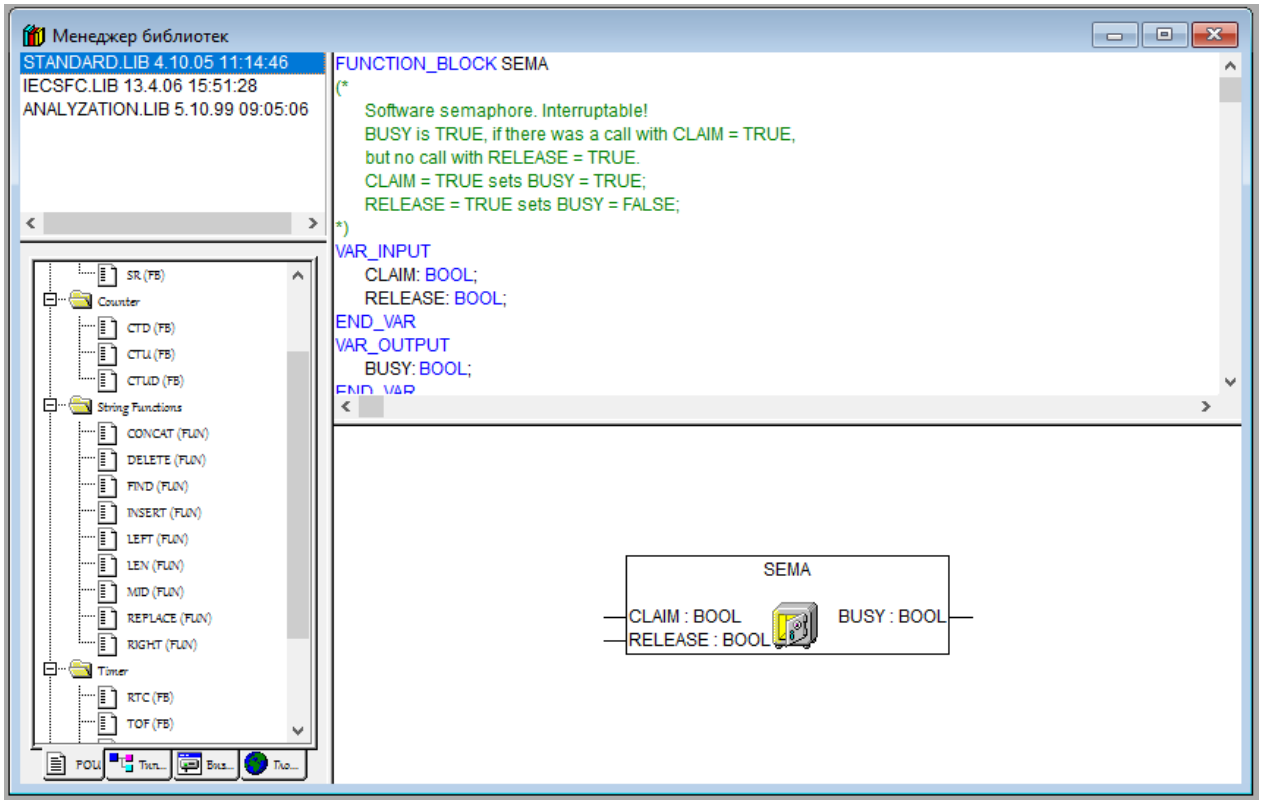


Рисунок 7.8 – Окно «Менеджер библиотек»

Для подключения библиотеки следует:

1. Выбрать команду «Добавить библиотеку» (Add library) в контекстном меню списка подключенных библиотек (отображаемого в верхней левой области окна «Менеджер библиотек») или команду **Вставка** → **Добавить библиотеку…** в главном меню.

2. В открывшемся окне выбора файлов 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 «Менеджер библиотек»).

Для удаления подключенной библиотеки следует:

1. Выделить требуемую библиотеку в списке подключенных (отображается в верхней левой области окна «Менеджер библиотек»).
2. Вызвать команду «Удалить» контекстного меню списка. Выделенная библиотека будет отключена от проекта.

Для включения в проект дополнительного программного модуля (то есть модуля, который содержится в подключенной к проекту библиотеке) следует:

1. Перейти на вкладку «POU» организатора объектов.
2. В дереве программных компонентов объекта выбрать требуемый компонент.
3. Вызвать команду **Правка** → **Ассистент ввода** в главном меню или команду «Ассистент ввода» в контекстном меню раздела объявлений.
4. В открывшемся окне «Ассистент ввода» (см. рисунок 7.9), в левой части, где отображается перечень доступных типов добавляемых объектов, выделить требуемый тип (в данном случае – «Стандартные функциональные блоки»). В правой части окна отобразится перечень доступных объектов выбранного типа. Если флажок переключателя «Структурно» в нижней части окна установлен, то перечень отображается в виде иерархического структурированного списка. В противном случае 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».

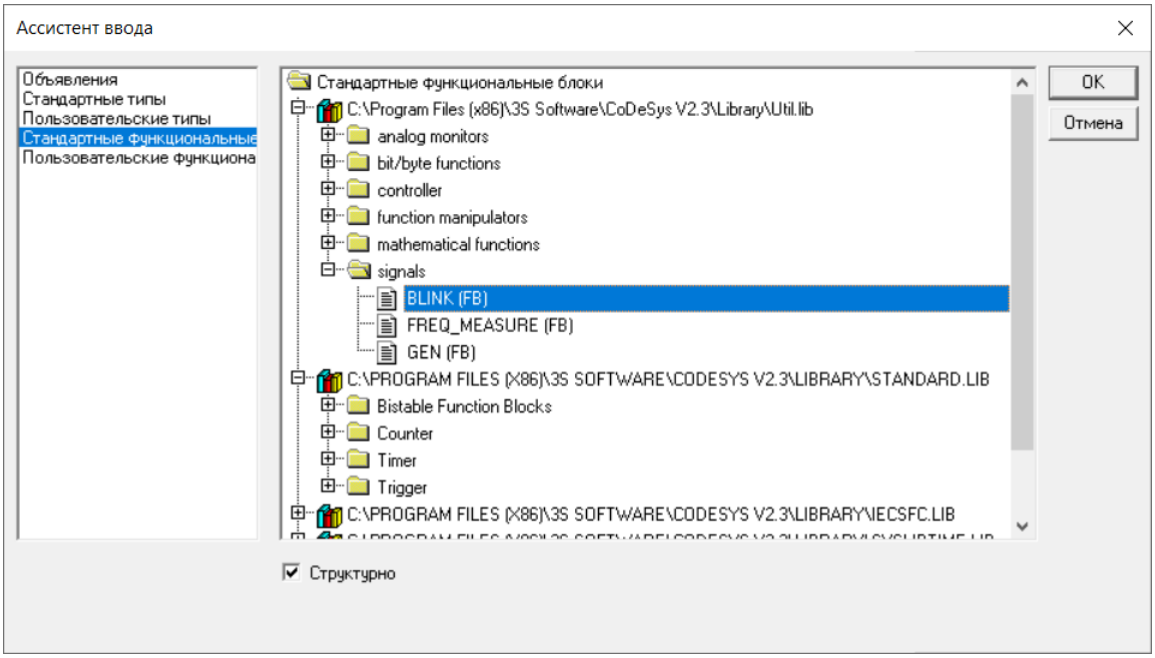


Рисунок 7.9 – Окно «Ассистент ввода»

Доступные дополнительные программные модули

Описания системных библиотек CODESYS доступны на сайте компании 3S Software и на странице CODESYS V2 на сайте owen.ru.

ПЛК110 поддерживают следующие библиотеки программных компонентов:

- OwenLibFileAsync.lib;
- OwenLibUSBSerial.lib;
- OwenLibNetControl.lib;
- OwenLibFactorySetups.lib;
- OwenLibHidEvent.lib;

- OwenLibPing.lib;
- Timer.lib;
- RetainControlLib.lib.

Рекомендуемые библиотеки:

- Standart.lib;
- Util.lib;
- SysLibTime.lib;
- SysLibCom.lib;
- SysLibProjectInfo.lib;
- SysLibMem.lib;
- SysLibSockets.lib;
- SysLibFile.lib;
- SysLibPorts.lib;
- NetVarUdp_LIB_V23.lib;
- ComService.lib;
- ModBus.lib;
- OwenNet.lib;
- Mercury.lib;
- UNM.lib;
- PID_Regulators.lib;
- SmsOwenLib.lib;
- OwenModbusSlave.lib;
- Oscat_basic_333.lib;
- Oscat_building_100.lib.

Установщик библиотек и их описания доступны на странице CODESYS V2 на сайте owen.ru. Помимо вышеперечисленных библиотек в состав установщика включены и некоторые сопутствующие библиотеки.

7.4.1 Модуль работы с файлами OwenLibFileAsync.lib

Модуль OwenLibFileAsync. lib поддерживает сохранение в файлы произвольных данных, чтение, удаление, копирование, переименование файлов и др. в асинхронном режиме, так как встроенное ПО контроллера в новой версии предусматривает работу с файловыми системами.

ПРИМЕЧАНИЕ
Библиотека OwenLibFileAsync.lib рекомендуется для всех новых разработок.

Файлы можно сохранять на следующие устройства хранения:

- внешний накопитель, подключенный к порту USB-Host (например, Flash-память, или жесткий диск);
- внутренний Flash-накопитель;
- внутренний виртуальный RAM диск (64 килобайт).

На носителях поддерживаются подкаталоги, а также доступны операции с файлами в директориях и просмотр содержимого директорий.

Устройство, на которое файл будет записываться, задается с помощью префикса к имени файла:

- **ffs:** – для внутренней Flash-памяти;
- **ram:** – для виртуального диска
- **usb:** – для внешнего накопителя, подключенного к порту USB-Host.

Пример
Чтобы записать файл с именем «file1.txt» на разные носители, следует преобразовать его в «ffs: file1.txt» для записи во внутреннюю flash-память, «ram:file1.txt» – для записи на виртуальный диск и «usb:file.txt» – для записи на внешний накопитель.

На ПЛК запущен TFTP-сервер для передачи данных по протоколу TFTP (см. *RFC 1350 – THE TFTP PROTOCOL (REVISION 2)* http://tools. ietf. org/html/rfc1350). Следует обратить внимание, что только RAM-диск доступен для TFTP протокола, а также отсутствует функция разграничения доступа. Сервер доступен на всех интерфейсах ПЛК по соответствующим IP-адресам на порту 69. Подробный пример доступен в документе «*Руководство по работе с файлами на ПЛК по протоколу TFTP*», который находится на странице CODESYS V2 на сайте owen.ru.

ВНИМАНИЕ
Не рекомендуется использовать встроенную Flash-память для записи часто переписываемых файлов, так как ее ресурс ограничен (около 50 000 циклов записи). В создаваемых пользовательских программах для контроллера рекомендуется программировать сохранение файла с предпочтением внешнего накопителя при его наличии.

ВНИМАНИЕ
Виртуальный диск расположен в ОЗУ контроллера, поэтому все файлы, записанные под именами с префиксом «ram:», будут храниться в контроллере до первого его выключения. Для сохранения файлов следует их копировать на Flash-носители – внутренний или внешний.

ПРИМЕЧАНИЕ
Пользовательская программа может работать одновременно максимум с пятью файлами.

Во время работы с USB-Host следует учитывать:

- Суммарное потребление тока хабом и подключенными устройствами не должно превышать 0,5 А;
- USB-Host имеет функцию защиты от перегрузки и короткого замыкания. Срабатывание защиты приводит к выключению питания на USB-Host с последующими периодическими попытками восстановления питания;
- к ПЛК могут быть подключены USB MassStorage и USB HID устройства. Общее число подключенных USB MassStorage и USB HID устройств не должно превышать 1 для каждого типа. Остальные устройства игнорируются. Устройства инициализируются в порядке подключения. Если два USB MassStorage устройства уже подключены и подается питание на ПЛК, то порядок их инициализации непредсказуем;
- стек USB-Host поддерживает опрос не менее 1 устройства класса USB HID;
- с помощью библиотеки OwenLibHidEvent.lib можно получать сообщения от HID устройства, например мыши и/или клавиатуры;
- в ПЛК поддерживается класс USB MassStorage с файловой системой FAT (12, 16, 32). Ограничения на размер накопителя налагаются только ограничениями файловой системы FAT.

  ПРИМЕЧАНИЕ
  Рекомендуется использовать файловую систему FAT32.

Запись на USB MassStorage накопитель происходит без кэширования, т. е. для безопасного отключения накопителя следует:

1. Завершить все процедуры записи (остановить запись из программы через библиотеку OwenLibFileAsync и закрыть открытые файлы, дождаться завершения всех системных загрузок файлов, остановить работу модуля(ей) архивации, остановить опрос файлов через 0x20 функцию ModBus slave).
2. Дождаться прекращения активности на накопителе (если индикация активности присутствует) или выждать не менее 3 секунд.
3. Отключить накопитель.

Не рекомендуется подключать USB MassStorage устройства на базе жестких дисков и SDD без дополнительного внешнего питания, т. к. это может привести к перегрузке по питанию и циклическому включению/выключению внешнего диска.

ПРИМЕЧАНИЕ
Не гарантируется корректная работа ПЛК с USB устройствами, если последние обратно не совместимы с протоколом USB 1.1.

В контроллере реализован асинхронный механизм доступа к файлам из пользовательской программы. Асинхронный доступ гарантирует отсутствие задержек в выполнении пользовательской программы при доступе к файлам, в том числе расположенным на внешних носителях. Основной особенностью использования модуля **OwenLibFileAsync. lib** является выполнение функций в два этапа:

1. Подача команду для работы с файлом.
2. Проверка завершенности выполнения указанной команды и разрешение для подачи следующих команд для дальнейших операций с файлом.

Любая функция библиотеки **OwenLibFileAsync.lib** возвращает 2 значения:

1. Состояние запроса к асинхронной библиотеке:
   - ASYNC_PAUSED -1000 (*Система по своим внутренним причинам приостановила обработку асинхронных запросов*);
   - ASYNC_QUERY_FULL -1001 (*более 5 запросов в очереди*);
   - ASYNC_BLOCK_ACCESS -1002 (*Запрос к уже обрабатываемому объекту с другой функцией*);
   - ASYNC_GENERAL_ERROR -1003 (*фатальная ошибка*);
   - ASYNC_INVALID_HANDLE_ERROR -1004 (*Запрос к неоткрытому/открытому не через асинхронную библиотеку файлу*);
   - ASYNC_WORKING 32766 (*идет работа асинхронной библиотеки*);
   - ASYNC_DONE 32767 (* работа завершена -> смотрите значение return value *).
2. При получении ASYNC_DONE требуется посмотреть второе возвращаемое значение, уже самой функции файловой системы:
   ```
   returnvalue:POINTER TO DWORD;
   ```
   В этой переменной лежит возвращаемое функцией значение, и расшифровку значения следует смотреть в описании соответствующей функции обычной, синхронной библиотеки SysLibFile.lib.

Функции работы с файлами в асинхронном режиме:

- **OwenFileOpenAsync** – используется для открытия существующего или создания нового файла. Выход hFile (DWORD) сообщает дескриптор файла. Он используется другими функциональными блоками для работы с данным файлом;
- **OwenSysFileCloseAllOpenAsync** – функциональный блок закрывает все открытые файлы. Имена или дескрипторы файлов не нужно сообщать, поскольку они все уже известны системе;
- **OwenSysFileCloseAsync** – используется для закрытия файла. После закрытия файл освобождается для других процессов, дескриптор более не имеет значения;
- **OwenSysFileWriteAsync** – используется для записи данных в файл. Файл должен быть предварительно успешно открыт с помощью SysFileOpenAsync;
- **OwenSysFileReadAsync** – используется для чтения данных из файла. Файл должен быть предварительно успешно открыт с помощью SysFileOpenAsync;
- **OwenSysFileDeleteAsync** – удаление файла с заданным именем;
- **OwenSysFileGetPosAsync** – возвращает позицию (смещение от начала файла в байтах) записи и чтения в файл;
- **OwenSysFileSetPosAsync** – задает позицию записи и чтения в файл;
- **OwenSysFileEOFAsync** – возвращает TRUE, если текущая позиция чтения/записи находится в конце файла, иначе возвращает FALSE;
- **OwenSysFileGetSizeAsync** – возвращает размер файла с заданным именем;
- **OwenSysFileGetTimeAsync** – возвращает время создания, последнего доступа и последнего изменения файла с заданным именем;
- **OwenSysFileCopyAsync** – копирование файла с заданным именем в файл с другим именем;
- **OwenSysFileRenameAsync** – переименование (перенос) файла с заданным именем.

Функция OwenSysFileOpenAsync

Функция OwenSysFileOpenAsync возвращает значение типа DWORD и используется для открытия существующего или создания нового файла. Возвращаемое значение – дескриптор файла, либо «0» в случае ошибки. Дескриптор файла используется для доступа к открытому файлу другими функциями библиотеки. В некоторых случаях сообщение об ошибке имеет вид «16#FFFFFFFF» (в десятичной системе это число 4294967295 при интерпретации как беззнакового числа или число –1 при интерпретации как числа со знаком).

Входные переменные:

- **FileName** типа STRING – имя файла;
- **Mode** типа STRING – режим работы с файлом, может имеет следующие значения:
  - **w+** – если требуется открыть файл только для записи. Если файл существовал до начала записи, то он будет стерт и создан пустой файл с заданным именем;
  - **r** – если требуется открыть файл только для чтения;
  - **a** – аналогично **w+**, но если файл существовал до начала операции, данные будут дописываться в конец файла.

Пример

Открытие файла в режиме «a» и разрешение на переход к дальнейшим операциям с файлом на языке ST:

```
0:
res:=OwenFileOpenAsync(filename,'a',ADR(handle));
IF res=ASYNC_WORKING THEN
    state:=1;
END_IF

1:
res:=OwenFileOpenAsync(filename,'a',ADR(handle));
IF res=ASYNC_DONE THEN
    IF handle<>0 THEN
        state:=2;
    ELSE
        state:=0;
    END_IF
ELSIF res<0 THEN
    state:=0;
END_IF
```

Функция OwenSysFileCloseAsync

Функция OwenSysFileCloseAsync закрывает файл, открытый ранее функцией OwenSysFileOpenAsync, и возвращает значение типа BOOL, которое равно TRUE при успешном закрытии файла, иначе (например, если файл не был открыт) FALSE. Входным параметром является дескриптор закрываемого файла.

Входные переменные:

- **hFile** типа DWORD – дескриптор файла, число, которое возвратила функция OwenSysFileOpenAsync.

Пример

Закрытие файла в режиме «a» и разрешение на переход к дальнейшим операциям с файлом на языке ST:

```
res:=OwenFileCloseAsync(handle,ADR(result));
IF res=ASYNC_WORKING THEN
    state:=7;
ELSE
    state:=0;
END_IF

7:
res:=OwenFileCloseAsync(handle,ADR(result));
IF res=ASYNC_DONE THEN
    IF result=0 THEN
        state:=8;
    ELSE
        state:=8;
    END_IF
ELSIF res<0 THEN
    state:=8;
END_IF
```

Функция OwenSysFileWriteAsync

Функция OwenSysFileWriteAsync записывает данные в файл, открытый с помощью OwenSysFileOpenAsync, и возвращает значение типа DWORD – количество записанных байт данных.

Входные переменные:

- **File** типа DWORD – дескриптор файла, число, которое возвратила функция OwenSysFileOpenAsync;
- **Buffer** – адрес буфера, содержащего данные, которые необходимо записать в файл, число, которое возвратила функция ADR с аргументом – именем переменной-буфера, тип – массив, например, массив байт, или строка;

- **Size** типа DWORD – размер буфера в байтах, может использоваться функция SIZEOF с аргументом (именем переменной-буфера).

Пример

Запись значения «buffout» в файл в режиме «a» на языке ST:

```
2:
res:=OwenFileWriteAsync(handle,ADR(bufout),14,ADR(result));
IF res=ASYNC_WORKING THEN
    state:=3;
ELSE
    state:=6;
END_IF

3:
res:=OwenFileWriteAsync(handle,ADR(bufout),14,ADR(result));
IF res=ASYNC_DONE THEN
    IF result=14 THEN
        state:=4;
    ELSE
        state:=6;
    END_IF
ELSIF res<0 THEN
    state:=6;
END_IF
```

Функция OwenSysFileReadAsync

Функция OwenSysFileReadAsync считывает данные из файла, открытого с помощью функции OwenSysFileOpenAsync, и возвращает значение типа DWORD – количество считанных байт данных.

Входные переменные:

- **File** типа DWORD – дескриптор файла, число, которое возвратила функция OwenSysFileOpenAsync;
- **Buffer** – адрес буфера, содержащего данные, которые необходимо записать в файл. Число, которое возвратила функция ADR с аргументом – именем переменной-буфера. Тип – массив, например, массив байт или строка;
- **Size** типа DWORD – размер буфера в байтах, может использоваться функция SIZEOF с аргументом – именем переменной-буфера.

Пример функции аналогичен функции OwenSysFileWriteAsync.

Функция OwenSysFileDeleteAsync

Функция OwenSysFileDeleteAsync удаляет файл и возвращает значение типа BOOL: TRUE в случае успешного удаления или FALSE в случае ошибки.

Входная переменная **FileName** типа STRING – имя удаляемого файла.

Функция OwenSysFileGetPosAsync

Функция OwenSysFileGetPosAsync возвращает число типа DWORD – фактическое смещение в байтах от начала до текущей позиции в открытом файле с заданным дескриптором. Число определяет «место» в файле, откуда будет считана или куда будет записана информация, если операция чтения или записи будет произведена без дополнительного предварительного смещения позиции (см. функцию OwenSysFileSetPosAsync).

Входная переменная **File** типа DWORD – дескриптор открытого файла.

Функция OwenSysFileSetPosAsync

Функция OwenSysFileSetPosAsync устанавливает для открытого файла с заданным дескриптором позицию чтения (записи) с помощью заданного смещения и возвращает значение типа BOOL. Если позиция была установлена, то возвращается значение TRUE, иначе возвращается FALSE.

Входные переменные:

- **File** типа DWORD – дескриптор открытого файла, в котором необходимо задать текущую позицию чтения (записи);
- **Pos** типа DWORD – позиция чтения (записи), заданная смещением в байтах от начала файла.

Функция OwenSysFileEOFAsync

Функция OwenSysFileEOFAsync определяет, достигнут ли конец файла, и возвращает значение типа BOOL, равное TRUE, если текущим положением позиции чтения (записи) является конец файла, иначе FALSE.

Входная переменная **File** типа DWORD – дескриптор открытого файла, в котором проверяется достижение текущей позицией конца.

Функция OwenSysFileGetSizeAsync

Функция OwenSysFileGetSizeAsync возвращает размер файла в байтах в виде значения типа DWORD.

Входная переменная **FileName** типа STRING – имя файла.

Функция OwenSysFileGetTimeAsync

Функция OwenSysFileGetTimeAsync определяет значения даты и времени создания, последнего изменения и последнего доступа к файлу, и возвращает значение типа BOOL: TRUE в случае успешного завершения выполнения функции и FALSE в случае любой ошибки (например, доступа к файлу). Для хранения трех значений времени в одной переменной используется структура FileTime, приведенная ниже.

```
TYPE FILETIME
    STRUCT
        dtCreation:DT; (* Дата и время создания файла *)
        dtLastAccess:DT; (* Дата и время последнего доступа *)
        dtLastModification:DT; (* Дата и время последнего изменения файла *)
    END_STRUCT
END_TYPE
```

Входные переменные:

- **FileName** типа STRING – имя файла;
- **ftFileTime** типа POINTER TO FILE TIME – адрес структуры, в которую будут сохраняться считанные данные о времени создания файла, последнего доступа к нему и его последней модификации. Извлекается с помощью функции ADR.

Пример

Программа на языке ST, считывающая структуру со значениями времени создания файла:

```
VAR
    file_time:POINTER TO FILETIME;
    returnvalue:POINTER TO DWORD;
    w: FILETIME;
END_VAR

8:
    OwenFileGetTimeAsync(filename, file_time, returnvalue);
    w:=file_time;
ELSE
    state:=0;
END_CASE
```

Функция OwenSysFileCopyAsync

Функция OwenSysFileCopyAsync копирует один файл в другой (файлы задаются именами) и возвращает значение типа UDINT, в котором содержится количество действительно скопированных байт.

Входные переменные:

- **FileDest** типа STRING – имя копии файла (файл-приемник);
- **FileSource** типа STRING – имя копируемого файла (файл-источник).

Функция OwenSysFileRenameAsync

Функция OwenSysFileRenameAsync переименовывает файл и возвращает значение типа BOOL: TRUE в случае успешного переименования или FALSE в случае ошибки. Ошибка может возникать, например, если попытаться переименовать открытый файл.

Входные переменные:

- **FileOldName** типа STRING – текущее имя файла;
- **FileNewName** типа STRING – новое имя файла.

7.4.2 Модуль контроля выхода в интернет OwenLibNetControl.lib

Модуль-библиотека **OwenLibNetControl.lib** предназначена для управления сетевыми устройствами в ПЛК. Библиотека позволяет включать/выключать интерфейсы (если это разрешено), получать статус интерфейса, его тип, атрибуты и адреса интерфейса (MAC, IP, APN, телефонный номер и т. п.). Библиотека подключается аналогично стандартным библиотекам CODESYS.

В ПЛК интерфейсы Ethernet и PPP имеют, соответственно, номера 0 и 1.

Функция START_IFACE

Рисунок 7.10 – Графические отображение функции START_IFACE

Входная переменная **IFACE** типа DWORD – номер запускаемого интерфейса.

Выходная переменная **START_IFACE** типа BOOL – значение, указывающее на успешность выполнения команды. Если возвращается значение «0», запуск произошел успешно.

Функция STOP_IFACE

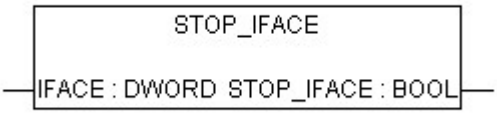


Рисунок 7.11 – Графические отображение функции STOP_IFACE

Входная переменная **IFACE** типа DWORD – номер закрываемого интерфейса.

Выходные переменные **STOP_IFACE** типа BOOL – значение, указывающие на успешность закрытия интерфейса. Если возвращается значение «0», интерфейс закрыт успешно.

Функция GET_IFACE_INFO

Рисунок 7.12 – Графические отображение функции GET_IFACE_INFO

Входная переменная **IFACE** типа DWORD – номер интерфейса.

Выходная переменная **GET_IFACE_INFO** типа POINTER TO IFACE_INFO возвращает указатель на структуру, содержащую информацию об интерфейсе:

```
TYPE IFACE_INFO:
STRUCT
    name: STRING(79); (* Имя интерфейса *)
    itype: IFACE_TYPES; (* Тип интерфейса *)
    addreses: ARRAY[0..9] OF ARRAY[0..31] OF BYTE; (* Все адреса интерфейса
(MAC, IP, имя порта) список зависит от типа *)
    atributes: IFACE_ATRIBUTES; (* Атрибуты интерфейса *)
END_STRUCT
END_TYPE
```

Поля структуры параметра **GET_IFACE_INFO**:

- **IFACE_TYPES** – типы интерфейсов, поле может принимать следующие значения:
  - *NO_IFACE 0* – нет интерфейса;
  - *ETHERNET_IFACE 1* – интерфейс Ethernet;
  - *PPP_IFACE 2* – использование модема.
- **IFACE_ATRIBUTES** – атрибуты интерфейса, поле может принимать следующие значения:
  - *HAVE_STATUS 1* – возвращает статус;
  - *AUTOSTART 2* – запускается автоматически;
  - *USE_DHCP 4* – настроен в режиме DHCP.
- **IFACE_ADDRESS_TYPES** – массив из 10 полей типа Array [0..31] of BYTES, каждый элемент массива содержит информацию об определенных адресах интерфейса:
  - *IFACE_MAC_ADDRESS 0* – массив, содержащий MAC-адрес;
  - *IFACE_IP_ADDRESS 1* – массив, содержащий IP-адрес;
  - *IFACE_IP_MASK 2* – массив, содержащий маску сети;
  - *IFACE_IP_GATE 3* – массив, содержащий шлюз сети;
  - *IFACE_TEL_NUMBER 4* – массив, содержащий телефонный номер текущего дозвона;
  - *IFACE_APN_NAME 5* – массив, содержащий адрес APN-оператора.

Функция GET_IFACE_STATUS

Рисунок 7.13 – Графические отображение функции GET_IFACE_STATUS

Входная переменная **IFACE** типа DWORD – номер интерфейса.

Выходная переменная **GET_IFACE_STATUS** типа IFACE_STATUS возвращает статус модема, битовые поля которого имеют следующие значения, по битам:

- *IFACE_NOT_PRESENT* – если возвращается 0, то интерфейс не запущен;
- *IFACE_PRESENT* – 0 – интерфейс запущен;
- *IFACE_CONFIGURED* – 1 – интерфейс настроен (есть настройки);
- *IFACE_CONNECTED* – 2 – есть физическое соединение;
- *IFACE_WORKING* – 3 – есть логическое соединение;
- *IFACE_SEND_DATA* – 4 – в течение 5 секунд была отправлена посылка;
- *IFACE_READ_DATA* – 5 – в течение 5 секунд была получена посылка;
- *IFACE_HAVE_ERROR* – 6 – есть ошибки;
- *IFACE_NO_IFACE* – если возвращается 16#FFFF, то такого интерфейса нет.

7.4.3 Модуль получения серийного номера USB-устройства OwenLibUSBSerial.lib

Модуль OwenLibUSBSerial. lib предназначен для получения статуса подключенных к USB-Host устройств и доступа к их серийному номеру. Модуль содержит только одну функцию – GETUSBSERIAL.

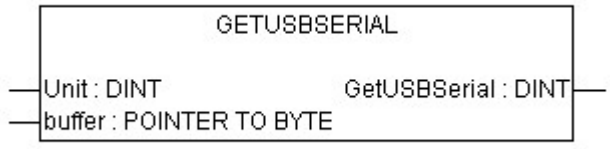


Рисунок 7.14 – Графическое отображение функции GETUSBSERIAL

Входные переменные:

- **Unit** типа DINT – номер устройства, обычно 0;
- **Buffer** типа POINTER TO BYTE – указатель на массив размером 24 байт, куда будет записан серийный номер.

Выходной параметр **GetUSBSerial** типа DINT возвращает значения:

- **0** – чтение серийного номера прошло успешно;
- **(–1)** – устройство USB не подключено;
- **(–2)** – серийный номер прочитан с ошибкой.

Расшифровка массива, содержащего серийный номер USB:

Таблица 7.2 – Значения массива

Offset	Field	Size	Value	Description
0	bLength	1	N+2	Size of this descriptor in bytes
1	bDescriptorType	1	Constant	STRING Descriptor Type
2	wLANGID[0]	2	Number	LANGID code zero
…				
N	wLANGID[x]	2	Number	LANGID code x

Расшифровка строки (вместо серийного номера):

Таблица 7.3 – Значения строки

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of this descriptor in bytes
1	bDescriptorType	1	Constant	STRING Descriptor Type
2	bString	N	Number	UNICODE encoded string

7.4.4 Модуль работы с HID-устройствами OwenLibHidEvent.lib

Модуль OwenLibHidEvent. lib предназначен для получения статуса подключенных к USB-Host HID-устройств. Общее число Mass Storage Device и HID-устройств не должно превышать 1 для каждого типа. Остальные устройства игнорируются. Инициализация устройств в порядке подключения.

Модуль содержит только одну функцию – GETHIDEVENT.

Рисунок 7.15 – Графическое представление функции GETHIDEVENT

Входная переменная **pHIDEv** типа POINTER TO BYTE – указатель на массив, в котором будут храниться данные. Размер массива должен быть равен 17 байтам.

Выходная переменная **GetUSBSerial** типа DINT возвращает указатель на массив, где хранятся данные:

- *Code:DINT; (*или xChange для мыши*) (*Для клавиатуры Code трактовать как DWORD*)*;
- *Value:DINT; (*или yChange для мыши*) (*Для клавиатуры Value трактовать как DWORD. Индицирует нажата/отпущена*)*;
- *WheelChange:DINT; (*как и x(y)Change относительное перемещение*)*;
- *ButtonState:DINT; (*биты нажатых кнопок*)*;
- *Event_type:BYTE; (*==1 - мышь ==2 - клавиатура*)*.

7.4.5 Модуль PING OwenLibPing.lib

Модуль OwenLibPing. lib предназначен для посылки команды «ping» по указанному адресу. Ответ ожидается в течение заранее установленного тайм-аута.

Функция SENDPING

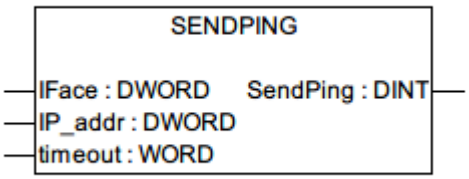


Рисунок 7.16 – Графическое представление функции SENDPING

Входные переменные:

- **IFace** типа DWORD – тип используемого интерфейса (0 – Ethernet, 1 – PPP);
- **IP_addr** типа DWORD – IP-адрес устройства, которое требуется пинговать (например, *16#0A020B32* – устройство с IP-адресом 10.02.11.50);
- **timeout** – предустановленный тайм-аут пинга в мс (минимум 50 мс, максимум 25000 мс).

Выходная переменная **SendPing** типа DINT возвращает статус пинга:

- *PING_SERVICE_READY:=1(*?*)*;
- *PING_SERVICE_IFACE_NOT_READY:=-1 (*интерфейс не поднят*)*;
- *PING_SERVICE_SENDING:=2 (*?*)*;
- *PING_SERVICE_TIMEOUT:=-2 (*таймаут ответа*)*;
- *PING_SERVICE_ANSV_RECEIVED:=3 (*?*)*;
- *PING_SERVICE_BUSY:=-3 (*запрос был послан,интерфейс занят*)*;
- *PING_SERVICE_DEST_UNREACHABLE:=-4 (*?*)*.

Функция GETPINGSTATUS

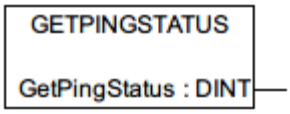


Рисунок 7.17 – Графическое представление функции GETPINGSTATUS

Функция не имеет входных переменных.

Выходная переменная **GetPingStatus** типа DINT возвращает статус. Если ошибка или *PING_SERVICE_ANSV_RECEIVED*, то после чтения статус сбрасывается в *PING_SERVICE_READY*.

7.4.6 Модуль таймера Timer.lib

Модуль Timer.lib используется для работы со встроенным таймером, по прерыванию которого может быть вызван отдельный программный элемент (POU), не связанный с выполнением основной программы ПЛК.

Подробное описание модуля находится в разделе 14 и на странице CODESYS V2 на сайте owen.ru.

7.4.7 Модуль RetainControlLib.lib

Модуль RetainControlLib. lib используется для принудительной записи RETAIN по команде из пользовательской программы.

ПРИМЕЧАНИЕ
Использование модуля RetainControlLib.lib возможно только в режиме записи по событию (SetCyclicMode установлен в значение «0»).

Модуль имеет одну функцию SAVENOW, которая служит для принудительной записи RETAIN-переменных.

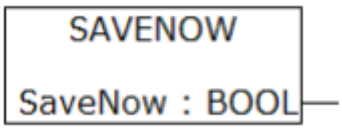


Рисунок 7.18 – Графическое представление функции SAVENOW

Функция не имеет входных переменных.

Выходная переменная **SAVENOW** типа BOOL – флаг, возвращаемый функцией по завершению записи.

ВНИМАНИЕ
Использование функции приостанавливает выполнение программы ПЛК до завершения полной записи RETAIN-переменных. Запись длится около 30 мс, поэтому рекомендуется вызывать функцию SAVENOW только когда значения RETAIN-переменных были изменены.

Пример

```
PROGRAM PLC_PRG

VAR
    xSaveRetain: BOOL; (* Переменная управления сохранением RETAIN.
                Запись происходит по переднему фронту *)

    xDone: BOOL; (* Флаг завершения записи *)
END_VAR

VAR RETAIN
    rSetPoint: REAL; (* RETAIN переменная, которую хотим принудительно
                                    сохранить *)
END_VAR

IF xSaveRetain = TRUE THEN
    xDone := SaveNow();
    IF xDone = TRUE THEN
        xSaveRetain := FALSE;
    END_IF
END_IF
```

7.4.8 Создание и использование дополнительных программных модулей

Пользователь может разрабатывать и применять дополнительные программные модули. 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 6.1).
2. В проекта создать объект типа «Функциональный блок» (см. рисунок 7.19).

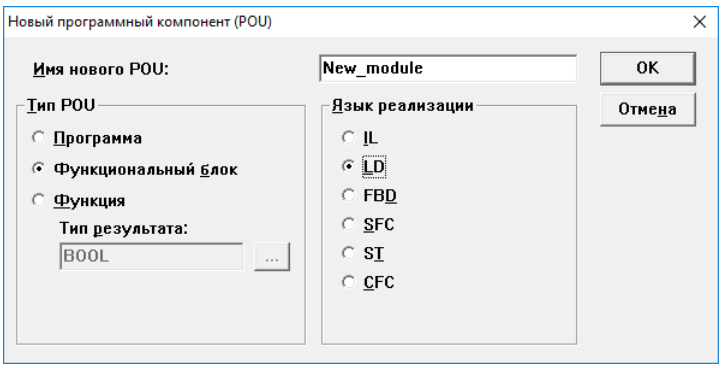


Рисунок 7.19 – Создание нового функционального блока

3. Написать программу функционального блока, которую предполагается использовать в качестве пользовательского программного модуля.

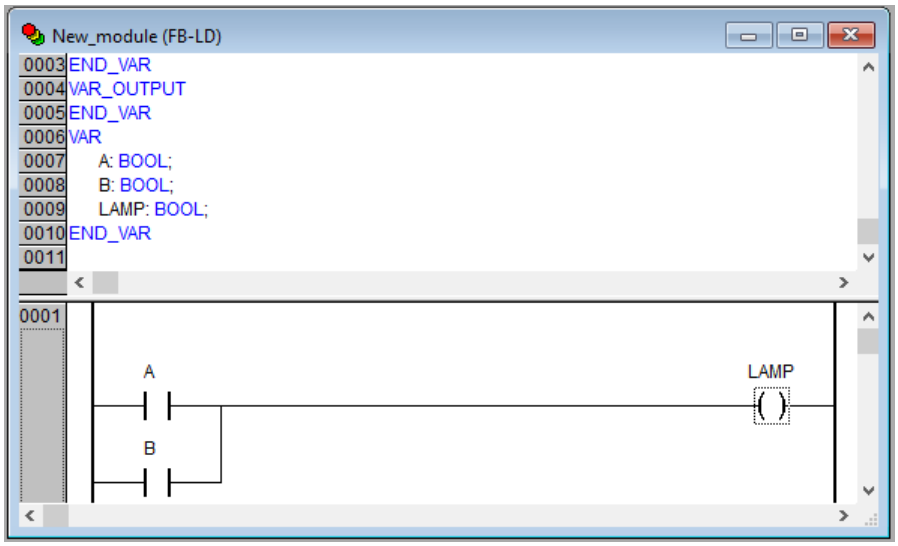


Рисунок 7.20 – Пример кода нового программного компонента

4. Удалить из проекта программный компонент PLC_PRG (команда «Удалить объект» контекстного меню объекта в дереве объектов), оставив в нем только созданный функциональный блок.
5. Сохранить функциональный блок командой **Файл** → **Сохранить как**, задав в открывшемся окне в поле «Тип файла» тип файла – «Внешняя библиотека (*.lib)» и нажав кнопку «Сохранить».
6. Сохраненный функциональный блок подключить к разрабатываемому проекту аналогично тому, как подключаются готовые программные модули (см. раздел 7.4).
7. 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 7.4).

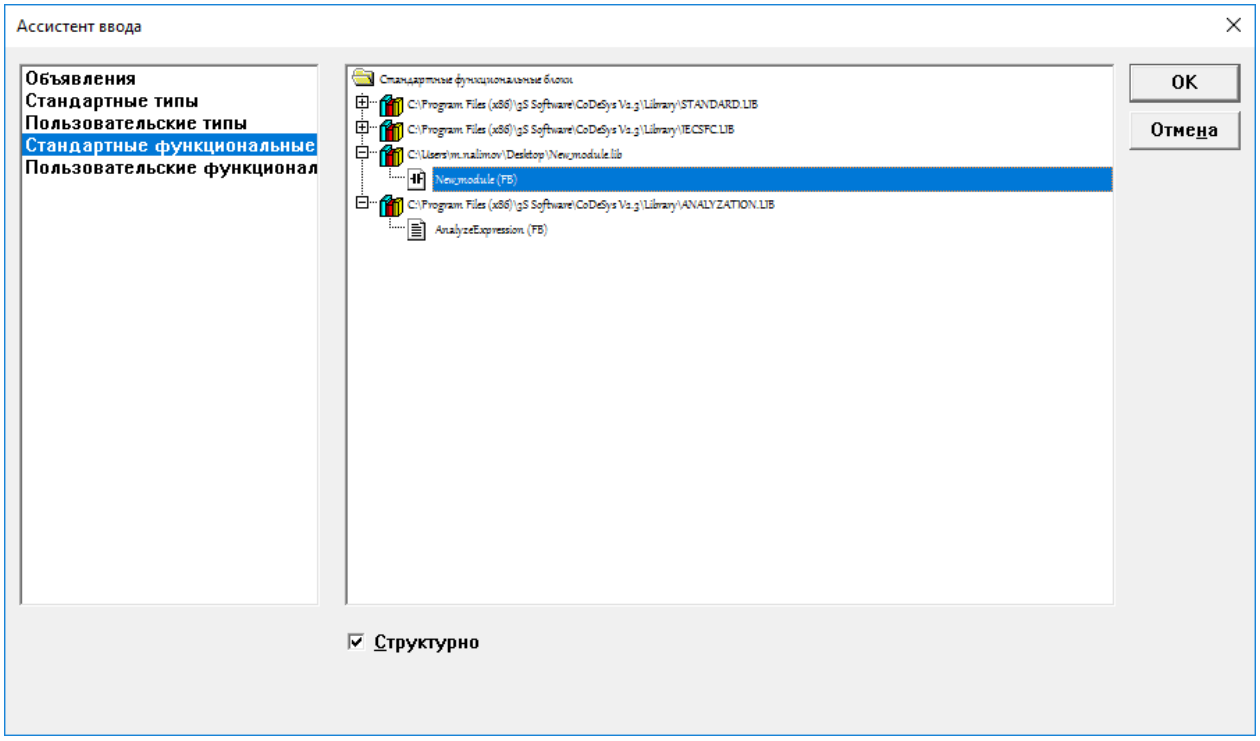


Рисунок 7.21 – Пользовательский модуль в окне «Ассистент ввода»

7.5 Многозадачность

По умолчанию в проекте всегда создается единственная «главная» программа PLC_PRG, выполняемая циклически (см. раздел 7.1). В проекте можно явно определить несколько **задач** с различными условиями выполнения. Задача – это единица обработки программы.

Каждая задача имеет:

- **Название** – служит идентификатором задачи;
- **Тип** – определяет условие вызова задачи. Условием может служить время (циклическое или свободное freewheeling выполнение), внутреннее или внешнее событие (например, превышение заданного порога глобальной переменной или прерывание в контроллере);
- **Приоритет** – задается числом (от 1 до 15) и в сочетании заданными условиями вызова задачи определяет хронологический порядок выполнения задач.

Для каждой задачи назначается ряд программ, которые будут в ней выполняться. Если задача выполняется в текущем цикле, то выполняются все включенные в нее программы (по одному циклу каждая). Порядок выполнения задач определяется комбинацией приоритетов и условий вызова задач.

Для каждой задачи можно задать контроль времени выполнения («сторожевой таймер»). Возможности его использования и настройки определяются целевой платформой.

Выполнение каждой задачи можно разрешить или запретить независимо от других.

7.5.1 Конфигурирование задач

Задачи определяются в окне «Конфигурация задач», которое открывается на вкладке «Ресурсы» организатора объектов.

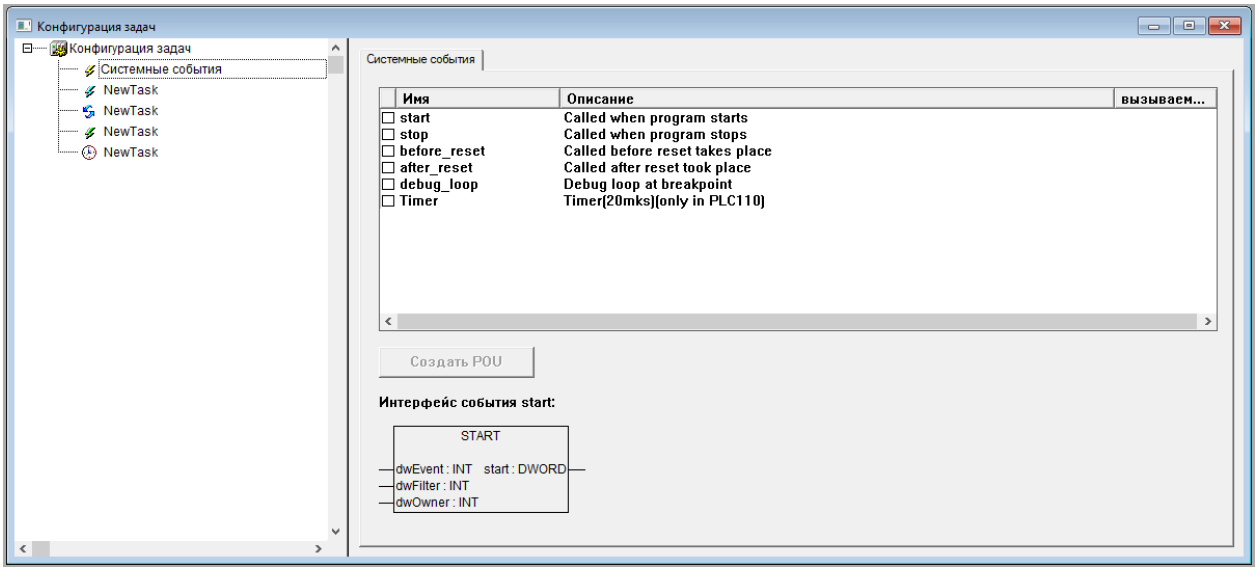


Рисунок 7.22 – Окно «Конфигурация задач»

В левой части окна отображается перечень задач текущего проекта в виде дерева конфигурации. В корневой позиции обязательно присутствует элемент «Конфигурация задач», под ним раскрывается список 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.

После добавления задачи в проект она включается в дерево задач и снабжается пиктограммой, отображающей тип задачи:

- – выполняемые по системным событиям (Старт, Стоп, Сброс);
- – циклически выполняемые задачи;
- – выполняемые по времени (свободному);
- – выполняемые по событию (связанному с глобальными переменными проекта);
- – 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), задавать связи с системными событиями (System events). Эта возможность зависит от выбора целевой платформы. Она должна быть поддержана в системе исполнения и разрешена в опциях целевой системы. Если стандартный набор настроек расширен специфическими параметрами, они будут представлены на отдельной вкладке «Parameter» в правой части окна.

ВНИМАНИЕ
Если в окне «Конфигурация задач» определена последовательность выполнения задач, то проект может не содержать PLC_PRG. В противном случае удалять или переименовывать программный компонент PLC_PRG нельзя. PLC_PRG является главной программой в однозадачном проекте.

Конфигурирование задач типа «Системные события», то есть выполняемых по одному из возможных системных событий включает:

- **Указание требуемого события** – установкой флажка в поле переключателя в требуемой строке списка системных событий;
- **Указание имени программного компонента (POU)**, который должен выполняться по наступлению события (ввод с клавиатуры в столбце «Вызываемый POU» в требуемой строке списка системных событий). Если POU с указанным именем не существует в текущем проекте, то активируется кнопка «Создать POU <Имя POU>». По нажатию этой кнопки автоматически формируется программный компонент проекта, имеющий указанное имя. Он может редактироваться так же, как другие POU.

Конфигурирование задач других типов включает:

- **указание типа задачи** – «Циклическая/Свободная/По событию/По внешнему событию;
- для задачи циклического типа – указание интервала выполнения;

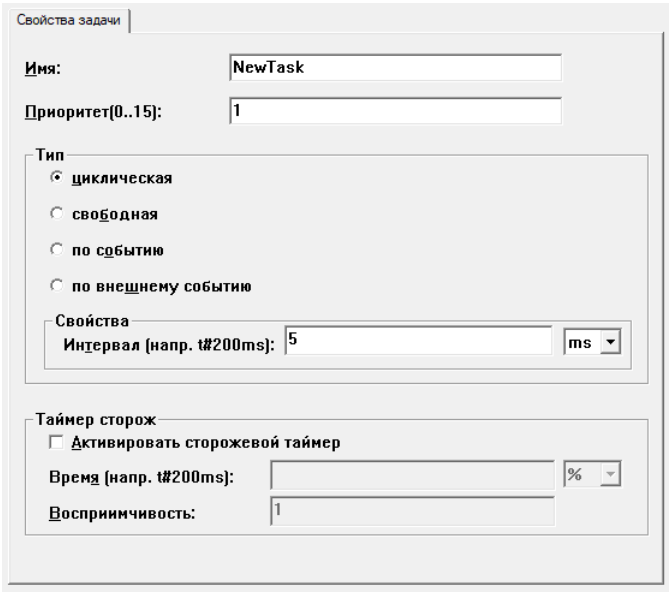


Рисунок 7.23 – Указание интервала выполнения для задачи циклического типа

- для задачи, выполняемой по событию – указание события, по нажатию кнопки у правого края поля открывается окно «Ассистент ввода», в котором можно выбрать требуемую переменную;

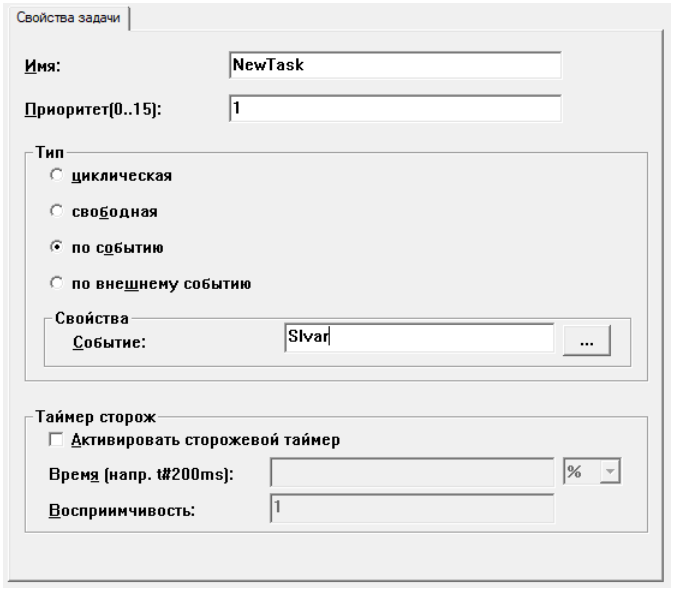


Рисунок 7.24 – Указание переменной для задачи, выполняемой по событию

- для задачи, выполняемой по внешнему событию – указание события, по нажатию кнопки у правого края поля открывается список задач, в котором можно выбрать требуемую задачу;

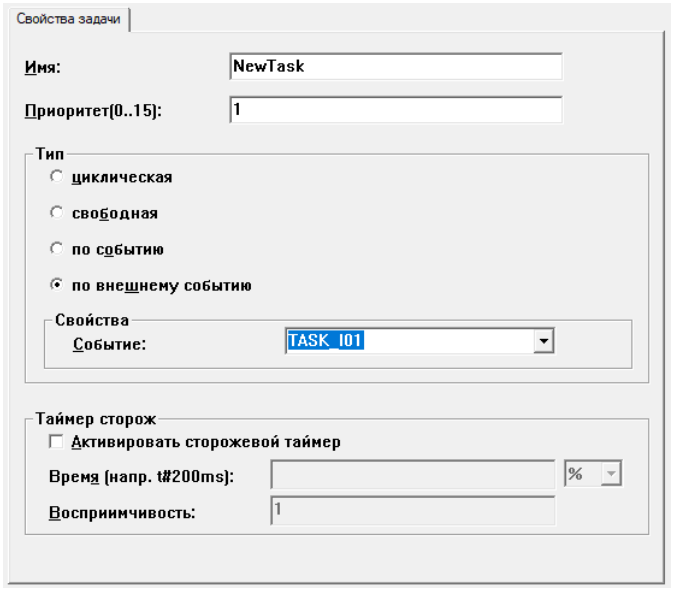


Рисунок 7.25 – Указание переменной для задачи, выполняемой по внешнему событию

- для задач любого типа задание параметров контроля времени выполнения («сторожевого таймера»), если эта опция доступна в используемом ПЛК.

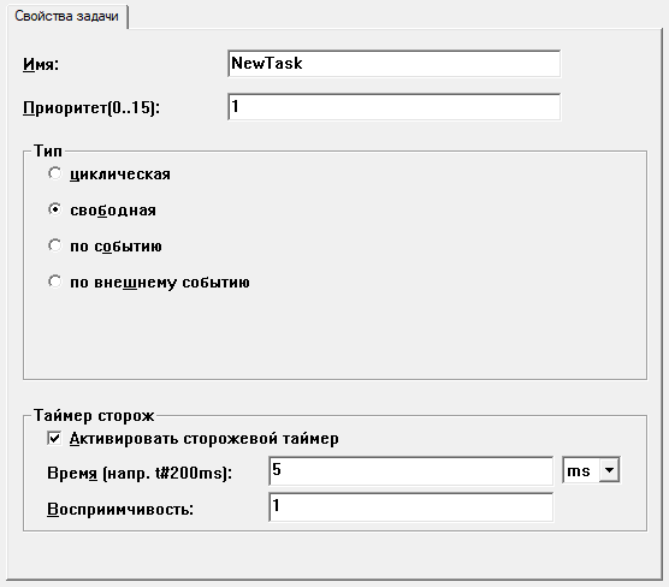


Рисунок 7.26 – Указание параметров сторожевого таймера

ПРИМЕЧАНИЕ
Не следует использовать одни и те же строковые функции в разных задачах, что может привести к ошибкам перезаписи данных.

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

7.5.2 Обработка событий

Системные события, которые контроллер способен обрабатывать (см. рисунок 7.22):

- **start** – вызов функции, если программа начала выполняться (после загрузки в ОЗУ автоматически,либо по команде пользователя). Функция выполняется до начала выполнения первого цикла программы;
- **stop** – вызов функции, если программа была остановлена. Функция выполняется сразу после получения системой команды остановки программы (из отладочной среды или с помощью управляющего тумблера на передней панели);
- **before_reset** – вызов функции, если был произведен горячий рестарт системы с помощью трехпозиционного переключателя (переключатель удерживался пять и более секунд в положении «Сброс»). Функция выполняется до перезагрузки контроллера;
- **after_reset** – вызов функции, если был произведен горячий рестарт системы с помощью трехпозиционного переключателя (переключатель удерживался пять и более секунд в положении «Сброс»). Функция выполняется после перезагрузки контроллера.
- **debug_loop** – вызов функции, если включен режим отладки и выполнение программы дошло до отладочной точки останова.
- **Timer** – вызов функции каждые 20 микросекунд.

Функции обработки событий должны иметь параметры, указанные во вкладке «Системные события» окна «Конфигурация задач». В случае с контроллером ПЛК110 функции обработки событий имеют одинаковый набор параметров: входными параметрами являются dwEvent, dwFilter и dwOwner типа INT, функция выдает значение типа WORD (см. Помощь по CODESYS – «Система программирования CODESYS» – «Ресурсы» – «Конфигуратор задач» – «Системные события»).

7.5.3 Отладка

Опция отладки заставляет компилятор формировать дополнительный код, упрощающий поиск ошибок. Опция «Отладочный код» включается установкой флажка переключателя «Отладочный код» в окне «Опции (Options)», вызываемом командой **Проект** → **Опции** главного меню на вкладке **Генератор кода**.

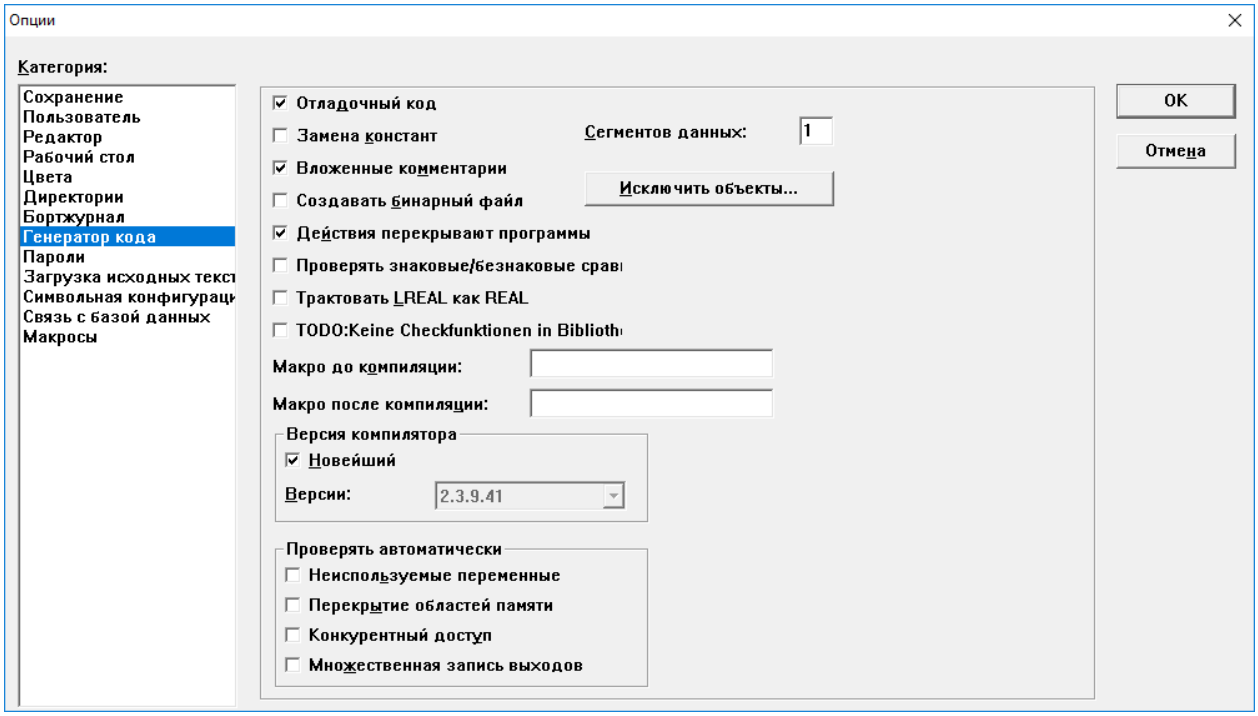


Рисунок 7.27 – Окно «Опции…», вкладка «Генератор кода»

7.5.4 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 языках FBD и LD – на графический элемент, в языке SFC – на шаг.

ВНИМАНИЕ
Система исполнения CODESYS SP32 Bit Full автоматически деактивирует сторожевой таймер задачи, если она выходит на точку останова.

7.5.5 Пошаговое выполнение

ПРИМЕЧАНИЕ
Подробнее о пошаговом выполнении можно узнать из встроенной помощи CODESYS:

1. Вызвать справку (**F1 или в главном меню → Справка → Содержание…**).
2. Выбрать в содержании **Система программирования CODESYS → Что есть что в CODESYS → Отладка и Online-функции**.

Пошаговое выполнение позволяет проверить логическую правильность программы. Под шагом подразумевается:

- **в языке IL** – выполнить программу до следующего оператора CALL, LD или JMP;
- **в языке ST** – выполнить следующую инструкцию;
- **в языках FBD, LD** – выполнить следующую цепь;
- **в языке SFC** – продолжить действие до следующего шага.

7.5.6 Выполнение по циклам

Команда **Онлайн → Один цикл** выполняет один рабочий цикл и останавливает контроллер после выполнения.

7.5.7 Эмуляция

Режим эмуляции последовательно включается и отключается выбором команды **Онлайн → Режим эмуляции** главного меню. Включенный режим маркируется установленным флажком в строке главного меню и записью «Эмул.» в строке состояния главного окна.

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

ВНИМАНИЕ
В режиме эмуляции функции внешних библиотек не выполняются.

7.5.8 Бортжурнал (Log)

«Бортжурнал (Log)» хронологически записывает действия пользователя, внутренние сообщения системы исполнения, изменения состояния и исключения в режиме «Online», что позволяет анализировать условия возникновения ошибки во время отладки программы.

Записи «Бортжурнала (Log)» можно просмотреть в режиме, вызываемом командой «Бортжурнал (Log)» дерева ресурсов проекта на вкладке «Ресурсы» организатора объектов.

8 Сложные структуры данных

Кроме стандартных типов данных (см. раздел 7.3.1) в проектах можно использовать определяемые пользователем сложные типы данных (массивы, перечисления, структуры и некоторые другие) – переменные или постоянные объекты, которые имеют внутреннюю структуру, доступную программисту. Переменные или постоянные объекты позволяют произвольно конструировать требуемые структуры данных из небольшого набора предопределенных типов.

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

8.1 Массив

Элементарные типы данных могут образовывать одномерные, двумерные 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 девятимерных («ARRAY[0..2] OF ARRAY[0..3] OF …»). Синтаксис:

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

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

8.2 Перечисление

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

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

Объявление перечисления должно начинаться с ключевого слова TYPE и заканчиваться ключевым словом END_TYPE. Синтаксис:

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

Переменная типа <Имя_перечисления> может принимать только перечисленные значения. Во время 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.

8.3 Структура

Структура создается командой «Добавить объект» контекстного меню во вкладке «Типы данных» организатора объектов. Новый объект отображается в дереве объектов, окно задания параметров объекта открывается в рабочей области главного окна.

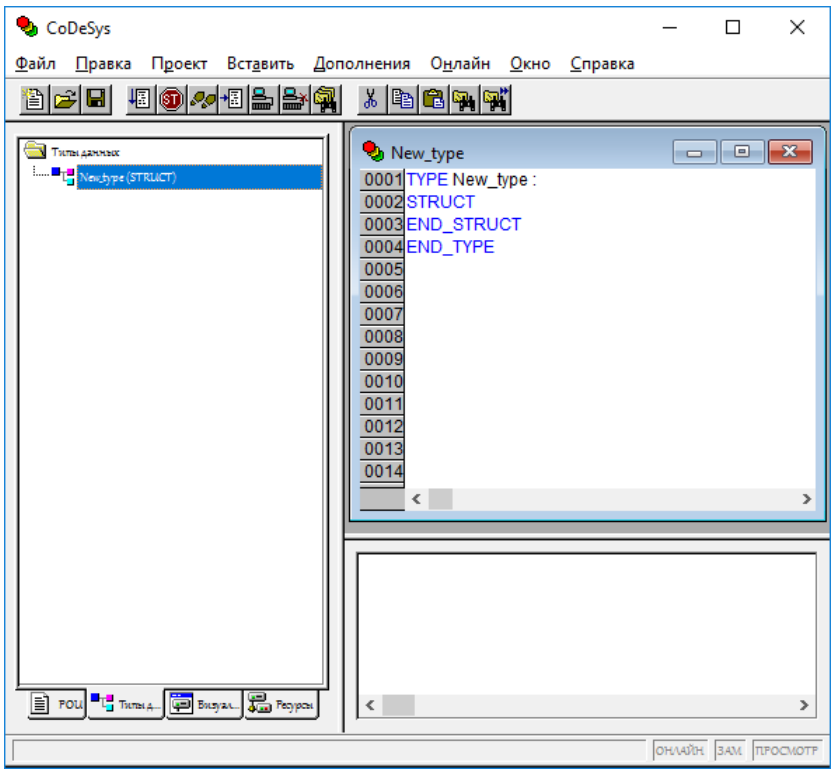


Рисунок 8.1 – Окно задания структуры

Объявление должно начинаться с ключевых слов TYPE и STRUCT и заканчиваться ключевыми словами 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.

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

Для доступа к элементам структуры используется следующий синтаксис:

```
<Имя_структуры>.<Имя_компонента>
```

Пример

Если структура «Week» содержит компонент «Monday», то обращение к нему будет выглядеть:

```
Week.Monday
```

8.4 Указатель

Указатель позволяет 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.

9 Визуализация проекта

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

Пример

Если в пользовательской 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 режиме «Online» представление элементов визуализации на экране изменяется в зависимости от значений переменных.

Свойства отдельных элементов визуализации, а также визуализации в целом устанавливаются в соответствующих диалоговых окнах конфигурации и в диалоговом окне свойств объекта, где определяется начальный вид элементов и выполняется привязка динамических свойств к значениям переменных проекта.

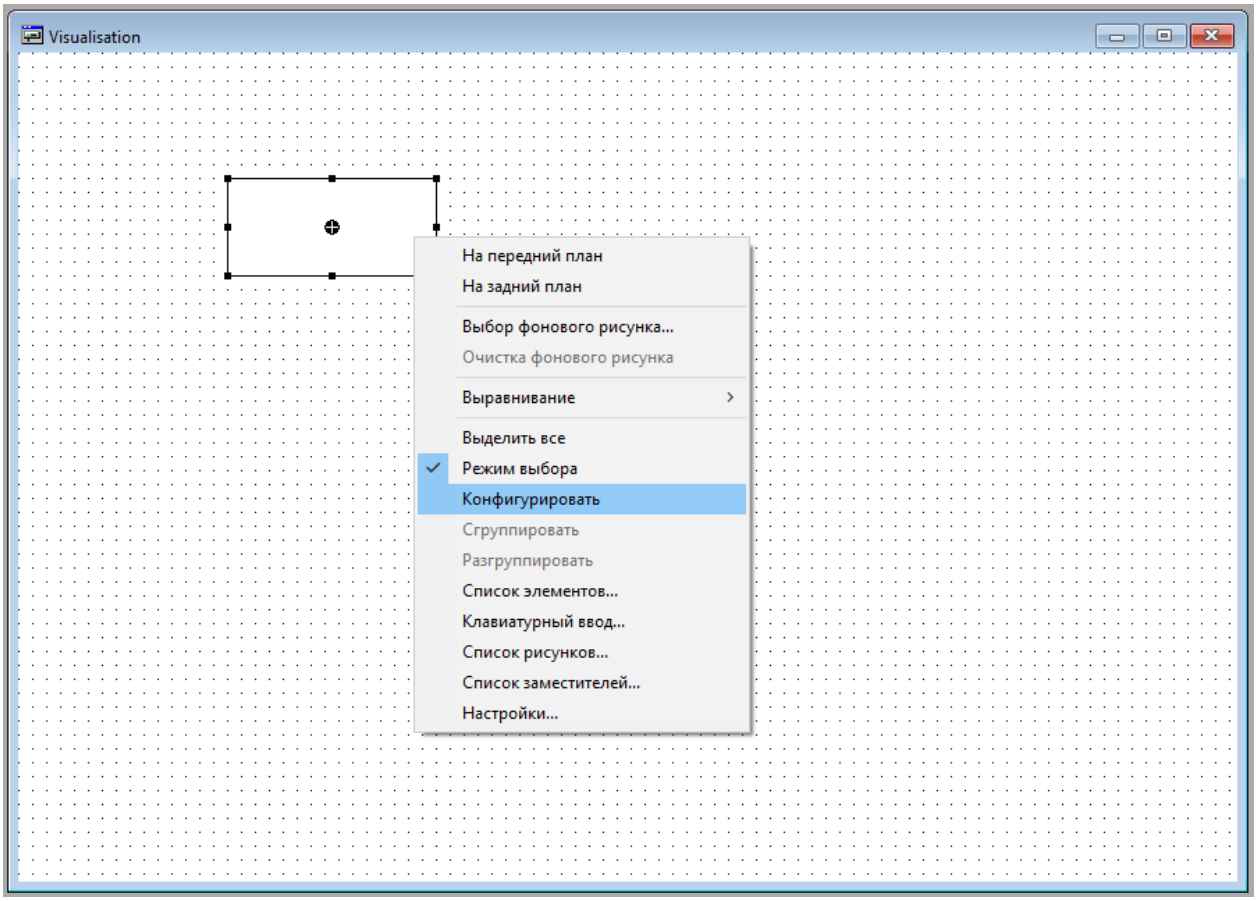


Рисунок 9.1 – Контекстное меню элемента визуализации

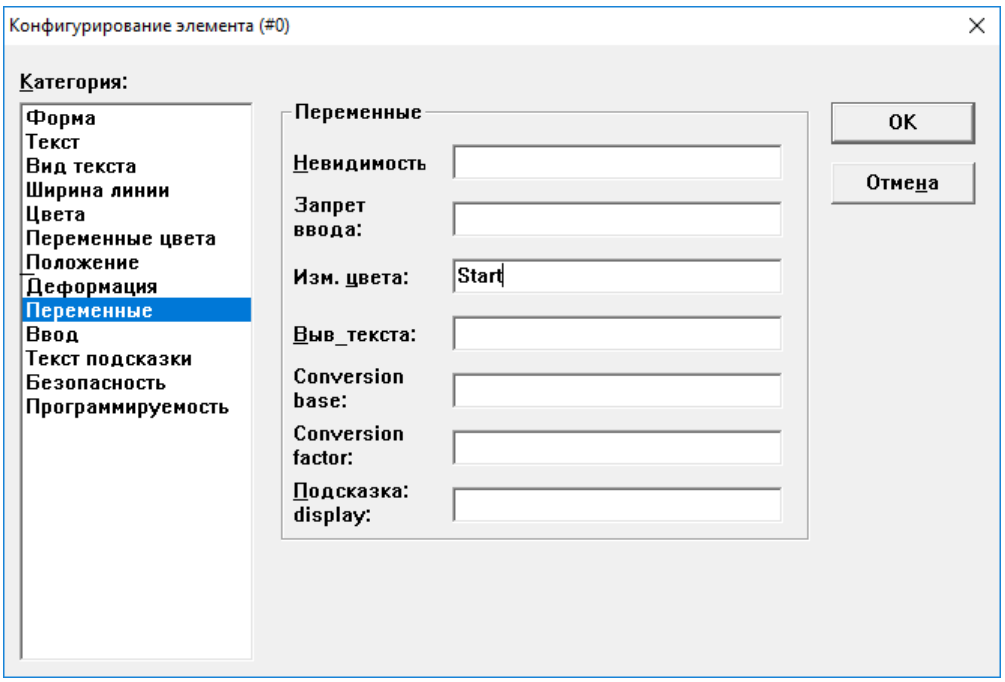


Рисунок 9.2 – Окно конфигурации элемента визуализации

Визуализация проекта может использоваться как пользовательский интерфейс для контроля и управления работой пользовательской программы в рабочем режиме. Для исключения возможности вмешательства оператора в работу программы визуализация может использоваться как единственный пользовательский интерфейс. Для этого ввод данных для пользовательской 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.

Способы использования созданной визуализации:

- ПО Win32 CODESYS HMI отображает формы визуализации на ПК в полноэкранном режиме. ПО Win32 CODESYS HMI не распространяется бесплатно в отличие от CODESYS;
- Web-визуализация отображает данные и предоставляет возможность удаленного управления через Интернет;
- для контроллеров со встроенным дисплеем доступна целевая визуализация.

Подробнее о создании окон визуализации см. документ «Визуализация CoDeSys V2.3. Дополнение к руководству пользователя по программированию ПЛК» на странице CODESYS V2 на сайте owen.ru.

9.1 CODESYS HMI

CODESYS HMI – это система исполнения визуализаций созданных в CODESYS.

Если проект содержит визуализацию, то во время запуска CODESYS HMI визуализация будет воспроизводиться в полноэкранном режиме. Пользователь сможет использовать заданные в программе функции управления и отображения с помощью мыши и клавиатуры, даже если проект CODESYS защищен от чтения.

Редактирование программ, меню и панели инструментов CODESYS недоступно пользователю, поэтому все элементы визуализации должны соответствовать необходимым функциям управления и отображения данных. Для этого в диалоге конфигурации элементов визуализации предусмотрены специальные возможности ввода для CODESYS HMI.

10 Конфигурирование контроллера

10.1 Память ввода-вывода

В процессе создания и отладки проекта следует 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 6.7).

Внешние устройства обмениваются данными с пользовательской программой ПЛК через область памяти ввода-вывода ПЛК (**%I** и **%Q**). Память ввода-вывода 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 ПЛК (см. раздел).

Память ввода-вывода настраивается в окне редактора «Конфигурация ПЛК», которое вызывается на вкладке «Ресурсы» организатора объектов.

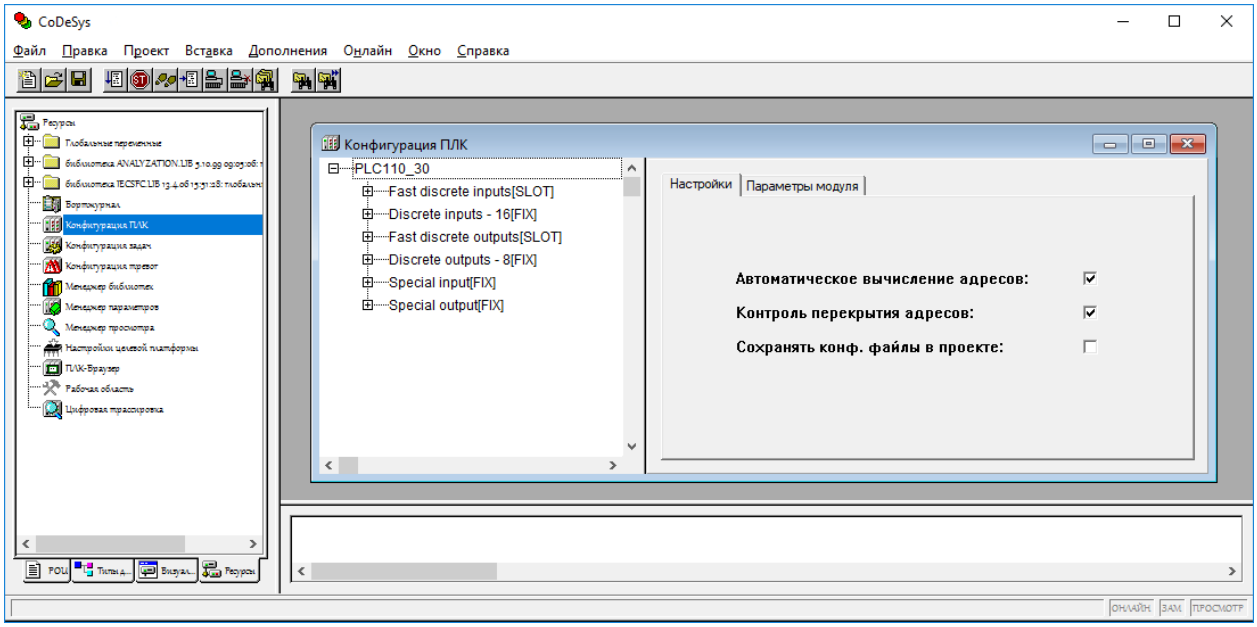


Рисунок 10.1 – Окно редактора «Конфигурация ПЛК»

Окно редактора «Конфигурация ПЛК» разделено на две части. В левой части окна отображается дерево конфигурации с ресурсами контроллера. Структура и компоненты дерева определяются файлом настроек целевой платформы (см. раздел) конфигурации, но могут быть изменены пользователем. В правой части окна отображаются параметры, доступные для текущего (выделенного) элемента дерева конфигурации. Параметры отображаются в виде одной или нескольких табличных вкладок (см. рисунок 10.2). 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 контроллера и подключенных к нему устройств.

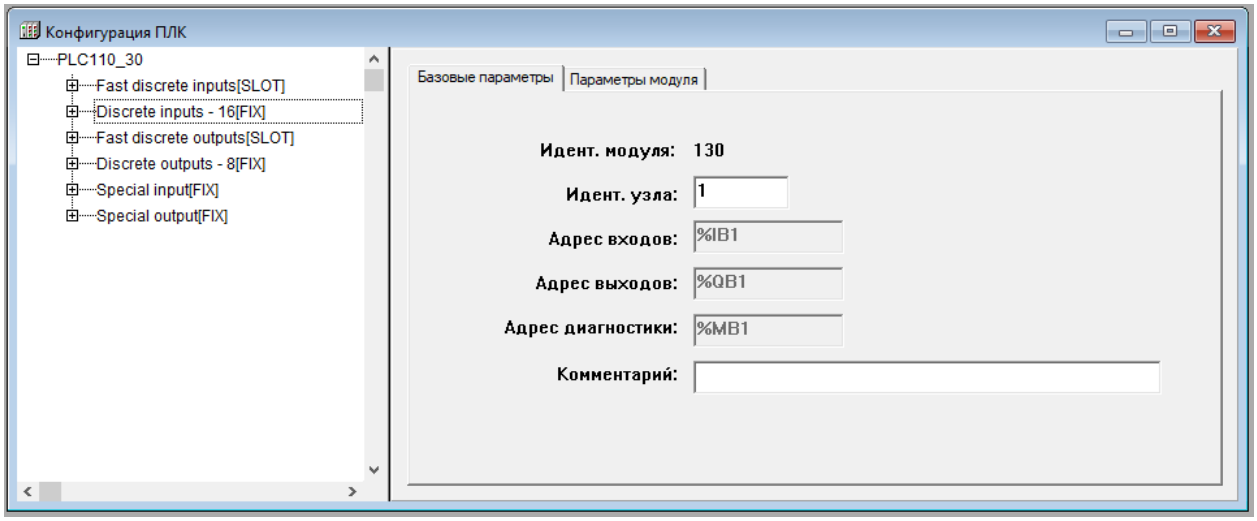


Рисунок 10.2 – Параметры в окне редактора «Конфигурация ПЛК»

ПРИМЕЧАНИЕ
Правая часть окна видна по умолчанию, но может быть скрыта выбором переключателя **Дополнения** → **Свойства** главного меню.

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

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 элементы:

- **Модуль (элемент конфигурации)** – независимая единица аппаратных средств. Модуль включает набор **каналов** ввода-вывода и (как и каждый отдельный канал) может иметь параметры. Каждый тип модуля имеет уникальный идентификатор. Модуль может иметь вложенные **подмодули** (подэлементы конфигурации);
- **Канал** – 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 ПЛК присутствуют модули, отвечающие за структурирование областей ввода и/или вывода, каждый из которых может содержать вложенные подэлементы (подмодули и каналы). Для каналов могут быть назначены символические имена. Прямые МЭК адреса отображаются в конфигурации для каждого символического имени.

Адреса каналов в области ввода-вывода ПЛК рекомендуется определять в автоматическом режиме установкой флажка переключателя «Автоматическое вычисление адресов» на вкладке «Настройки». В случае изменения положения модуля адреса его каналов соответствующим образом смещаются. Альтернативой может служить фиксированная адресация. В случае фиксированной адресации 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 V2.3» на сайте owen.ru.

Некоторые элементы конфигурации требуют самостоятельной настройки. Настройка может заключаться в добавлении и/или удалении модулей и подмодулей, а также в задании требуемых значений параметрам элементов конфигурации.

ВНИМАНИЕ
Добавление и удаление модулей конфигурации, а также настройка их параметров осуществляются при контроллере, отключенном от CODESYS. Для отключения контроллера следует вызвать команду **Онлайн** → **Отключение** главного меню или нажать кнопку **Отключение** () на панели инструментов.

ВНИМАНИЕ
Во время конфигурирования ПЛК следует иметь в виду, что можно изменять только значения переменных, лежащих в области вывода. Значения переменных из области ввода можно только считывать.

Если в процессе создания пользовательской 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 окно редактора «Конфигурация ПЛК» и выбрать команду **Дополнения** → **Стандартная конфигурация** главного меню.

Если предполагается переход от одного типа контроллера к другому, то переменные следует задавать в режиме («ресурсе») «Глобальные переменные». Во время задания стандартной конфигурации («Standard Configuaration») переменные, заданные в редакторе «Конфигурация ПЛК», пропадают, и ранее созданное распределение и именование переменных теряется. Глобальные переменные и их имена не будут потеряны, и в случае перехода к другому target-файлу достаточно только скорректировать адреса.

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

Для объявления глобальной переменной следует:

1. Открыть окно объявлений «Глобальные переменные» на вкладке «Ресурсы» организатора объектов.
2. В контекстном меню окна объявлений «Глобальные переменные» выбрать команду «Авто объявление…» (см. рисунок 10.3).

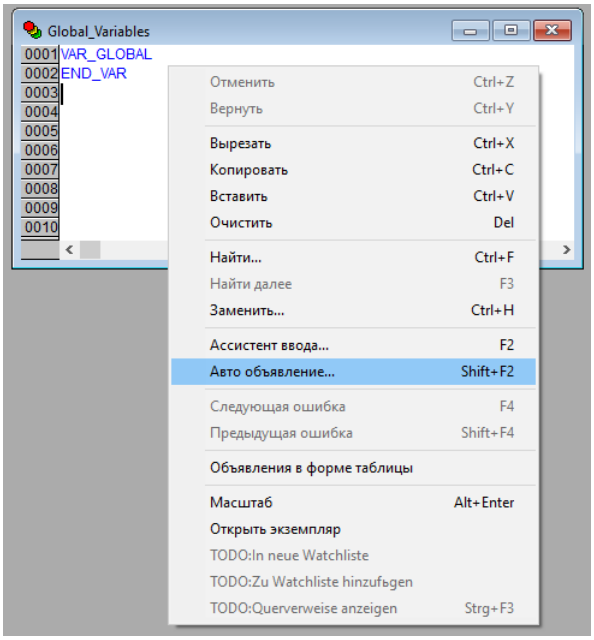


Рисунок 10.3 – Команда «Авто объявление…»

3. Задать и сохранить глобальную переменную в открывшемся окне автоматического объявления переменной (см. рисунок 10.4).

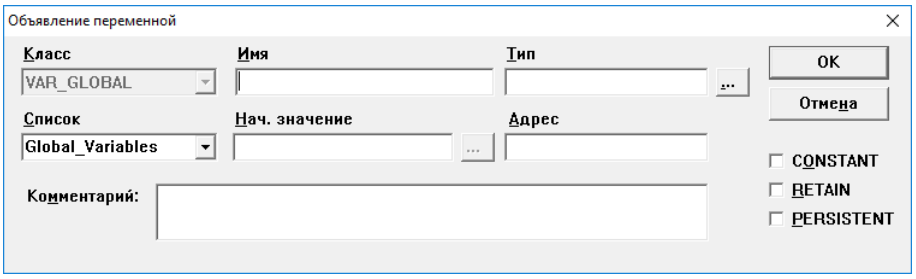


Рисунок 10.4 – Окно объявления переменной

Подробнее об объявлении и применении глобальных переменных см. раздел «Глобальные и конфигурационные переменные, файл комментариев» документа «Руководство пользователя по программированию ПЛК в CoDeSys V2.3», который доступен в разделе CODESYS V2 на сайте owen.ru.

10.2 Редактирование конфигурации ПЛК

Начальный вид конфигурации ПЛК задает файл конфигурации (**. cfg*) ПЛК, расположенный в директории, определенной в установленном целевом файле (target-файле) и считываемый во время открытия проекта в CODESYS.

Редактирование элементов конфигурации ПЛК заключается в выполнении операций над элементами дерева окна редактора «Конфигурация ПЛК» (добавление, замена и удаление модулей, подмодулей и каналов) и редактировании значений параметров элементов дерева в правой части окна.

10.2.1 Типы и виды модулей в конфигурации

Виды модулей в конфигурации:

- **Фиксированные модули** – заданы обязательно и не могут быть удалены или заменены. Доступно только редактирование их параметров;
- **Добавляемые модули** – добавляются (заменяются, удаляются) пользователем в процессе конфигурирования. Модули подразделяются на типы:
  - **SLOT** – для модуля зарезервировано место, которое может быть занято или оставлено пустым. На одно зарезервированное место может быть установлен один модуль;
  - **VAR** (свободный) – можно установить любое количество модулей (с учетом физических возможностей области ввода-вывода).

10.2.2 Добавление подмодуля (подэлемента)

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

Для добавления подмодуля (подэлемента) в текущую конфигурацию следует:

1. Выделить требуемый модуль (элемент) конфигурации и нажатием ПКМ вызвать контекстное меню.
2. Выбрать в контекстном меню требуемую команду **Добавить Подэлемент → <Имя Подэлемента>**. Выбранный подэлемент будет добавлен в редактируемую конфигурацию.

Другой способ добавления подмодуля (подэлемента) в текущую конфигурацию:

1. Выделить требуемый элемент (модуль) конфигурации.
2. Выбрать команду **Вставка → Добавить Подэлемент → <Имя Подэлемента>** в главном меню. Выбранный подэлемент будет добавлен в редактируемую конфигурацию.

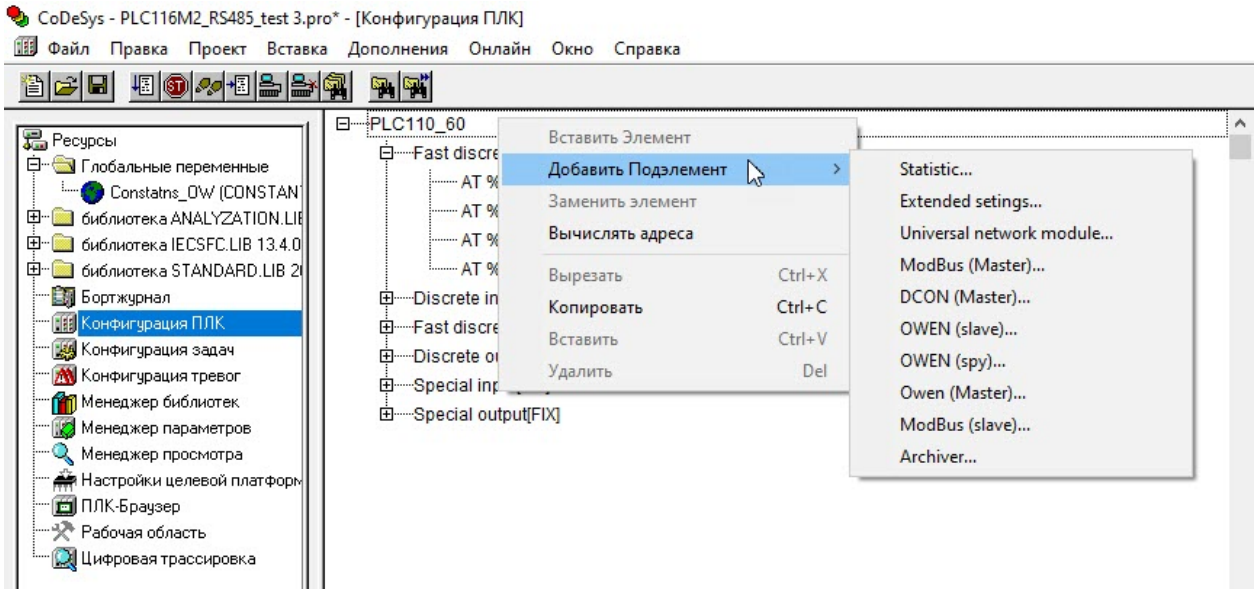


Рисунок 10.5 – Добавление подмодуля в контекстном меню

10.2.3 Замена модуля (элемента)

Для замены модуля (элемента) в текущей конфигурации следует:

1. Выделить требуемый модуль (элемент) конфигурации.
2. В контекстном меню модуля выбрать команду **Заменить Элемент → <Имя элемента>**. Выбранный элемент заместит собою выделенный модуль (элемент) редактируемой конфигурации.

Другой способ замены модуля (элемента) в текущей конфигурации:

1. Выделить требуемый модуль (элемент) конфигурации.
2. Выбрать команду **«Дополнения → Заменить элемент → <Имя элемента>** в главном меню. Выбранный элемент заместит собою выделенный модуль (элемент) редактируемой конфигурации.

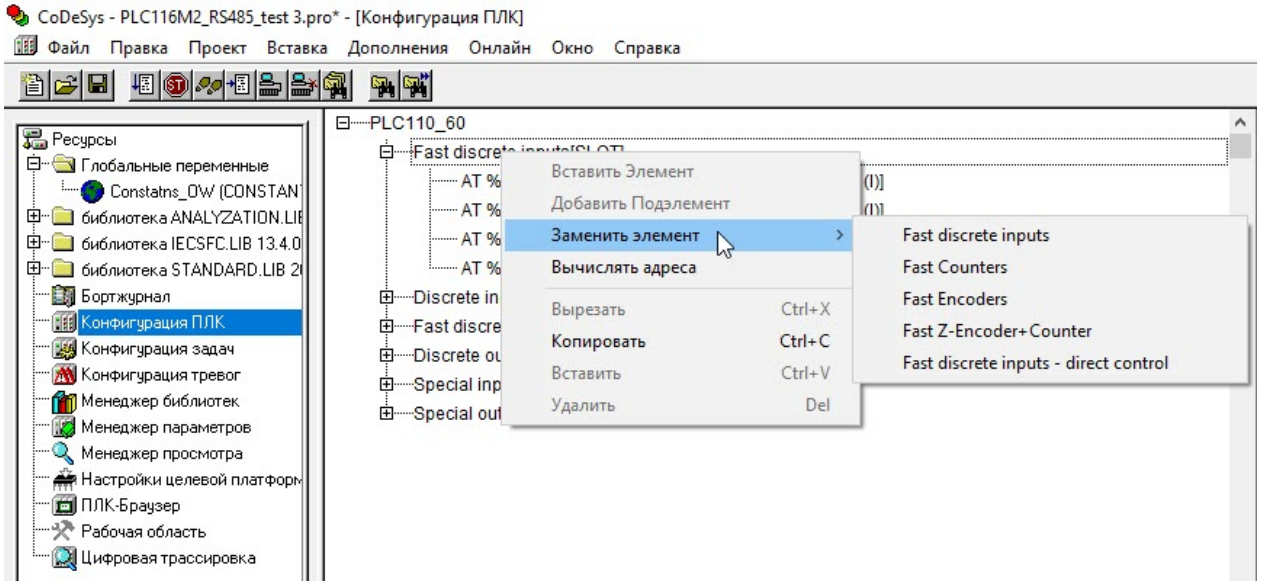


Рисунок 10.6 – Замена элемента в контекстном меню

10.2.4 Удаление модуля (элемента)

Для удаления модуля (элемента) из текущей конфигурации (можно удалить только добавляемые модули, фиксированные модули не могут быть удалены из конфигурации) следует:

1. Выделить требуемый модуль в дереве конфигурации.

2. Выбрать команду **Удалить** в контекстном меню модуля.
3. 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.

10.3 Размер файла конфигурации

Для определения размера файла конфигурации ПЛК следует:

1. Отключить ПЛК от ПК.
2. Создать копию проекта.
3. Удалить из копии проекта все кроме файла конфигурации.
4. Выбрать команду **Онлайн** → **Создание загрузочного проекта**. В папке с проектом появится файл *<имя_проекта>.PRG*. Размер файла примерно соответствует размеру файла конфигурации.

Верхний предел размера файла конфигурации – **150 Кб**.

Если отведенных 150 Кб не хватает для проекта, то рекомендуется использовать для опроса входов и выходов библиотеки программных компонентов *ModBus.lib* и OwenModbusSlave.lib, см. раздел 7.4.

ПРИМЕЧАНИЕ
В среднем ПЛК корректно обрабатывает до 800 каналов ввода-вывода, настроенных через конфигурацию ПЛК, но возможны значительные отклонения в большую и меньшую сторону в зависимости от перегруженности и оптимизации проекта.

ПРИМЕЧАНИЕ
В случае увеличения числа каналов ввода-вывода может потребоваться увеличение времени цикла ПЛК.

10.4 Параметры модулей

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

10.4.1 Вкладка «Базовые параметры»

В полях вкладки «Базовые параметры» отображаются значения параметров:

- **Идент. модуля** – идентификационный номер модуля;
- **Идент. узла** – положение модуля на его уровне иерархии в общей конфигурации. Значение можно редактировать, в таком случае аналогичные идентификаторы других модулей одного уровня иерархии будут сдвигаться;
- **Адрес входов**, **Адрес выходов**, **Адрес диагностики** – адреса областей ввода-вывода (приводятся конкретные номера). Адреса могут использоваться для обращения при программировании, значения недоступны для редактирования;
- **Комментарий** – произвольный текст комментария.

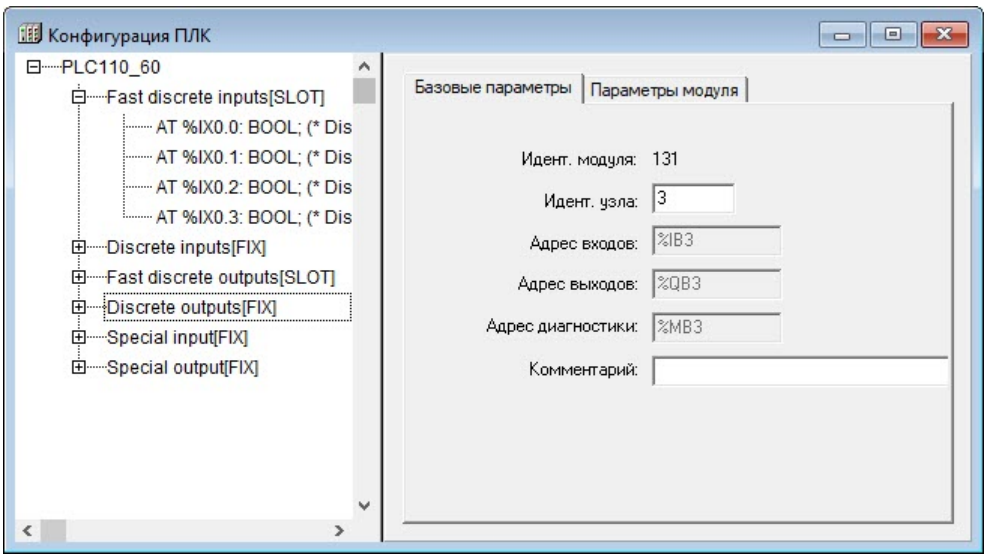


Рисунок 10.7 – Вкладка «Базовые параметры» окна редактора «Конфигурация ПЛК» для модуля дискретных выходов

10.4.2 Вкладка «Параметры модуля»

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

- **Индекс**;
- **Имя**;
- **Значение** (текущее);
- **По умолчанию** – значение по умолчанию;
- **Мин.** – минимальная величина диапазона возможных значений;
- **Макс.** – максимальная величина диапазона возможных значений.

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

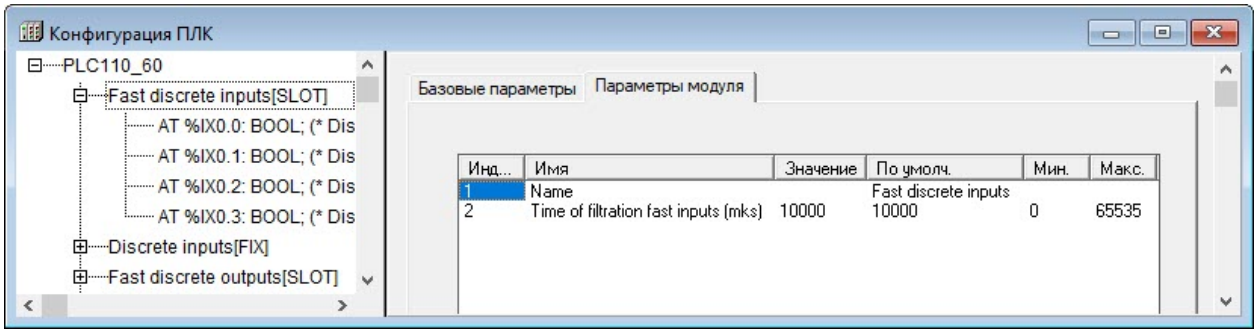


Рисунок 10.8 – Вкладка «Параметры модуля» окна редактора «Конфигурация ПЛК», модуль дискретных входов

Для редактирования цифровых или символьных значений параметров следует щелкнуть ЛКМ 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 ЛКМ кнопку с треугольной стрелкой, отображаемую рядом со значением параметра. Нажатие этой кнопки раскрывает список допустимых значений параметра, в котором следует выбрать требуемое значение. Значение будет подставлено в перечень параметров. После щелчка ЛКМ в любой другой области окна выбранное значение сохраняется в списке.

10.4.3 Каналы

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

Каждый канал – это транслятор данных от внешнего оборудования в область памяти ввода-вывода ПЛК и, если требуется, их преобразователь в цифровой вид. Через канал передается значение входов и 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).

Каналу и соответствующей ячейке памяти может быть присвоено имя. Правила наименования переменных:

- имя может состоять из латинских букв, цифр и знака «_» (нижнее подчеркивание);
- имя должно начинаться с буквы или знака «_»;
- имя должно быть уникальным;
- в некоторых случаях редактирование имен каналов может быть запрещено.

По присвоенному имени к переменной можно обращаться из программы. Переменную канала можно вызвать из программы по адресу, который установлен аппаратно (например, **%IX 0.0.1**).

Способы получения данных из канала:

- после клика по ключевому слову AT появляется поле ввода (см. рисунок 10.9) и в программе можно будет обращаться к данным по указанному имени канала (рекомендованный способ).

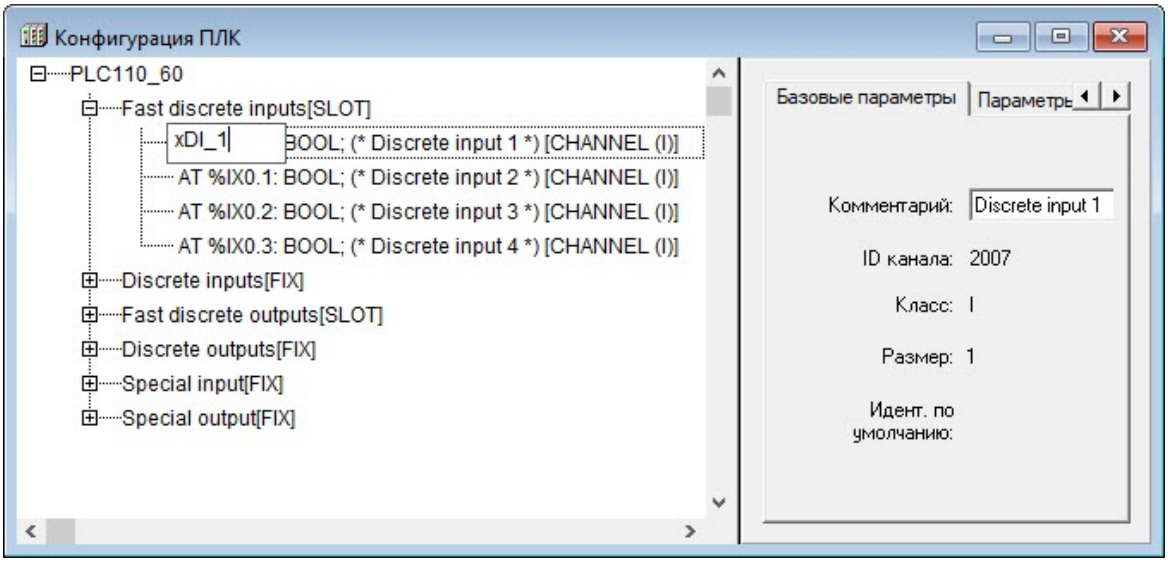


Рисунок 10.9 – Ввод и редактирование имени переменной канала

- объявление необходимого количества глобальных переменных (в окне «Глобальные переменные» и связь добавленных переменных с областью ввода-вывода через МЭК адрес, указывающий в какой области памяти ввода-вывода хранится значение переменной;

Пример

Дискретный вход 1: **DI1 AT %IX0.0.1: bool**,

где **DI1** – имя переменной, задаваемое пользователем;

AT – указатель, что свое значение переменная будет получать из памяти ввода-вывода;

%IX0.0.1 – указание на ячейку хранения значения в области памяти ввода-вывода.

- обращение к значениям переменных в программе непосредственно через МЭК-адрес переменной в области ввода-вывода.

ВНИМАНИЕ

Не рекомендуется использовать прямую адресацию с помощью ключевого слова **AT**. В случае использования прямой адресации компилятор не проверяет за пользователем область памяти, на которую он ссылается во время объявления переменной. Переменным, размещаемым в области конфигурации ПЛК, следует присваивать имена непосредственно в области конфигурации. Дополнительное объявление переменных, объявленных в области конфигурации, не требуется.

Данные 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** – указывается в битах.

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

10.5 Задание времени цикла

Для изменения параметров времени цикла ПЛК следует:

1. В дереве окна «Конфигурация ПЛК» выделить корневой элемент (например, «PLC110-60», см. рисунок 10.10).
2. В области задания параметров перейти на вкладку «Параметры модуля».

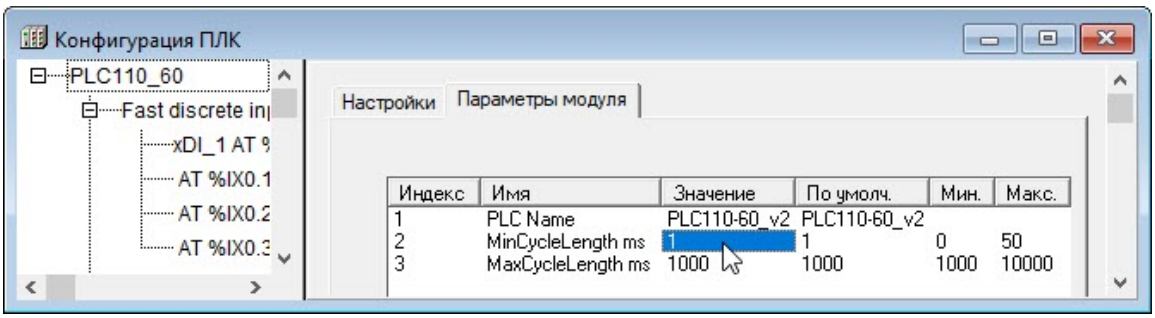


Рисунок 10.10 – Задание минимального значения цикла работы ПЛК

3. Задать требуемые значения параметров времени цикла ПЛК:
   - **PLCName** – имя контроллера, используемое в текущем проекте. Максимальная длина имени – 80 символов. Имя может содержать любые русские и латинские буквы и знаки и предназначено для хранения описаний элементов конфигурации ПЛК и их назначения непосредственно в ПЛК. Имя контроллера может быть использовано для идентификации контроллера в сети или для вывода на ЖК-панель.
   - **MinCycleLength, ms (Минимальное значение цикла работы ПЛК, в мс)** – параметр определяет минимальный период, с которым ПЛК выполняет полный цикл своей работы. Диапазон значений от 0 до 50 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 учитывать, 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. 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. В таком случае 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.

     Узнать о времени выполнения пользовательской программы, сервисных функций и о времени простоя процессора можно в модуле «Statistic» (см. раздел 10.7.8).

     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 вызывается следующий цикл ПЛК и не гарантируется строгое выполнение цикла через равные промежутки времени, т. к. длительность выполнения сервисных функций может изменяться от цикла к циклу.

     ПРИМЕЧАНИЕ
     Не рекомендуется устанавливать время цикла равное нулю.
   - **MaxCycleLength, ms (Максимальное значение цикла работы ПЛК, в мс)** – параметр определяет максимальное допустимое время, за которое ПЛК выполняет полный цикл своей работы. Если в процессе работы ПЛК заданная величина будет превышена (зависание программы или выполнение бесконечного цикла), то ПЛК будет принудительно перезагружен. То есть параметр **MaxCycleLength** задает время ожидания **сторожевого таймера** («WatchDog Timer»). Диапазон значений от 1000 до 10000 мс, значение по умолчанию – 1000 мс.

10.6 Фиксированные модули (элементы) конфигурации. Входы и выходы

10.6.1 Fast discrete inputs (Быстрые дискретные входы)

Модуль быстрых (высокочастотных) дискретных входов (Fast discrete input) отображает в области ввода-вывода значения, характеризующие состояния дискретных быстрых (высокочастотных) входов ПЛК.

Модуль имеет битовые каналы по числу быстрых входов.

Параметры модуля:

- **Name** – имя элемента конфигурации ПЛК, используемое в текущем проекте. Максимальная длина имени – 80 символов, имя может содержать любые кириллические и латинские символы и предназначено для хранения описаний элементов конфигурации ПЛК и их назначения непосредственно в ПЛК. Значение имени может быть использовано, например, для вывода на ЖК-экран или панель;
- **Time of filtration fast inputs, mks (Время фильтрации быстрых входов, мкс)** – период опроса состояния одного дискретного входа; задается в микросекундах (1 ед. = 1 мкс). Диапазон значений от 0 до 65535, значение по умолчанию – 10000 (10 мс).

Принцип действия фильтрации:

– в сдвиговом регистре в драйвере каждого дискретного входа накапливаются значения четырех последних состояний, полученных в результате опроса с периодом, заданным в параметре «Время фильтрации»;

– 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 три и более, то битовый канал переключается на **1 (TRUE)**;

– если количество единиц в сдвиговом регистре равно двум, то состояние битового канала дискретного входа не меняется.

Режим фильтрации можно отключить установкой в параметре «Время фильтрации» значения **0**. Отключение фильтрации требуется для работы с подчиненными модулями энкодеров, чтобы не пропускать высокочастотные сигналы, а также в тех случаях, когда ПЛК функционирует без ограничения цикла по частоте, т. е. на максимальной возможной частоте.

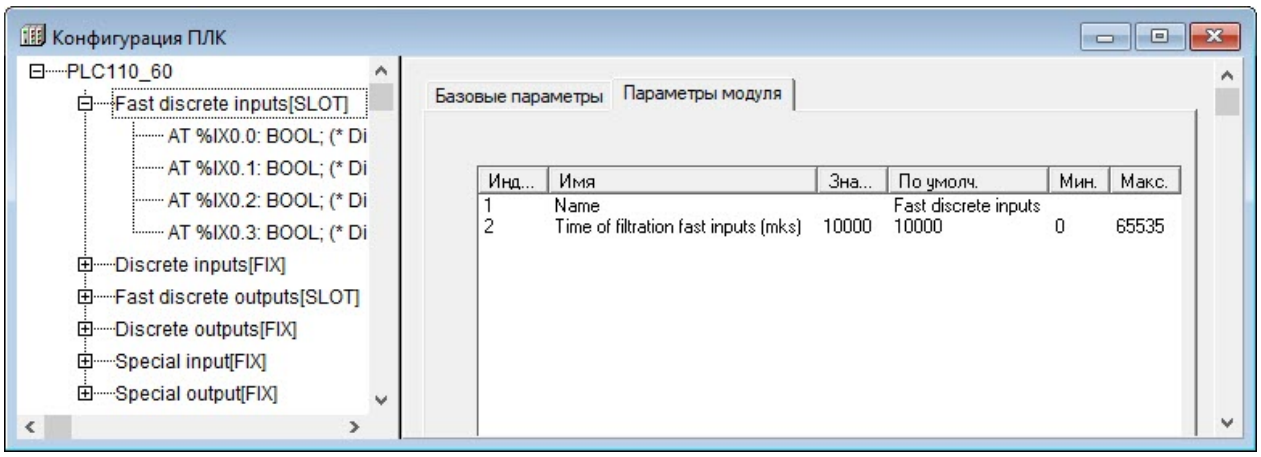


Рисунок 10.11 – Параметры модуля быстрых (высокочастотных) дискретных входов (Fast Discrete input)

10.6.1.1 Замещающие элементы (модули)

Процедура замещения модуля описана в разделе 10.2.3.

Модули для замены модуля «Быстрые дискретные входы» (Fast discrete input):

- **Fast Counters** (Высокочастотный Счетчик) (см. раздел 10.6.1.2);
- **Fast Encoders** (Высокочастотный энкодер) (см. раздел 10.6.1.3);
- **Fast Z-Encoder+Counter** (Высокочастотный Z-энкодер) только для ПЛК110-60 и ПЛК160 (см. раздел 10.6.1.4);
- **Fast discrete inputs - direct control** (Прямое управление дискретными быстрыми входами) только для ПЛК110-60 (см. раздел 10.6.1.5).

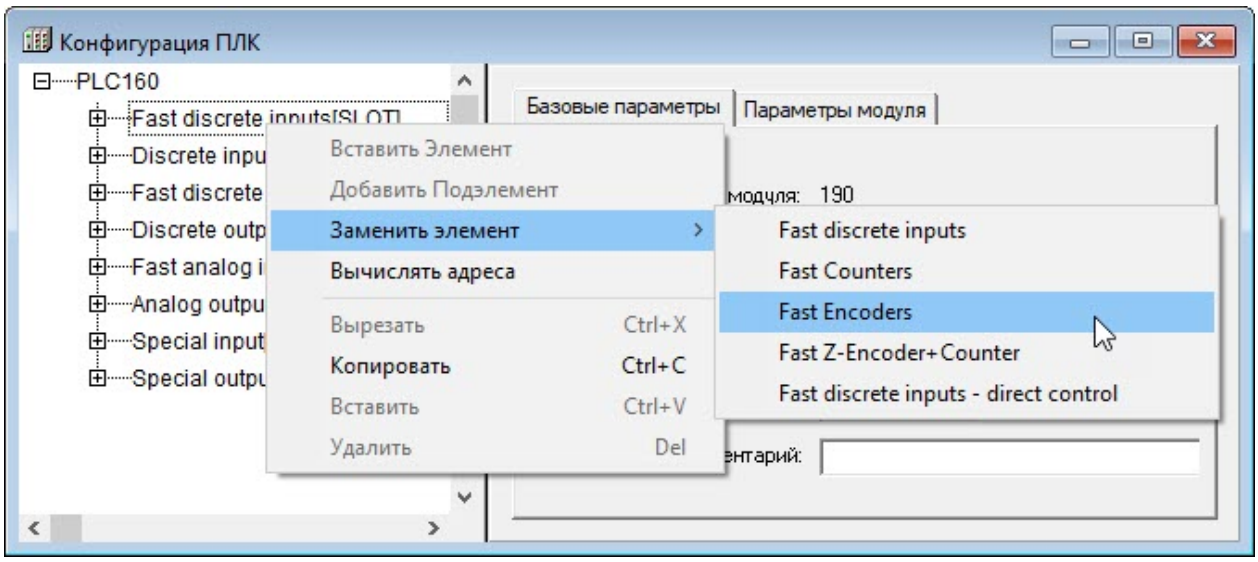


Рисунок 10.12 – Модули для замены модуля «Быстрые дискретные входы» (Fast discrete input)

10.6.1.2 Fast Counters (Высокочастотный счетчик)

Высокочастотный счетчик – программный модуль, ведущий учет входных импульсов, поступающих на быстрые дискретные входы ПЛК, и экспортирующий учетные данные в программу ПЛК, в соответствующее место в области памяти входов-выходов.

Модуль **«Высокочастотный счетчик» (Fast Counter)** является модулем, замещающим модуль быстрых дискретных входов.

Значение быстрого счетчика каждый программный цикл увеличивается на количество импульсов на входе, которое он зарегистрировал в течение цикла. Счетчик обнуляется в случае достижения значения большего, чем 65535 ($FFFF_{16}$), либо, если происходит не увеличение, а уменьшение значения счетчика, после того, как будет достигнут ноль, значение счетчика станет равно 65535 ($FFFF_{16}$). Импульсы считаются по переднему (возрастающему) фронту импульса. Таким образом, для использования и обработки значения данного канала, необходимо считывать его каждый раз в начале цикла пользовательской программы. Например, можно ввести в программу дополнительную переменную, и в начале цикла передавать в эту переменную значение переменной, привязанной к счетчику.

Модуль имеет два (для ПЛК110-30 и ПЛК110-32) или четыре (для ПЛК110-60 и ПЛК160) шестнадцатибитовых каналов, по числу быстрых входов.

Модуль не имеет параметров.

10.6.1.3 Fast Encoder (Высокочастотный энкодер)

Высокочастотный энкодер – программный модуль, позволяющий подключать к двум быстрым дискретным входам **относительного энкодера** для получения данных о вращении или линейном перемещении контролируемого механизма с последующей передачей информации в цифровой форме в программу ПЛК. Для работы с механическими энкодерами следует включать режим фильтрации дребезга сигналов в параметре **Time of filtration in mks**. Схемы подключения энкодеров к быстрым входам ПЛК приводятся в руководствах по эксплуатации.

Модуль является замещающим для модуля Быстрых дискретных входов (Fast discrete input).

Модуль имеет один или два шестнадцатибитовых канала (формат WORD) по максимальному числу подключаемых к контроллеру энкодеров.

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Time of filtration in mks (Время фильтрации, в микросекундах)** – время установления сигнала, значения от 0 до 255 (1 единица = 1 мкс). Параметр задает время, в течение которого контроллер игнорирует дребезг контактов механического энкодера. Отсчет времени начинается с момента переключения выхода энкодера в противоположное положение. Если по истечении заданного времени сигнал выхода энкодера не изменяется, то считается, что он установился и дребезг контакта закончился.

Значение параметра задается в сотнях микросекунд (т. е. 1 единица равна 100 мкс, 10 ед. = 1 мс).

Для механических энкодеров рекомендуется устанавливать значение параметра в диапазоне от 20 до 40 (от 2 до 4 мс). Для отключения фильтрации (при применении оптических энкодеров) следует задать значение 0. Если в модуле только один параметр, то его действие распространяется на все энкодеры, подключенные к контроллеру.

- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

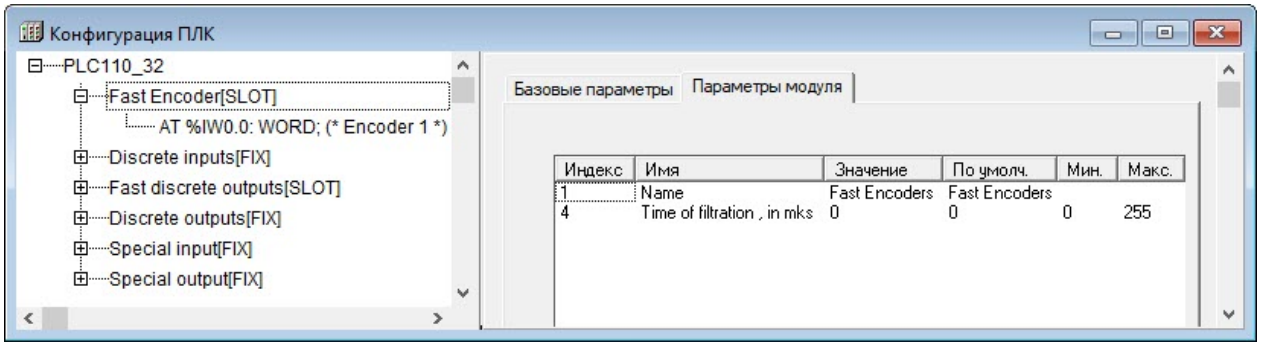


Рисунок 10.13 – Параметры модуля Высокочастотный энкодер (FastEncoder) для ПЛК110-32

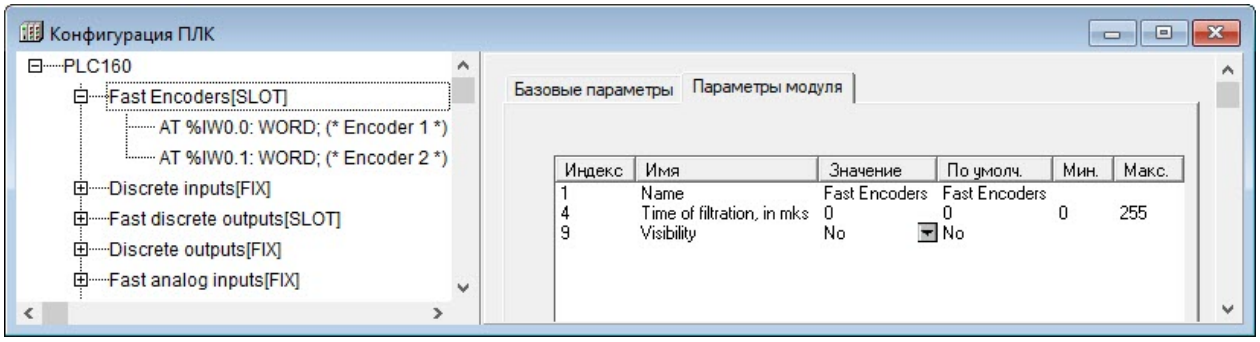


Рисунок 10.14 – Параметры модуля Высокочастотный энкодер (FastEncoder) для ПЛК160

10.6.1.4 Fast Z-Encoder+Counter (Высокочастотный Z-энкодер + счетчик)

Высокочастотный Z-энкодер – программный модуль, который позволяет подключать на трех быстрых дискретных входах контроллера **относительный энкодер с отметкой пересечения нуля** для получения данных о вращении или линейном перемещении контролируемого механизма, с последующей передачей информации в цифровой форме в программу ПЛК. Для работы с механическими энкодерами следует включать режим фильтрации дребезга сигналов в параметре **Time of filtration in mks**.

Модуль является замещающим для модуля **Быстрых дискретных входов (Fast discrete input)**. **Модуль доступен только в ПЛК110-60 и ПЛК160.**

Неиспользуемый при подключении Z-энкодера дискретный вход может быть использован в качестве высокоскоростного счетчика.

Модуль имеет два шестнадцатибитовых канала (формат WORD), по числу подключаемых к контроллеру Z-энкодеров и счетчиков.

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Time of filtration in mks (Время фильтрации, в микросекундах)** – время установления сигнала, значения от 0 до 255, 1 единица = 1 мкс (см. раздел 10.6.1.3);
- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

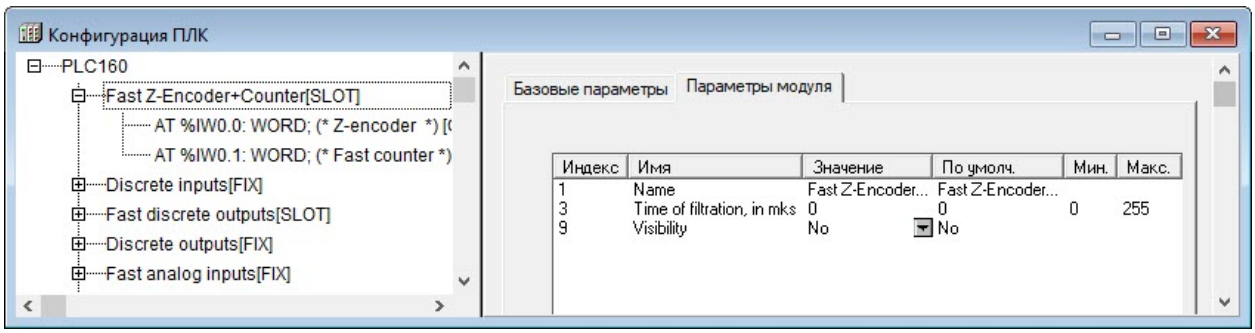


Рисунок 10.15 – Параметры модуля «Высокочастотный Z-энкодер + счетчик»

Особенностью подключения Z-энкодера является порядок подключения выходов энкодера. Контроллер требует следующего порядка подключения:

- на первый быстрый вход (DI1) – сигнал A;
- на второй быстрый вход (DI2) – сигнал B;
- на третий быстрый вход (DI3) – сигнал Z;
- четвертый быстрый вход (DI4) – может использоваться в качестве высокоскоростного счетчика.

10.6.1.5 Fast discrete inputs - direct control (Прямое управление быстрыми дискретными входами)

Модуль переводит быстрые дискретные входы в режим прямого управления из функций стандартной библиотеки SysLibPort. Дискретные входы не отображаются в пространстве области ввода (%I).

Прямое управление быстрыми входами из библиотеки SysLibPort служит для управления быстрыми входами из процедуры обработки прерываний высокочастотного таймера, т. к. вызов обработчика по таймеру происходит чаще, чем заканчивается цикл ПЛК и, соответственно, чаще, чем происходит обращение к памяти вывода.

Модуль является замещающим для модуля **Быстрых дискретных входов (Fast discrete input)**.

Во время установки модуля все быстрые входы переключаются на режим прямого управления из библиотеки SysLibPort и не реагируют на изменение значений каналов в памяти ввода.

Модуль доступен в ПЛК110 и ПЛК160 и не имеет параметров и каналов.

Работа с высокочастотным таймером подробно описана в разделе 14.

10.6.2 Discrete inputs (Дискретные входы)

Фиксированный модуль «Discrete inputs» (Дискретные входы) отображает в области ввода-вывода значения, характеризующие состояния дискретных входов ПЛК.

Модуль имеет два (для ПЛК110-30, ПЛК110-32 и ПЛК160) или четыре (для ПЛК110-60) восьмибитовых канала.

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Time of filtration general inputs in ms (Время фильтрации основных входов в миллисекундах)** – период опроса состояния одного дискретного входа, задается в сотнях микросекунд (1 ед. = 1 мс). Диапазон значений от 0 до 1000, значение по умолчанию – 10.

  Принцип действия фильтрации описан в разделе 10.6.1. Режим фильтрации может быть отключен установкой в параметре «Время фильтрации» значения «0». Фильтрацию следует отключать во время работы с подчиненными модулями энкодеров, чтобы не пропускать высокочастотные сигналы, а также в тех случаях, когда ПЛК функционирует без ограничения цикла по частоте, т. е. на максимальной возможной частоте.

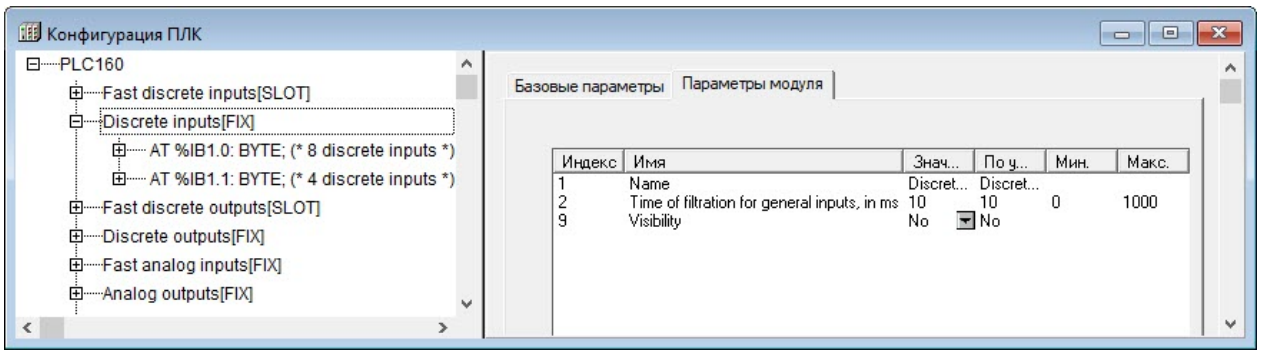


Рисунок 10.16 – Параметры модуля «Дискретные входы» (Discrete input)

10.6.3 Fast discrete outputs (Быстрые дискретные выходы)

Фиксированный модуль быстрых (высокочастотных) дискретных выходов (Fast discrete output) отображает в области памяти ввода-вывода значения быстрых дискретных выходов ПЛК.

Модуль имеет несколько битовых каналов по числу быстрых выходов контроллера.

ПРИМЕЧАНИЕ
Следует обратить внимание, что в ПЛК160 при частых коммутациях ресурс электромеханических реле может быстро исчерпаться. Частота срабатывания выходов ограничена возможностями электромеханических реле.

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Safe Value (Безопасное значение)** – **TRUE** или **FALSE** для быстрого дискретного выхода.

  Во время загрузки или в случае сбоя в работе ПЛК, выходы могут оказаться выключены или включены. Такая неопределенность может быть недопустима во время эксплуатации управляемого оборудования, и для исключения подобной ситуации ПЛК переводит выходы во время сбоя или загрузки в состояние, заданное в параметре **«Безопасное состояние выхода» (Safe Value).**Значение параметра **FALSE** означает, что выход выключен (0), **TRUE** – выход включен (1). Значение параметра устанавливается раздельно для каждого битового канала.

- **Visibility (Видимость)** – задает видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

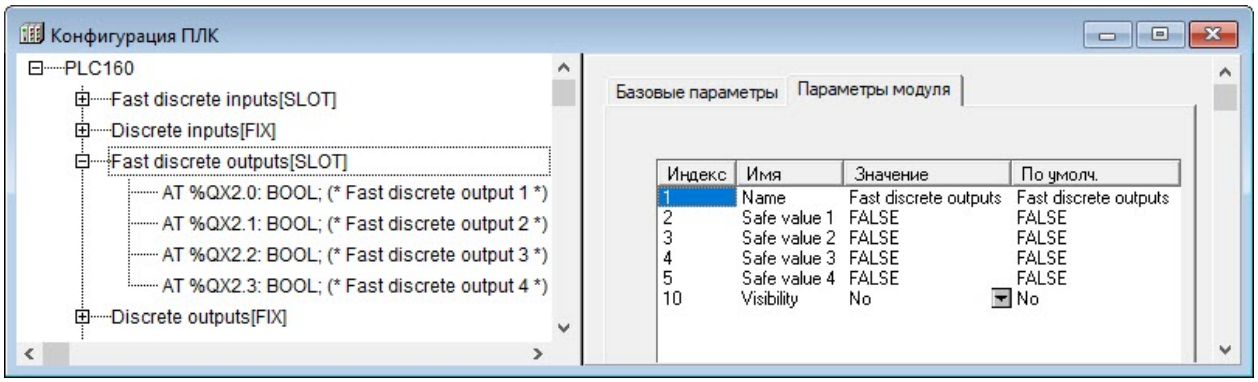


Рисунок 10.17 – Параметры модуля Быстрые дискретные выходы (Fast discrete output)

10.6.3.1 Замещающие элементы (модули)

Процедура замещения модуля описана в разделе 10.2.3.

Модули для замены модуля «Быстрые дискретные выходы (Fast Discrete output)»:

- **PWM (Pulse-wide modulator)** – ШИМ, см. раздел 10.6.3.2;
- **Fast discrete outputs - Direct control** – прямое управление быстрыми дискретными выходами, см. раздел 10.6.3.3.

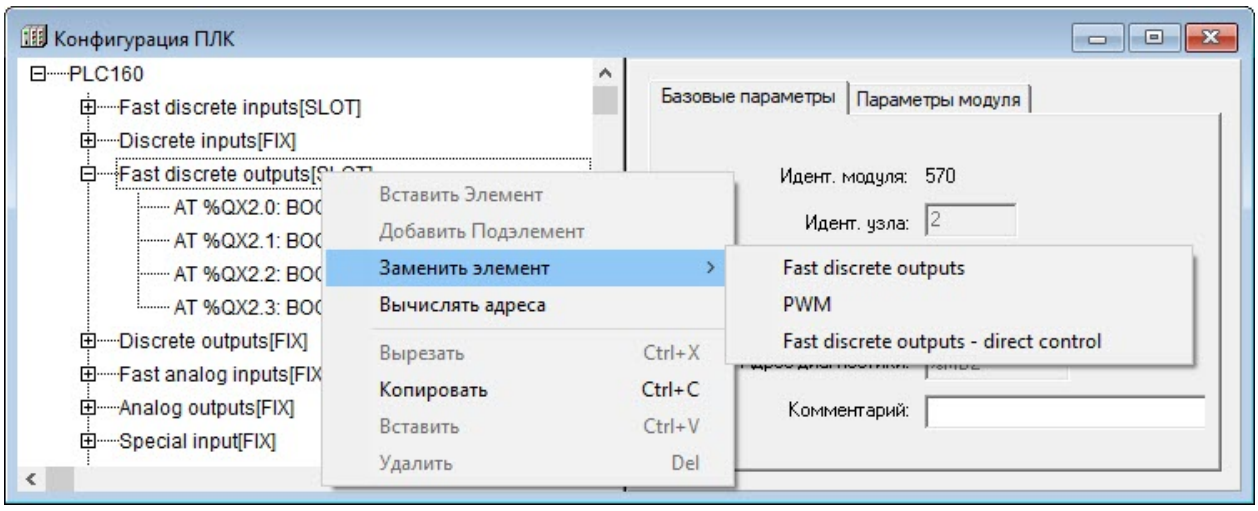


Рисунок 10.18 – Замещающие модули модуля Быстрые дискретные выходы (Fast Discrete output)

10.6.3.2 PWM (Pulse-wide modulator) – ШИМ

Модуль ШИМ (PWM – Pulse-wide modulator) предназначен для работы с генератором ШИМ, который подключен к дискретному выходу. Модуль является замещающим для модуля быстрых дискретных выходов.

ПРИМЕЧАНИЕ
Следует обратить внимание, что в ПЛК160 при частых коммутациях ресурс электромеханических реле может быстро исчерпаться. Частота срабатывания выходов ограничена возможностями электромеханических реле.

Каналы модуля:

- **PWM power** – шестнадцатибитовый канал (формат WORD), задающий значение скважности ШИМ. Изменяется от 0 (0 %) до 1000 (100 %);
- **PWM Period** – тридцатидвухбитный канал (формат DWORD), позволяющий задать или прочитать значение периода ШИМ, заданного в мкс.

  В начале работы значение периода ШИМ записывается из одноименного параметра в канал, затем оно может быть изменено в канале. Измененное значение канала PWM Period не передается в одноименный параметр модуля, поэтому в случае выключения контроллера значение не сохраняется.

Параметры модуля:

- **Min. duration of PWM in mksec (минимальная длительность импульса ШИМ в мкс)** – ограничение на минимальную длительность импульса ШИМ. Диапазон значений от 1 до 65000, значение по умолчанию – 3000 мкс;
- **PWM default Period in mks (период ШИМ в мкс)** – длительность одного периода ШИМ-регулирования. Принимает значения от 2 до 4 294 000 000 мкс, соответственно задавая период ШИМ до 4294 секунд. При наличии одноименного канала в модуле значение параметра переписывается в канал при загрузке пользовательской программы. В дальнейшем значение периода ШИМ может быть изменено в канале модуля;
- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

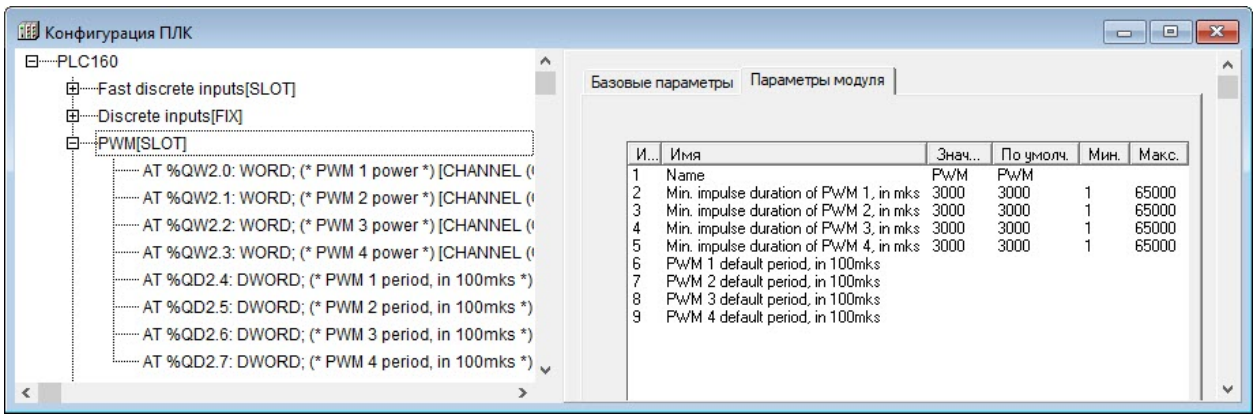


Рисунок 10.19 – Параметры модуля ШИМ (PWM – Pulse-wide modulator)

10.6.3.3 Fast discrete outputs – Direct control (Прямое управление быстрыми дискретными выходами)

Модуль переводит быстрые дискретные выходы в режим прямого управления из функций стандартной библиотеки SysLibPorts.lib. Выходы не отображаются в пространстве области вывода (%Q).

ПРИМЕЧАНИЕ
Следует обратить внимание, что в ПЛК160 при частых коммутациях ресурс электромеханических реле может быстро исчерпаться. Частота срабатывания выходов ограничена возможностями электромеханических реле.

Прямое управление быстрыми выходами из библиотеки SysLibPorts.lib требуется для управления быстрыми выходами из процедуры обработки прерываний высокочастотного таймера, т. к. вызов обработчика по таймеру происходит чаще, чем заканчивается цикл ПЛК и, соответственно, чаще, чем происходит обращение к памяти вывода.

В случае установки модуля все быстрые выхода переключаются на режим прямого управления из библиотеки SysLibPorts.lib и не реагируют на изменение значений каналов в памяти вывода.

Модуль не имеет параметров и каналов.

10.6.4 Discrete outputs (Дискретные выходы)

Модуль дискретных выходов (Discrete outputs) отображает в области памяти ввода-вывода значения дискретного выхода ПЛК.

Модуль имеет несколько битовых каналов (количество каналов зависит от варианта исполнения контроллера).

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Safe Value (Безопасное значение)** – **TRUE** или **FALSE** для быстрого дискретного выхода.

  Во время загрузки или в случае сбоя в работе ПЛК его выходы могут оказаться выключены или включены. Такая неопределенность может быть недопустима во время эксплуатации управляемого оборудования, и для исключения подобной ситуации ПЛК переводит выходы во время сбоя или загрузки в состояние, заданное в параметре **«Безопасное состояние выхода» (Safe Value).** Значение параметра **FALSE** означает, что выход выключен (0), **TRUE** – выход включен (1). Значение параметра устанавливается раздельно для каждого битового канала.

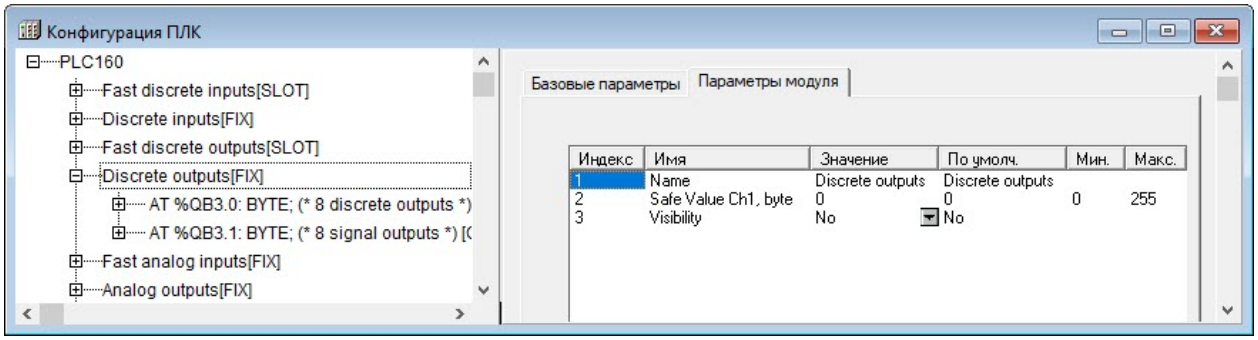


Рисунок 10.20 – Параметры модуля Дискретные выходы (Discrete outputs)

10.6.5 Fast analog inputs (Аналоговые входы)

Структура модуля

Фиксированный модуль «Analog input» (Аналоговые входы) отображает в области ввода-вывода значения, характеризующие состояния аналоговых входов ПЛК.

Модуль имеет восемь идентичных каналов (%IR4.0–%IR4.7), измеряющих напряжение в диапазоне от 0 до 10 В или ток в диапазонах 4–20 мА, 0–20 мА и 0–5 мА.

Аналоговые входы имеют групповую гальваническую изоляцию на 560 В от остальной части схемы. Период обновления значений каналов %IR4.0–%IR4.7 равен 10 мс. Каналы аналогового ввода во всех режимах работы имеют защиту от перегрузки и выдерживают подачу на вход сигналов с напряжением от –40 до +40 В.

Структура одного из входных каналов приведена на рисунке ниже.

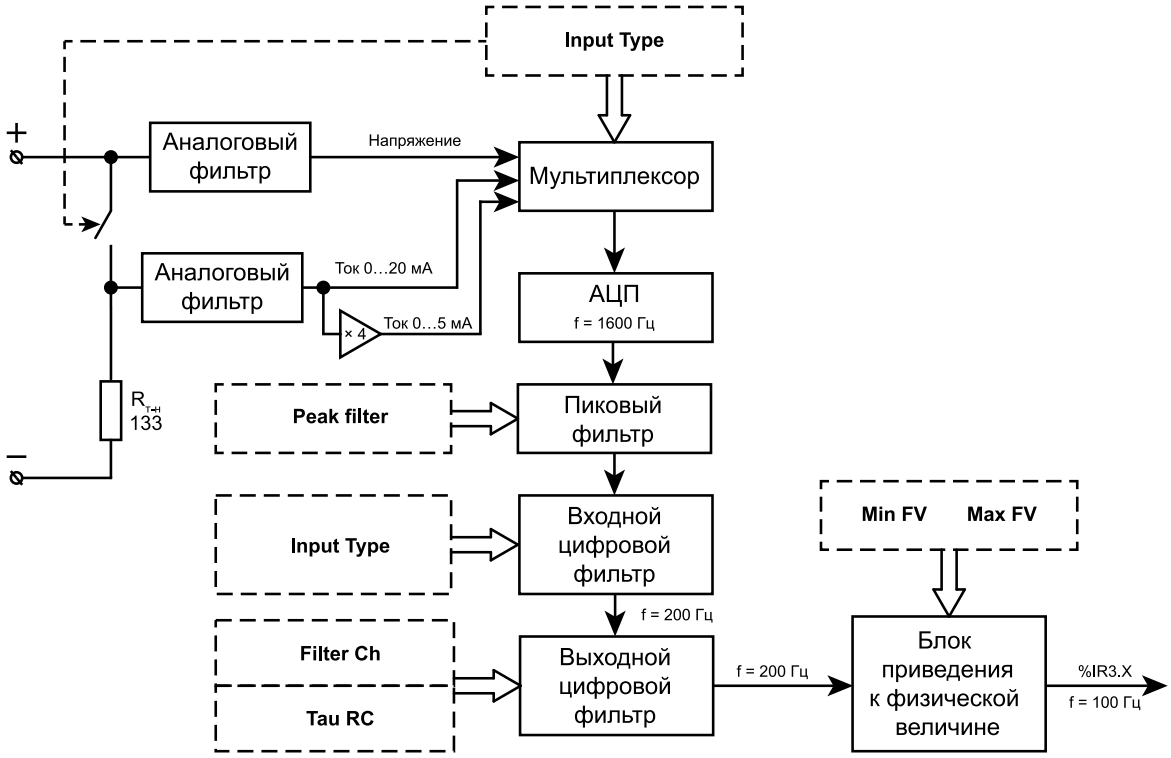


Рисунок 10.21 – Структура каналов модуля аналогового ввода

Параметры модуля:

- **FItr Comm (Тип входного фильтра)** – общий для всех каналов параметр. Входной цифровой фильтр обеспечивает предварительную фильтрацию сигналов на аналоговых входах. Частота отсчетов на входе фильтра равна 1600 Гц, частота выходных отсчетов 200 Гц. Параметр может принимать следующие значения (по умолчанию установлен фильтр «50 Hz 4por»):
  - **Откл** – входной фильтр отключен. При отключенном входном фильтре обеспечивается минимальная задержка результатов измерения, но не гарантируется заявленная точность

измерения. Режим работы с отключенным входным фильтром применять не рекомендуется;

– **50 Hz 1por** – входной фильтр среднего, первого порядка, длиной 32. Групповое время задержки фильтра равно 10 мс;

– **50 Hz 2por** – входной фильтр среднего, второго порядка, длиной 32. Групповое время задержки фильтра равно 20 мс;

– **50 Hz 4por** – входной фильтр среднего, четвертого порядка, длиной 32. Групповое время задержки фильтра равно 40 мс;

– **200 Hz 1por** – входной фильтр среднего, первого порядка, длиной 8. Групповое время задержки фильтра равно 2,5 мс. При работе с этим типом фильтра не гарантируется заявленная точность измерения.

ПРИМЕЧАНИЕ
Фильтры меньшего порядка рекомендуется устанавливать только в случае необходимости получения задержки измерения менее 40 мс.
Фильтры **50 Hz** 1, 2 и 4 порядков обеспечивают подавление помех с частотами, кратными 50 Гц.

- **Input Type (Тип аналогового входа)** – параметр определяет тип измеряемого сигнала: 4–20 мА, 0–20 мА, 0–5 мА, 0–10 В;
- **Peak Filter (Пиковый фильтр)** – пиковый фильтр используется для подавления импульсных помех. Режим работы пикового фильтра устанавливается индивидуально для каждого канала. Параметр может принимать значения от 1 до 200. Работа фильтра описана ниже;
- **Filter Ch (Канальный фильтр)** – канальный фильтр используется для дополнительной, индивидуально для каждого канала, фильтрации измеряемого сигнала. Параметр может принимать следующие значения:

  – **Off** – фильтр отключен;

  – **Avr 4** – фильтр среднего длиной 4, время групповой задержки 10 мс;

  – **Avr 8** – фильтр среднего длиной 8, время групповой задержки 20 мс;

  – **Avr 12** – фильтр среднего длиной 12, время групповой задержки 30 мс;

  – **Avr 16** – фильтр среднего длиной 16, время групповой задержки 40 мс;

  – **RC** – аналог RC фильтра. Постоянная времени RC-фильтра определяется параметром «Tau RC».

- **Tau RC (Постоянная времени RC-фильтра)** – постоянная времени RC-фильтра. Параметр может принимать значения от 10 до 10 000 мс;
- **Min FV** и **Max FV** – параметры приведения результатов измерения к физической величине.

  Параметры обеспечивают приведение результатов измерения к физической величине и могут принимать значения от –1E9 до +1E9. При подаче на вход измерения минимального сигнала выбранного диапазона результат измерения будет равным «Min FV», при подаче на вход измерения максимального сигнала выбранного диапазона результат измерения будет равным «Max FV». При подаче на вход измерения сигналов с уровнями от минимального до максимального результат измерения будет линейно преобразован в диапазон от «Min FV» до «Max FV» (прямо пропорционально при Max FV > Min FV и обратно пропорционально при Min FV > Max FV).

ВНИМАНИЕ
Новые настройки аналоговых входов вступают в силу после загрузки проекта с новыми настройками и перезагрузки ПЛК по питанию.

Пример
На датчике с выходным током от 4 до 20 мА (параметр «Input Type» равен 4–20 мА), который контролирует давление в диапазоне от 0 до 25 атм, в параметре «Min FV» задано значение 00,00, в параметре «Max FV» – значение 25,00. Вывод результатов измерения будет выполняться в атмосферах. При значении измеренного тока, равном 4 мА, результат измерения будет равен 00,00, при значении измеренного тока, равном 20 мА, результат измерения будет равен 25,00.

Работа пикового фильтра

Пиковый фильтр работает в соответствии с формулой:

$$\begin{array}{l} Y = [X + P] \text{ при } [(X_i - Y_{i-1}) < P]; \\ Y = [X - P] \text{ при } [(X_i - Y_{i-1}) < P]; \\ Y = X \text{ при других условиях} \end{array},$$

где *X* – сигнал на входе фильтра;

Y – сигнал на выходе фильтра;

Y_{i-1} – сигнал на выходе фильтра в предыдущий такт (5 мс) измерения;

P – параметр ограничения скорости, рассчитанный по формуле ниже.

$$P = \frac{PeakFilter \times \text{диапазон измерения}}{200}.$$

Если порог ограничения скорости выбран правильно, то ограничитель скорости не оказывает влияния на измеряемый сигнал. Во время поступления на вход сигнала импульсной помехи, амплитуда помехи будет уменьшена в соответствии с установленным параметром ограничения скорости.

Значение параметра ограничения задается в долях изменения входного сигнала относительно измеряемого диапазона за одну секунду.

Пример
Если максимальная скорость изменения тока равна 50 мА/с, измеряемый диапазон равен 20 – 4 = 16 мА, то скорость изменения тока за одну секунду будет равна $\text{Peak Filter} = \frac{50 \text{ мА/с}}{16 \text{ мА}} = 3{,}125 \text{ диапазона/с}.$

В случае установки параметра равным 4, исключается влияние фильтра на полезный сигнал, и в то же время обеспечивается эффективная защита от импульсных помех. Значение параметра, равное 200, отключает фильтр. По умолчанию значение параметра «Peak Filter» устанавливается равным 200.

Поведение модуля в исключительных ситуациях

В случае возникновения исключительных ситуаций, в результате измерения соответствующего канала (старший байт переменной типа REAL) устанавливается специальное значение, соответствующее исключительной ситуации, остальные байты принимают значение 0xFF.

Модуль аналогового ввода распознает следующие исключительные ситуации:

- результаты измерения заведомо не верны – 0xF1;
- результаты измерения не готовы – 0xF6;
- сигнал на входе больше возможного – 0xFA;
- сигнал на входе меньше возможного – 0xFB;
- перегрузка в канале измерения тока – 0xFC;
- обрыв датчика в канале измерения напряжения – 0xFD.

10.6.6 Analog output (Аналоговые выходы)

Фиксированный модуль «Analog output» (Аналоговые выходы) отображает в области ввода-вывода значения, характеризующие состояния аналоговых выходов ПЛК. Модуль имеет четыре идентичных канала (% QR4.0– % QR4.3). В каждый канал может выводится значение с плавающей точкой в диапазоне от 0,0 мА до 20,0 мА для токового модуля и от 0,0 В до 10,0 В для модуля напряжения. В случае записи в канал значения, выходящего за указанные пределы, значение автоматически ограничивается. Значения, записываемые пользовательской программой в каналы аналогового вывода (% QR4.0– % QR4.3) с периодом 50 мс выводятся в ЦАП соответствующего канала и обновляют состояния аналоговых выходов.

Каждый из каналов имеет следующие параметры:

- **Mode (Режим работы выхода)** – тип аналогового выхода. Возможные значения: «ток 4…20 мА» или «напряжение 0…10 В». Если параметр **Type** задан неверно, модуль работать не будет. Выбор параметра должен соответствовать исполнению ПЛК (значение по умолчанию – «ток 4…20 мА»):
  - ПЛК 160-Х.И-М – «ток 4…20 мА»;
  - ПЛК 160-Х.У-М – «напряжение 0…10 В»;

– ПЛК 160-X.A-M – «ток 4…20 мА» или «напряжение 0…10 В».

- **SafeState (Безопасное состояние)** – значение задается в тех же единицах, что и основные параметры, значение по умолчанию – 0. В безопасное состояние аналоговые выходы устанавливаются при подаче питания на ПЛК и при паузе в обмене между верхней и средней платой на время более 100 мс (которая может быть вызвана сбоем основного процессора или отказом канала связи между верхней и средней платой).

ВНИМАНИЕ
Для корректной работы ПЛК160-х.У-М [M02] в конфигурации ПЛК следует указать режим работы всех аналоговых выходов как «напряжение 0…10 В» вне зависимости от их использования.

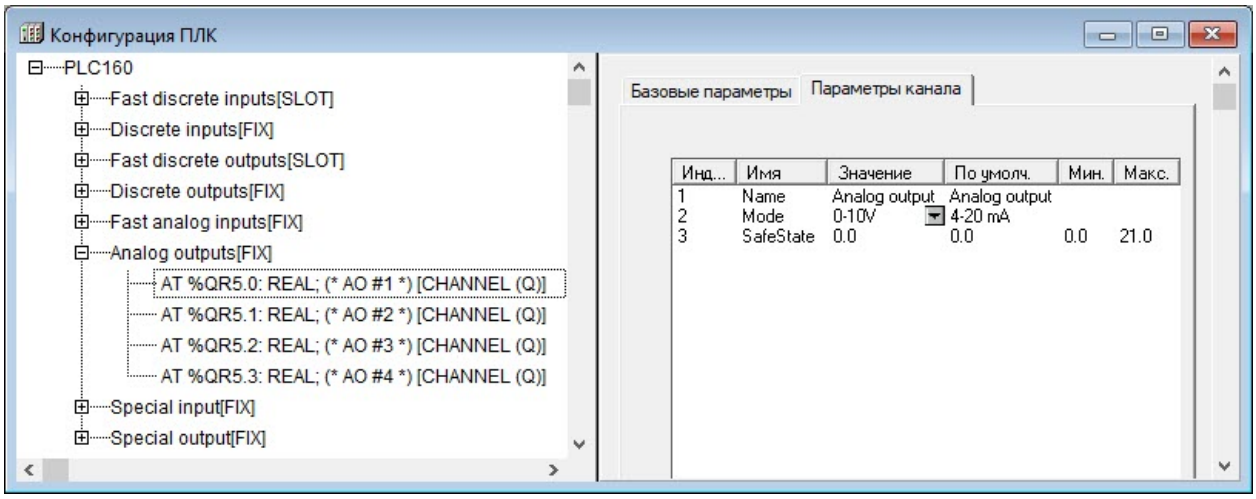


Рисунок 10.22 – Параметры модуля аналогового выхода

10.6.7 Special input (Специальный дискретный вход)

Модуль специального дискретного входа (Special input) – программный модуль, отображающий в область памяти ввода-вывода состояние трехпозиционного переключателя, расположенного на передней панели контроллера. В рабочем режиме переключатель функционирует как дискретный вход: положению «Работа» соответствует состояние **TRUE**, положению «Стоп» – **FALSE**. Состояние «Сброс» никак не отображается.

Модуль имеет один битовый выходной канал и один параметр **«Видимость» (Visibility)**, задающий видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

10.6.8 Special output (Специальный дискретный выход)

Модуль специального дискретного выхода Special output – модуль, содержащий битовую переменную, управляющую устройством подачи звукового сигнала. Если значение переменной – **TRUE**, то подается непрерывный звуковой сигнал.

Модуль имеет битовый канал и один параметр: **«Видимость» (Visibility)**, задающий видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

10.7 Добавляемые модули и подмодули (подэлементы) конфигурации

Процедура добавления подмодулей (подэлементов) в текущую конфигурацию описана в разделе 10.2.2.

В конфигурацию могут быть добавлены подмодули (подэлементы) настройки режимов работы (Master/Slave) и интерфейсов связи:

- ModBus (Master);
- ModBus (Slave);
- DCON (Master);
- OWEN (Slave);

- OWEN (Spy);
- OWEN (Master).

В конфигурацию могут быть добавлены вспомогательные модули:

- Statistic (модуль статистики);
- Universal network module;
- Extended settings;
- Archiver (архиватор).

10.7.1 Модуль ModBus (Master)

Модуль ModBus (Master) используется для работы контроллера в режиме Мастера сети, т. е. для опроса и контроля других Modbus устройств, работающих в сети в подчиненном режиме (slave) – например, модули ввода-вывода, панели оператора, частотные преобразователи и т. д. Во время установки модуля «ModBus (Master)» выбирается коммуникационный интерфейс для обмена данными с другими устройствами, добавляются и настраиваются требуемые переменные.

Модуль имеет один параметр: **«Visibility (Видимость)»**, который задает видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Во время опроса модулем «ModBus (Master)» подчиненных устройств информация о ходе обмена записывается в соответствующих каналах его переменных.

Каналы модуля:

- **Last Address** – адрес последнего опрошенного **ModBus (Slave)** устройства. Модуль запрашивает устройство, и, соответственно, тут же меняется значение: показывается значение адреса последнего запроса.
- **Last Error** – код ошибки. В переменной отображается код ошибки, если информационный обмен прошел неудачно, что требуется для корректности работы опрашиваемого устройства. Коды ошибок данного модуля представлены в приложении Сообщения об ошибках в ПЛК .

Добавление и настройка коммуникационных интерфейсов «ModBus (Master)»

При добавлении модуля «**ModBus (Master)**» в конфигурацию ПЛК в состав модуля уже подключен порт **Debug RS-232**. В случае необходимости работы через другой коммуникационный интерфейс этот порт можно заменить требуемым последовательным портом или модемом (команда **Заменить элемент** → **<Элемент>** контекстного меню строки «Debug RS-232»).

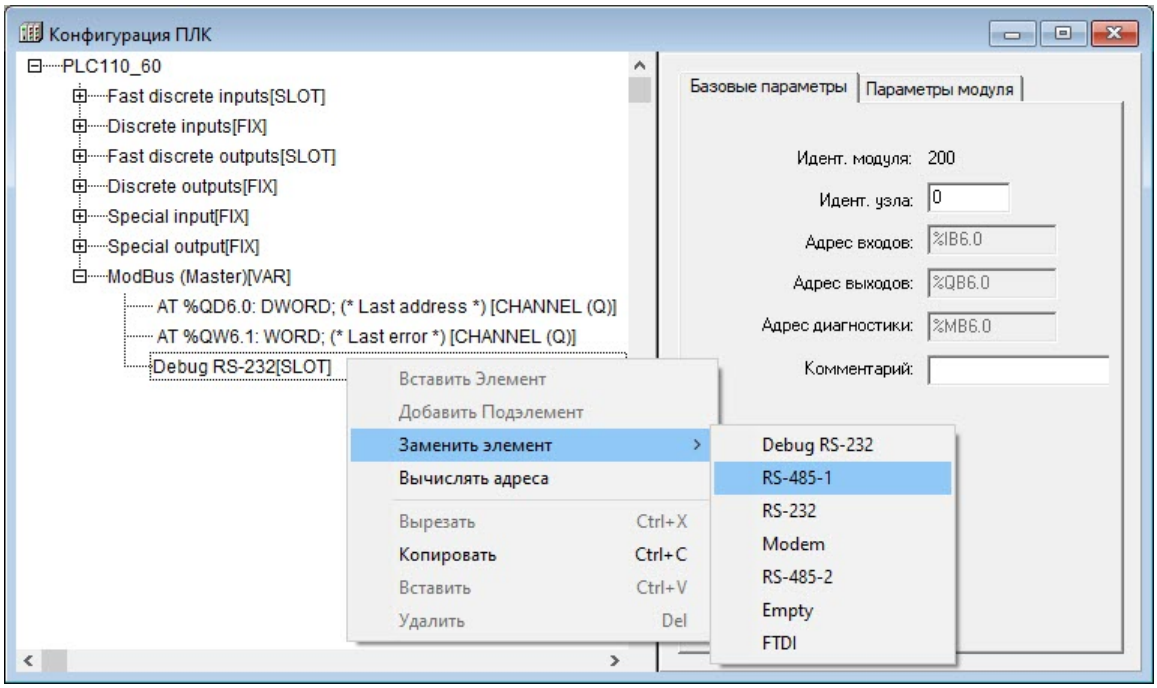


Рисунок 10.23 – Выбор коммуникационных интерфейсов «ModBus (Master)»

ВНИМАНИЕ
Если предполагается, что Мастер сети будет работать с устройствами по протоколу TCP, то необходимая настройка производится в подмодуле устройства (параметр «TCP port» модуля Universal ModBus Device).

10.7.2 Подмодуль Universal Modbus Device

Для добавления в список опроса, проводимого мастером сети, устройств, работающих в режиме ModBus (Slave), следует в модуле ModBus Master добавить подмодуль «Universal Modbus Device» (универсальное устройство Modbus). Для добавления в список нескольких опрашиваемых устройств эту процедуру следует повторить столько раз, сколько устройств должно быть подключено. Для каждого устройства должен быть добавлен соответствующий ему модуль Universal Modbus Device с индивидуальными настройками.

Чтобы добавить подмодуль «Universal Modbus Device» в конфигурацию, следует выбрать команду **Добавить подэлемент → Universal Modbus Device** контекстного меню строки «ModBus (Master)».

Добавленные подмодули «Universal Modbus device» опрашиваются последовательно, в порядке следования в конфигурации (если другой порядок не задан отдельно, в настройках модуля).

Подмодуль «Universal Modbus Device» имеет один канал: **Command (0xff - Start)**:

- если в него записывается значение **0xFF**, то происходит старт работы данного устройства Modbus;
- если модуль уже запущен, то повторная запись в канал значения **0xFF** приводит к внеочередному запросу одной очередной переменной устройства Modbus;
- если в канал записано значение **0xFE**, то происходит его остановка и прекращение всех посылок в сеть;
- если в канал ничего не записано, то устройства опрашиваются автоматически, в порядке очереди.

В случае необходимости устройство можно исключить из списка опроса, подав соответствующую команду в канал **Command (0xff - Start)**. Во время добавления подмодуля «Universal ModBus Device» его параметры и идентификаторы не привязаны к конкретному внешнему устройству (модулю ввода-вывода, операторской панели). Тип внешнего устройства конкретизируется заданием значений параметров подмодуля (конфигурированием модуля).

Параметры подмодуля «Universal ModBus device»:

- **Name** – имя элемента, см. раздел 10.6.1;
- **ModuleIP (IP-адрес)** – IP-адрес ModBus Slave устройства, которым управляет **Мастер**, если обмен будет идти по TCP;

  ПРИМЕЧАНИЕ
  В качестве разделителей октетов IP-адреса используется двоеточие. **Пример: 192:168:1:1**.
- **Max timeout (Максимальный тайм-аут, в мс)** – максимальное время, в течение которого устройство должно ответить на запрос. Если по истечении этого времени **Мастер** не получил ответ на запрос, то это значит, что произошел сбой или авария. Информация о сбое фиксируется в переменной модуля **Last error**. **Мастер** продолжает опрос других устройств. Максимальное значение не ограничено, оно может быть любым, в том числе дробным, но не меньше 10 мс, значение по умолчанию – 150 мс;
- **TCPport (Порт TCP)** – используется только при обмене по протоколу TCP, стандартное значение для протокола **Modbus TCP** – 502 (значение по умолчанию), но в случае необходимости может быть установлено и другое;
- **NetMode (Режим работы в сети)** – значения выбираются из списка («TCP» и «Serial»), значение по умолчанию – «Serial». Значение «**TCP»** – подчиненное устройство работает по протоколу TCP используется интерфейс **Ethernet,** опрашиваемое внешнее устройство идентифицируется по IP-адресу. Значение «**Serial»** – подчиненное устройство обменивается данными через последовательный интерфейс, опрашиваемое внешнее устройство идентифицируется по адресу в сети;
- **ModuleSlaveAddress (Адрес подчиненного устройства)** – диапазон значений от 1 до 247, значение по умолчанию – 1.

  ВНИМАНИЕ
  Значение «0» – специфично и используется для широковещательных сообщений. Например, для работы через шлюз.
- **Work mode (Режим работы)** – режим работы модуля **ModBus (Master)** при опросе внешних устройств может иметь несколько значений (значение по умолчанию – «Polling time»):

– **By Poll time** – «по времени» – контролируемые устройства опрашиваются с периодичностью, заданной в параметре Polling time (Период опроса устройства);

– **By Value change** – «по изменению значения переменных» – модуль **ModBus (Master)** генерирует запрос к соответствующему Slave-устройству в случае изменения значений выходных переменных модуля;

ПРИМЕЧАНИЕ
Выходные переменные – значения, которые модуль ModBus (Master) передает (записывает) в Slave-устройства, входные переменные – параметры, значение которых Мастер запрашивает у Slave-устройств.

– **Both** – «оба варианта» – параметр опрашивается с временным интервалом, заданным в параметре «**Polling time»** и когда изменяются значения выходных переменных (в соответствии с значением «**By Value change**»);

– **By Command** – «по команде» – производится однократная посылка запроса, когда в командный канал «Command» переменной записывается значение **0xFF**.

ВНИМАНИЕ
Для переменных с командным каналом при работе в режиме **By Command** (По команде) управление осуществляется следующим образом: первая посылка значения **0xFF** в командный канал включает функционирование этой переменной, повторная посылка значения **0xFF** инициирует проведение опроса. Аналогично опрос инициируется для переменных с командным каналом при работе в других режимах. При посылке в командный канал значения **0xFE** переменная исключается из цикла опроса мастера.

- **Amount Repeat (Число повторов)** – число повторов чтения/записи переменных при неудачном сеансе связи. В режиме **«По времени» (Polling time)** значение этого параметра не используется. Рекомендуемый диапазон значений от 0 до 5, значение по умолчанию – 0;
- **Byte Sequence (Порядок передачи байтов посылки)** – значения выбираются из списка: «Native» (порядок байтов, используемый в ПЛК) и «Trace_mode» (порядок байтов, используемый в программе Tracemode). Значение по умолчанию – «Trace_mode». Параметр определяет, в каком порядке будут передаваться байты посылки протокола **ModBus** для переменных длиной **32 бита**. У устройств разных производителей этот порядок разный, он не стандартизирован в рамках протокола и поэтому должен быть задан для конкретного устройства. Для работы с модулями ввода-вывода компании «ОВЕН» (например, МВ110-224.8А) следует задать значение параметра **Trace_mode**;
- **Polling time (Период опроса устройства, в мс)** – период опроса внешнего устройства. Диапазон значений от 10 до 10000, значение по умолчанию – 100.

ПРИМЕЧАНИЕ
В Мастере, когда он работает в режиме **«По изменению значения переменных»** или **«По команде»**, нельзя ставить значение параметра **Polling time** слишком маленьким. По умолчанию его значение 100 мс, и в этих режимах оно не влияет на периодичность посылки запросов мастера. Но если на реальном проекте будет замечено, что Мастер при загрузке программы или при **Login** формирует лишние пакеты и/или запросы, которых не должно быть, значение параметра следует увеличить (до 200, 300 и т. д.) до предотвращения появления лишних пакетов.

- **Visibility (Видимость)** – задает видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

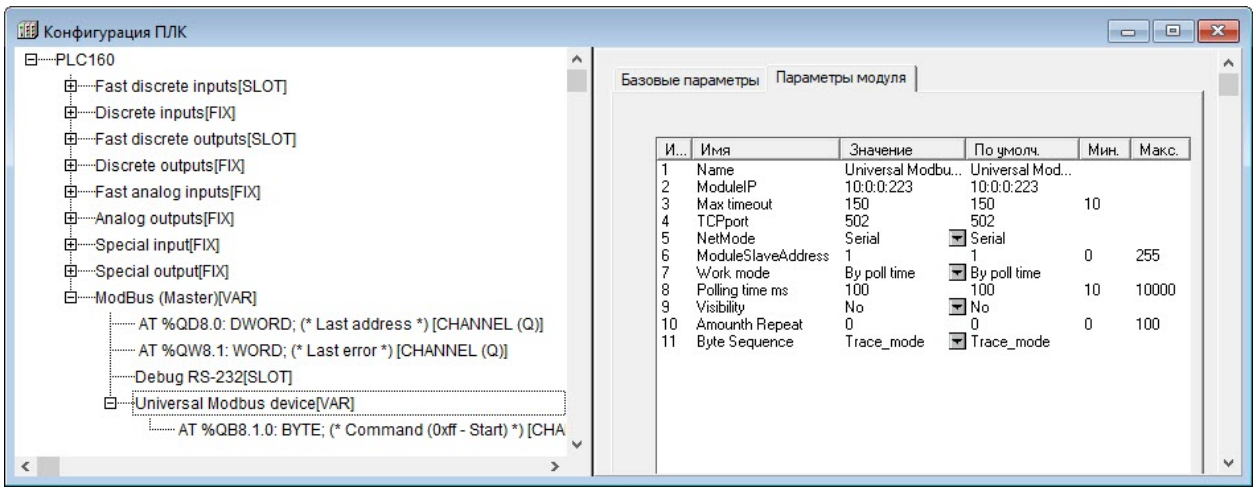


Рисунок 10.24 – Параметры подмодуля Universal ModBus device

Настройка входов и выходов подмодуля

После задания значений параметров подмодуля «Universal Modbus Device» к нему требуется подключить каналы, задающие входные параметры (параметры, значение которых Мастер запрашивает у Slave-устройств) и выходные параметры (значения, которые Мастер передает (записывает) в Slave устройства) подмодуля.

Каналы добавляются выбором команды **Добавить подэлемент → <Наименование подэлемента>** контекстного меню строки «Universal Modbus Device».

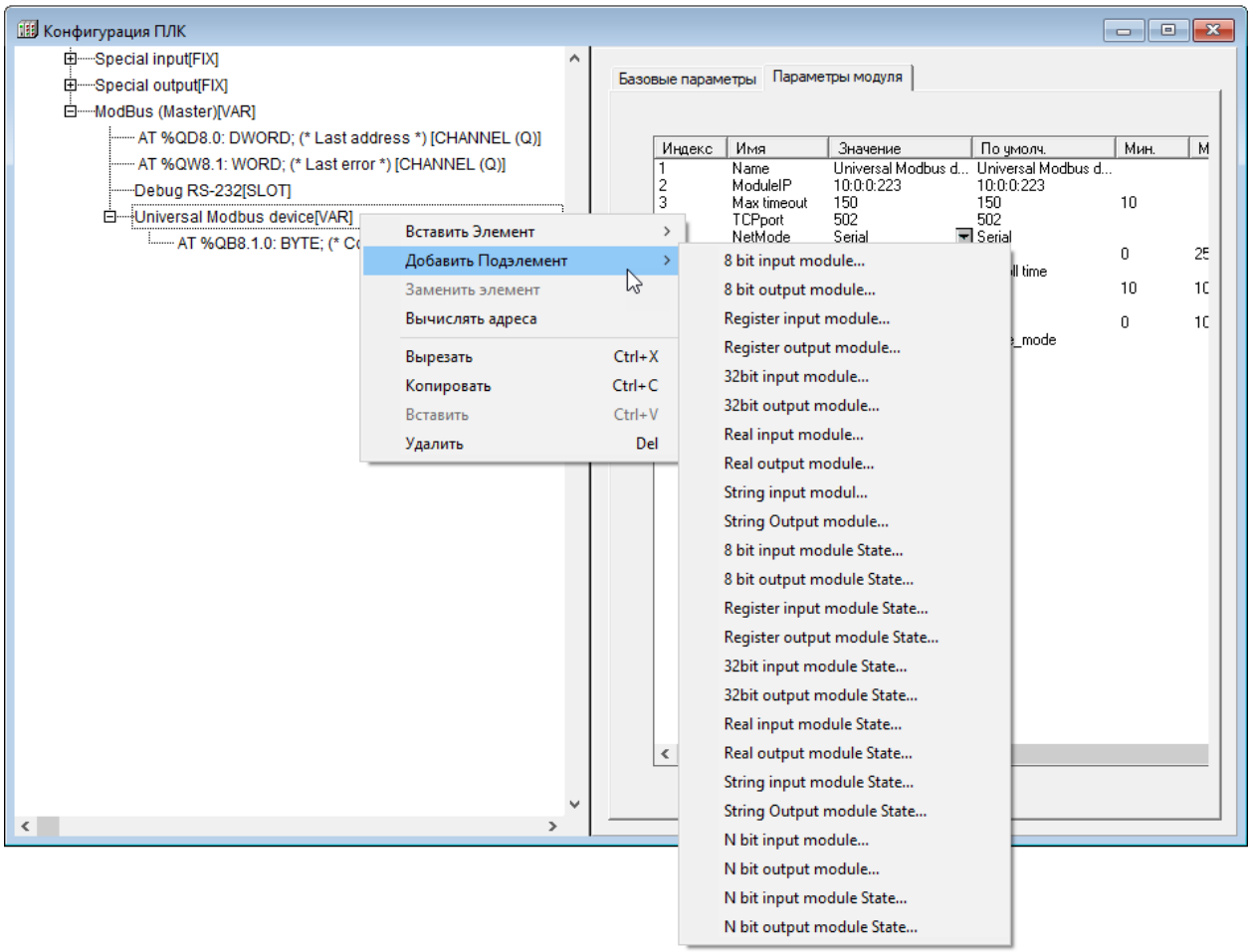


Рисунок 10.25 – Добавление каналов ввода-вывода подмодуля Universal Modbus device

После добавления канала в «Universl Modbus Device» в дерево конфигурации добавляется подкаталог, содержащий внутри себя канал, в котором и отображаются полученное/передаваемое по сети значение.

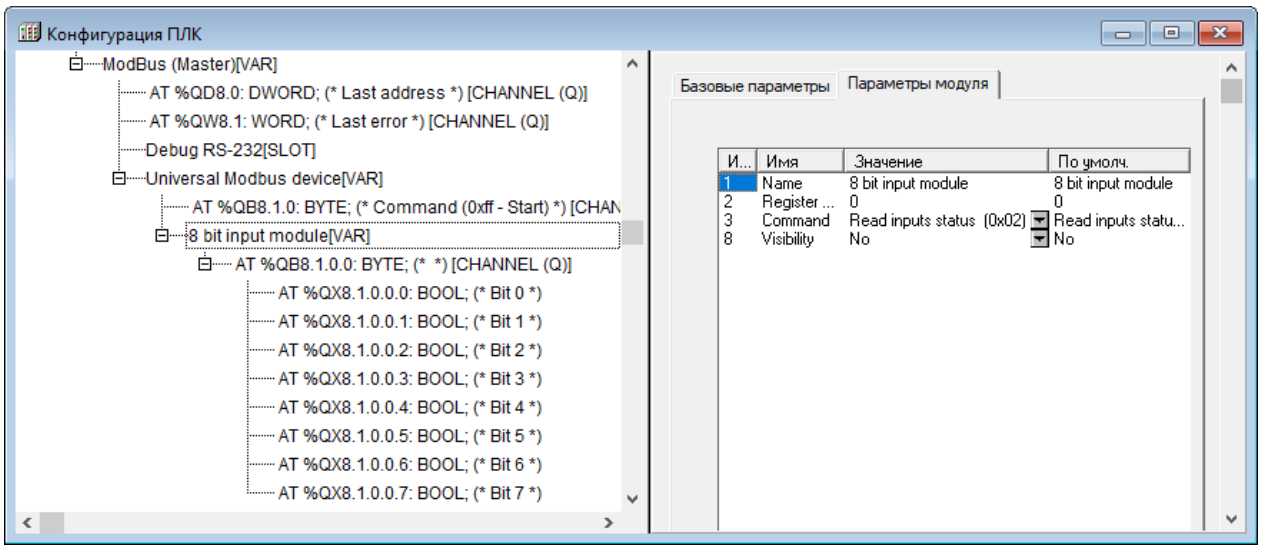


Рисунок 10.26 – Подкаталог канала ввода-вывода

В подмодуль «Universal Modbus Device» могут быть добавлены каналы из таблицы ниже. Каналы (переменные) могут принадлежать следующим типам: REAL (32 бита), STRING (настраивается параметром, по умолчанию – 80 символов), 4 байта, 2 байта или 8 бит.

Таблица 10.1 – Типы каналов подмодуля «Universal Modbus Device»

<table>
<tr><th>Канал</th><th>Размер в памяти</th><th>Command (Команда)</th><th>Чтение/запись</th></tr>
<tr><td>8-bit input module</td><td>8 бит</td><td rowspan="10">Read coils status (0x01),
Read inputs status (0x02),
Read holding registers (0x03),
Read input registers (0x04),
Read bytes (0x70)</td><td rowspan="10">Чтение</td></tr>
<tr><td>Register input module</td><td>2 байта</td></tr>
<tr><td>32-bit input module</td><td>32 бит</td></tr>
<tr><td>Real input module</td><td>32 бита</td></tr>
<tr><td>String input module</td><td>80 бит*</td></tr>
<tr><td>8-bit input module State</td><td>8 бит</td></tr>
<tr><td>Register input module State</td><td>2 байта</td></tr>
<tr><td>32-bit input module State</td><td>32 бита</td></tr>
<tr><td>Real input module State</td><td>32 бита</td></tr>
<tr><td>String input module State</td><td>80 бит*</td></tr>
<tr><td>8-bit output module</td><td rowspan="2">8 бит</td><td rowspan="2">Force multiply coils (0x0f),
Write bytes (0x71)</td><td rowspan="12">Запись</td></tr>
<tr><td>8-bit output module State</td></tr>
<tr><td>Register output module</td><td rowspan="2">2 байта</td><td rowspan="2">Preset single register (0x06),
Write bytes (0x71),
Write multiple registers (0x10)</td></tr>
<tr><td>Register output module State</td></tr>
<tr><td>String output module</td><td rowspan="2">80 бит*</td><td rowspan="2">Force multiple coils (0x0f),
Preset multiple Registers (0x10),
Preset single Register (0x06),
Write bytes (0x71)</td></tr>
<tr><td>String output module State</td></tr>
<tr><td>32-bit output module</td><td rowspan="4">32 бита</td><td rowspan="4">Force multiply coils (0x0f),
Preset multiple Registers (0x10),
Write bytes (0x71)</td></tr>
<tr><td>Real output module</td></tr>
<tr><td>32-bit output module State</td></tr>
<tr><td>Real output module State</td></tr>
<tr><td colspan="4">ПРИМЕЧАНИЕ
* Для каналов типа String указан размер 80 бит, задаваемый по умолчанию. Размер может изменяться пользователем.</td></tr>
</table>

Каналы с обозначением **«State»** кроме канала получения/передачи данных содержат дополнительный управляющий канал (Command) для управления передачей, позволяющий организовать обмен значений отдельно взятых переменных (каналов) по команде пользователя (см. рисунок 10.27).

Запуск/остановка обмена по каналам «Command» управляется значением, записываемым в канал:

- **0xFF** – старт работы данного устройства Modbus;
- если модуль уже запущен, то повторная запись в канал значения **0xFF** приводит к внеочередному запросу одной очередной переменной устройства Modbus;
- **0xFE** – остановка и прекращение всех посылок в сеть;
- если в канал ничего не записано, то устройства опрашиваются автоматически, в порядке очереди.

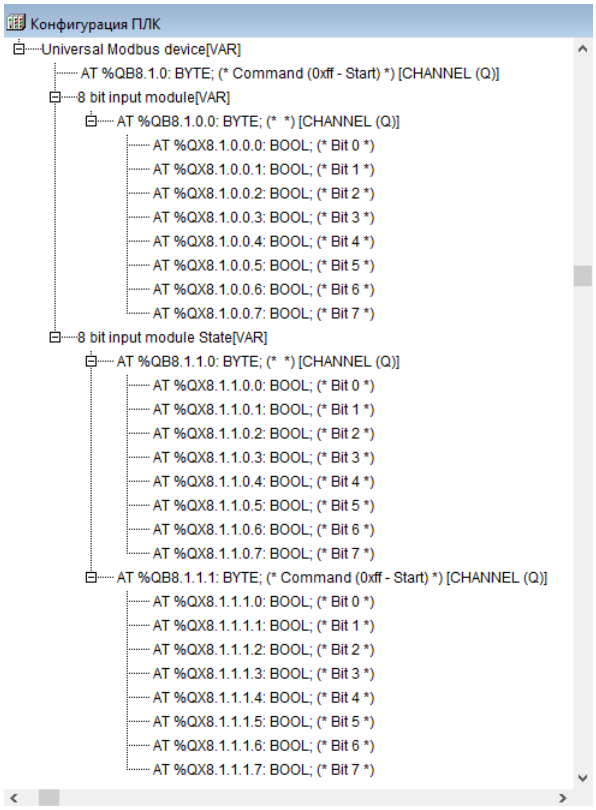


Рисунок 10.27 – Канал ввода и канал ввода типа «State»

Параметры каналов ввода-вывода подмодуля «Universal Modbus Device»:

- **Command (Номер команды протокола)** – номер команды (номер функции) протокола, по которой будет производиться обмен. Команда для обмена определяется Slave-устройством и описана в документации на него. Во время конфигурирования требуемая команда выбирается из списка;
- **Register Address (Адрес регистра Slave-устройства)** – адрес регистра опрашиваемого устройства. Адрес определяется Slave-устройством и описан в документации на него. Во время конфигурирования ПЛК указывается адрес регистра Slave-устройства;
- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

ВНИМАНИЕ
Особенность Modbus для CODESYS: для любого номера команды все параметры хранятся в одной и той же области памяти, и к одному и тому же участку памяти можно обращаться как к разным переменным разных типов. Задавая новые переменные Modbus следует помнить расположение данных в памяти.

10.7.3 Модуль ModBus (Slave)

Модуль «ModBus (Slave)» используется для работы контроллера в сети в подчиненном режиме (slave), т. е. отвечает на запросы устройства, работающего в режиме Master. Во время установки модуля «ModBus (Slave)» выбирается коммуникационный интерфейс для обмена данными с другими устройствами, добавляются и настраиваются требуемые переменные.

Модуль «ModBus (slave)» имеет в своем составе подмодуль **ModBus (FIX)**, см. раздел 10.7.3.1.

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Address (адрес устройства)** – адрес ПЛК, по которому прибор будет опрашиваться в сети Master-устройством (например, панелью оператора). Параметр имеет значения в диапазоне от 1 до 247, значение по умолчанию – 1;
- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Переменные, которыми будет обмениваться ПЛК по протоколу **Modbus**, выбираются пользователем командой контекстного меню «Append Subelements (добавить подэлемент)».

ПРИМЕЧАНИЕ
При случайном отключении питания во время работы ПЛК последние (текущие) значения переменных сохраняются в энергонезависимой памяти и восстанавливаются при возобновлении работы прибора.

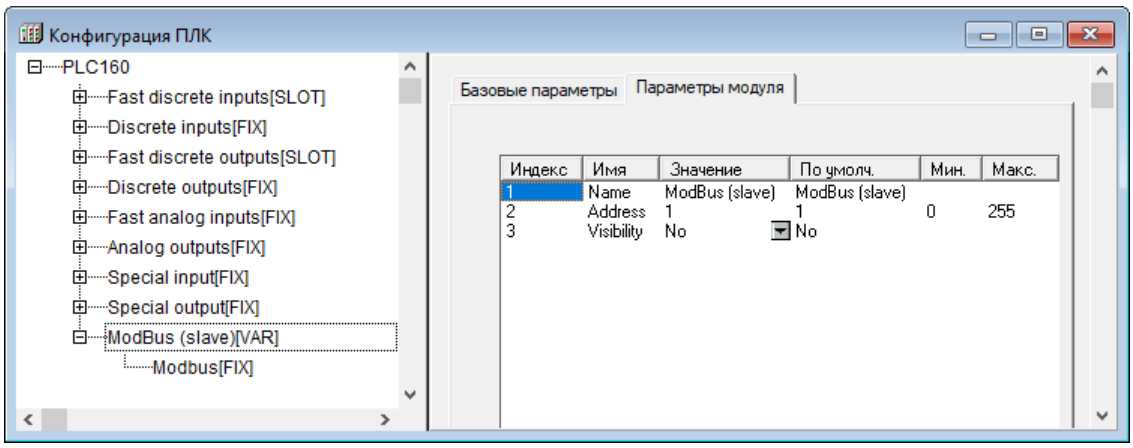


Рисунок 10.28 – Параметры модуля ModBus (Slave)

10.7.3.1 Подмодуль ModBus (FIX). Настройка коммуникационных интерфейсов

При добавлении модуля «ModBus (slave)» в конфигурацию ПЛК в состав модуля уже подключен подмодуль «ModBus (FIX)», к которому, в свою очередь, подключается коммуникационный интерфейс (см. рисунок 10.30).

В ПЛК предусмотрена возможность обмена данными по интерфейсам:

- RS-232;
- RS-485;
- TCP (Ethernet).

Для работы с разными коммуникационными интерфейсами в ПЛК предусмотрены соответствующие подмодули (подэлементы). Подэлементы подключаются выбором требуемой команды **Добавить подэлемент** → **<Имя подэлемента>** контекстного меню (см. рисунок 10.29).

Во время работы ПЛК в режиме «Ведомый (slave)» можно использовать нескольких разных портов, т. е. опрос может вестись по разным интерфейсам. Таким образом, подключая несколько разных портов, один модуль можно соединить с разными Master-устройствами по разным физическим линиям (и интерфейсам). Этот прием может использоваться, например, для создания межсетевого шлюза и/или линии резервного управления (например, подключения SCADA-системы в резервном варианте): например, при выходе из строя порта Ethernet (авария, сбой и пр.), вышестоящая SCADA понимает, что произошел сбой, и начинает информационный обмен с устройством по резервной линии – через COM-порт. Скорость обмена информацией уменьшается, но функционирование устройства продолжается.

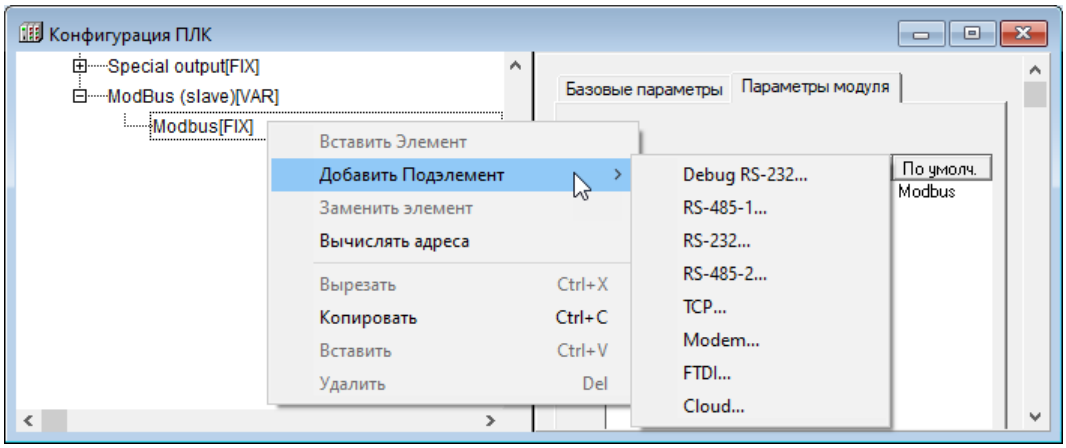


Рисунок 10.29 – Контекстное меню добавления подмодулей настройки коммуникаций

Количество подключаемых портов ограничено конструкцией ПЛК.

Пример подключения нескольких портов представлен на рисунке ниже.

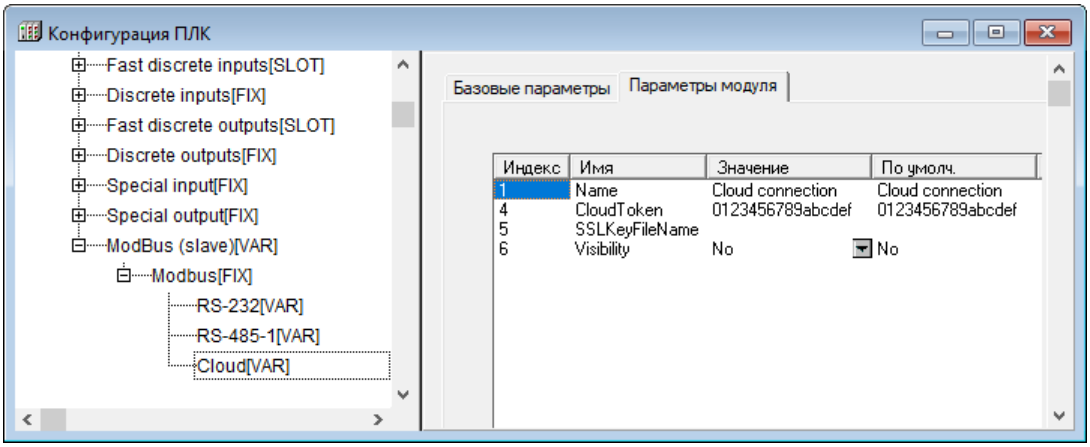


Рисунок 10.30 – Подключение нескольких портов в подмодуле ModBus (FIX)

Настройка входов и выходов подмодуля

После задания значений параметров подмодуля «ModBus (Slave)» к нему подключаются каналы (переменные) входа/выхода.

Каналы добавляются выбором команды **Добавить подэлемент → <Наименование подэлемента>** контекстного меню строки «ModBus (Slave)».

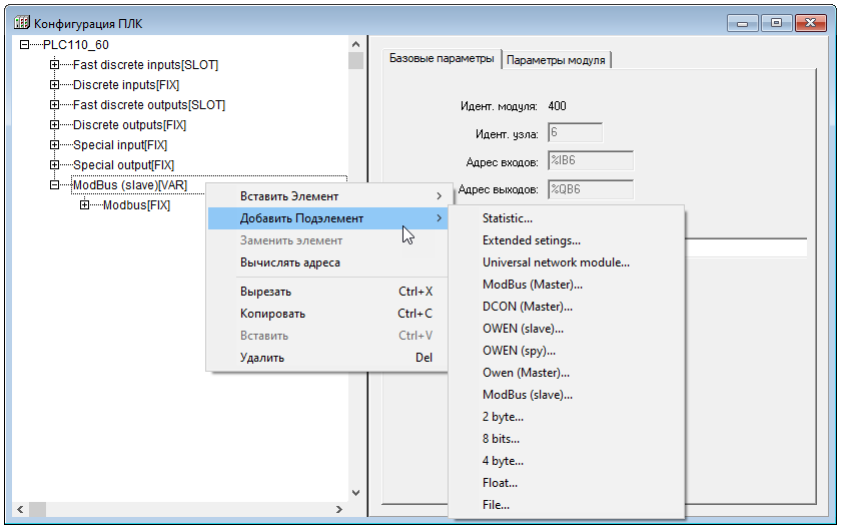


Рисунок 10.31 – Добавление переменных в модуль «ModBus (Slave)»

Возможные типы каналов (переменных) приведены в таблице ниже.

Таблица 10.2 – Типы каналов (переменных) модуля «ModBus (Slave)»

Типы передаваемых переменных	Канал	Размер в памяти	Команды считывания	Команды записи
1…8 (8 bits) бит	8 bit	8 бит	01, 02	05
Word, INT, (2 byte) 16 бит	2 byte	2 байта	03, 04	6, 16
Dword, DINT, (4 byte) 32 бита	4 byte	4 байта		16
Real (Float)	Float	32 бита		
Файл (File)	File*	–	20	–

ПРИМЕЧАНИЕ
* Об использовании подэлемента «File» см. раздел 10.7.3.2.

Во время работы по протоколу ModBus обращение Master-устройства к используемым переменным производится по адресу переменной в памяти модуля «ModBus (Slave)». Адресация переменных автоматически формируется CODESYS при программировании контроллера, но не отображается в интерфейсе программы. Поэтому для обращения к переменной ее адрес следует вычислить, учитывая особенности распределения адресов ячеек памяти («выравнивание» переменных).

Выравнивание адресации

Все используемые переменные хранятся в одном адресном пространстве, и считывать данные из этого пространства можно разными способами.

Типы переменных для адресации:

- **Логический (BOOL)** – занимает один бит памяти и адресуется как бит с заданным номером в пронумерованном байте;
- **Короткие знаковые и беззнаковые целые, байты (BYTE, SINT, USINT)** – занимают восемь бит и имеют адрес, который увеличивается на единицу для каждых следующих восьми бит, то есть адресуется каждый байт;
- **Знаковые и беззнаковые целые, слова (WORD, INT, UINT)** – занимают 16 бит и имеют адрес, который увеличивается на единицу для каждых следующих шестнадцати бит, то есть адресуется каждые два байта, и началу следующей по адресу ячейке с двухбайтовой переменной соответствует начало байта с адресом, равным адресу слова, умноженному на два;
- **Знаковые и беззнаковые двойные целые, двойные слова (DWORD, DINT, UDINT)** – занимают 32 бита и имеют адрес, который увеличивается на единицу для каждых следующих 32 бит, то есть адресуются каждые четыре байта, и началу следующей по адресу ячейки с четырехбайтовой переменной соответствует байт с адресом, равным адресу двойного слова, умноженному на четыре.

Данный способ адресации наглядно показан в таблице 10.3.

Автоматическая адресация переменных производится последовательно, начиная с нулевого адреса (как для битовых переменных, так и для переменных, передаваемых регистрами).

Если в модуле используются переменные одного типа, то при запросе устройством – Мастером регистра с адресом «0», модуль считывает первые два байта, для регистра с адресом «1» – вторые два байта и т. д.

Если переменные имеют длину более двух байт, то во время запроса регистра с адресом «0», модуль считывает первые два байта первой переменной, для регистра с адресом «1» – вторые два байта первой переменной и т. д.

Пример

Таблица 10.3 – Адресация битов и регистров в памяти модуля

Адрес бита	Адрес байта (BYTE, SINT, USINT)	Адрес двухбайтовой переменной (WORD, INT, UINT)	Адрес четырехбайтовой переменной (DWORD, DINT, UDINT)
0…7	0x0000	0x0000	0x0000
8…15	0x0001		
16…23	0x0002	0x0001	
24…31	0x0003		
32…39	0x0004	0x0002	0x0001
40…47	0x0005		
48…55	0x0006	0x0003	
56…63	0x0007		

Если в модуле используются переменные разных типов (например, одновременно восьмибитный, двухбайтный и четырехбайтный), то во время распределения адресов CODESYS «выравнивает» адресов переменных – упорядочивать адреса переменных в памяти модуля. Упорядочение адресов заключается в организации памяти таким образом, что переменные размером 8 бит, 2 байта и 4 байта располагаются только по определенным адресам: четырехбайтным переменным присваиваются адреса, кратные 2, двухбайтным – кратные 1. Вне зависимости от порядка задания переменных, выравнивание назначает переменным адреса, кратные их длине.

Первая восьмибитная переменная будет расположена в 0… 7 битах памяти модуля, вторая – в 8… 15 и т. д. Если вторая переменная двухбайтная, то она будет располагаться в 16…31 битах, т. е.,

по любому (то есть, кратному 1) адресу. Четырехбайтная переменная займет следующее свободное место, кратное 2.

Такой порядок размещения переменных в памяти модуля может образовать адресные пространства, не занятые переменными. Эти пространства не отображаются в области ввода-вывода, но они обязательно должны учитываться при организации опроса переменных. Учитывать этот порядок размещения переменных следует еще на стадии задания переменных.

Пример

Таблица 10.4 – «Выравнивание» адресации переменных в памяти модуля

Адрес регистра	Адрес бита	Адресация переменных
0x00	0…7	8 бит (1 байт)
	8…15	↓ **Сдвиг выравнивания: пропущенные адреса**
0x01	16…23	2 байта
	24…31	
0x02	32…39	8 бит (1 байт)
	40…47	8 бит (1 байт)
0x03	48…55	8 бит (1 байт)
	56…63	↓ **Сдвиг выравнивания: пропущенные адреса**
0x04	64…71	4 байта
	72…79	
0x05	80…87	
	88…95	

ВНИМАНИЕ

В некоторых версиях CODESYS во время подключения модулей «ModBus (Slave)» в конфигурацию ПЛК могут возникнуть ошибки выравнивания адресации. Например, один и тот же адрес может быть ошибочно присвоен двум каналам разных модулей.

Основной способ исправления ошибки выравнивания – вставить перед неправильно выровненным модулем модуль другого типа. Автоматическое переназначение адресов исправит ошибку.

В случае возникновения подобной проблемы с выравниванием пространства адресации следует обратиться в группу технической поддержки компании «ОВЕН» или описать свою проблему на форуме сайта www.owen.ru – технические специалисты компании посоветуют способ исправить ошибку, чтобы она не возникала в дальнейшем.

10.7.3.2 Подмодуль «File». Передача архивных данных

Подмодуль «File» используется для передачи архивных данных (файла) при помощи функции 20 протокола Modbus.

Записанные с помощью модуля **Archiver** (см. раздел 10.7.10) архивные данные (файл) могут быть переданы по последовательному интерфейсу по протоколу Modbus с помощью специализированной функции № 20. Данные могут быть приняты специализированным ПО на ПК или OPC-сервером, поддерживающим работу с этой функцией (например Lectus OPC).

Данные из файла передаются по коммуникационному интерфейсу, который настраивается так же, как при передаче обычных данных.

Для реализации передачи архивных файлов следует подключить к модулю «ModBus (Slave)» подмодуль «File». Параметры модуля:

- **File name (имя)** – имя файла с архивной информацией, находящегося на Flash-диске ПЛК, внешнем Flash-диске либо на RAM-диске ПЛК, записанного модулем **Archiver** (см. раздел 10.7.10). Расположение файла задается с помощью префикса, так же, как и в модуле работы с файлами (см. раздел 7.4.1);
- **Amount Byte (размер записи)** – размер одной архивной записи в байтах (в запросе записей может быть больше одной);

- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

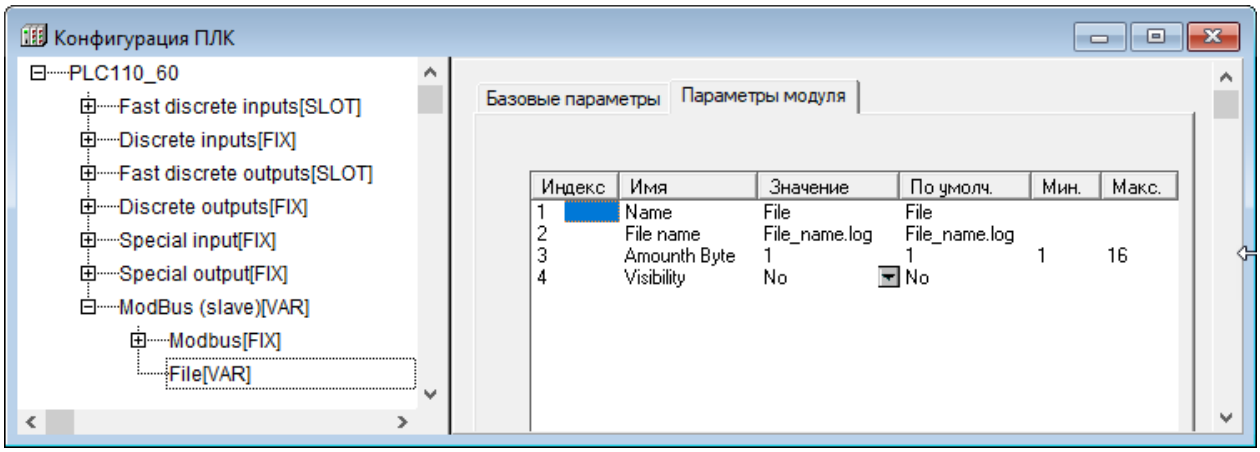


Рисунок 10.32 – Параметры подмодуля «File»

В случае необходимости работать с несколькими архивными файлами следует последовательно подключить в модуль **ModBus (Slave)** требуемое количество подмодулей **File**, настроив каждый из них. Во время настройки следует иметь в виду, что в запросе нет возможности передать имя файла. Поэтому соответствие имени файла (которых может быть много) и запроса осуществляется следующим образом: номер файла в запросе соответствует позиции файла в дереве режима «Конфигурация ПЛК» (PLC Configurations), начиная с нуля. Таким образом, запрос с нулевым номером файла будет читать данные из первого файла в конфигурации подмодуля, первый – из второго, и т. д. Если запрошен файл, которого нет на диске (или в конфигурации он не указан), то выдается код ошибки с номером 0x04.

ПРИМЕЧАНИЕ
Чтобы Lectus OPC работал с этой функцией, следует дополнительно поместить в рабочую директорию библиотеку ModBus.dll.

Форматы запросов и ответов

Информация данной части раздела носит справочный характер – она необходима в случае создания собственного ПО для обмена, запускаемого на ПК. В случае использования готового ПО, поддерживающего работу с функцией 20 протокола Modbus, особенности обмена скрыты от пользователя.

Формат запросов и ответов приведен в таблице ниже.

Таблица 10.5 – Формат запросов и ответов

Function Code	0x14	Byte	Код функции
Формат запросов			
Byte count	0x07	Byte	Количество байт, следующих ниже
Referens Type	0x06		Подфункция (здесь константа = 6)
Hi File number	–		Старший байт номера требуемого файла
Lo File number			Младший байт номера требуемого файла
Hi Rec addr			Старший байт адреса записи в файле
Lo Rec addr			Младший байт адреса записи в файле
Hi Rec num			Старший байт количества запрашиваемых записей
Lo Rec num			Младший байт количества запрашиваемых записей
Формат ответов			
Byte count	0x07	Byte	Количество байт, следующих ниже
Byte count			Количество байт, следующих ниже (необходимо по стандарту)
Referens Type	0x06		Подфункция (здесь константа = 6)

Продолжение таблицы 10.5

Function Code	0x14	Byte	Код функции
Data	–	Byte*Rec_ num*Amount_ byte	Данные длиной Rec num из (запроса), умноженные на длину одной записи из конфигурации

ПРИМЕЧАНИЕ

1. File number может принимать значение от 0x0000 до 0xffff.
2. Rec number может принимать значение от 0x0000 до 0xffff.
3. Rec addr может принимать значение от 0x0000 до 0xfffe.
4. В случае ошибки возвращается стандартный код ошибки Modbus 0x02.
5. При запросе с адресом Recaddr = 0xffff происходит удаление файла.

10.7.4 Модуль «DCON (Master)»

Модуль «DCON (Master)» используется для работы контроллера по протоколу DCON в режиме Мастера сети, т. е. для опроса и контроля других устройств, работающих в сети в подчиненном режиме (Slave). Во время работы по протоколу DCON ПЛК может функционировать только в режиме Мастера сети.

Во время установки модуля «DCON (Master)» требуется выбрать коммуникационный интерфейс для обмена данными с другими устройствами, добавить и настроить требуемые переменные.

Модуль имеет один параметр: **«Visibility (Видимость)»**, который задает видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Во время опроса модулем «DCON (Master)» подчиненных устройств информация о ходе обмена записывается в канал модуля:

- **Last Error** – код ошибки. В переменной отображается код ошибки, если информационный обмен прошел неудачно, что требуется для корректности работы опрашиваемого устройства. Коды ошибок данного модуля представлены в приложении Сообщения об ошибках в ПЛК .

Настройка коммуникационных интерфейсов

При добавлении модуля «DCON (Master)» в конфигурацию ПЛК в состав модуля уже подключен порт Debug RS-232. Для работы через другой коммуникационный интерфейс порт Debug RS-232 можно заменить требуемым последовательным портом или модемом (выбрать команду **Заменить элемент** → **<Элемент>** контекстного меню строки «Debug RS-232»).

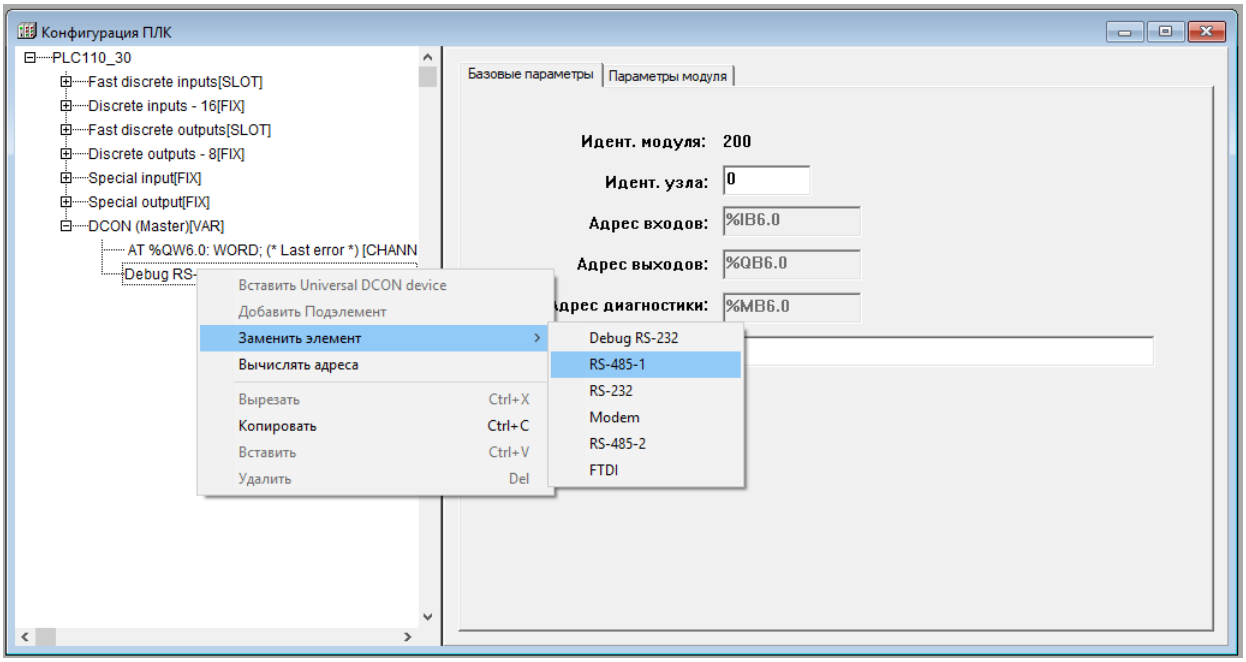


Рисунок 10.33 – Выбор коммуникационных интерфейсов «DCON (Master)»

10.7.4.1 Подмодуль Universal DCON Device

Для добавления в список опроса, проводимого мастером сети, устройств, работающих в режиме DCON (Slave), следует в модуле DCON (Master) добавить подмодуль «Universal DCON Device» (универсальное устройство DCON). Для добавления в список нескольких опрашиваемых устройств процедуру следует повторить столько раз, сколько устройств должно быть подключено: для каждого устройства должен быть добавлен соответствующий ему модуль Universal DCON Device с индивидуальными настройками.

Чтобы добавить подмодуль «Universal DCON Device» в конфигурацию, следует выбрать команду **Добавить подэлемент → Universal DCON Device** контекстного меню строки «ModBus (Master)».

Добавленные подмодули «Universal DCON device» опрашиваются последовательно, в порядке следования в конфигурации (если другой порядок не задан отдельно, в настройках модуля).

Подмодуль «Universal DCON device» имеет один канал **Status**:

- если в него записывается значение **0x00FF**, то происходит старт работы данного устройства DCON;
- если модуль уже запущен, то повторная запись в канал значения **0x00FF** приводит к внеочередному запросу одной очередной переменной устройства DCON;
- если в канал записано значение **0x00FE**, то происходит его остановка и прекращение всех посылок в сеть;
- если в канал ничего не записано, то опрос устройства производится автоматически, в порядке очереди.

В случае необходимости устройство можно исключить из списка опроса, подав соответствующую команду в канал Start\Stop.

При добавлении модуля «Universal DCON Device» его параметры и идентификаторы не привязаны к конкретному внешнему устройству (модулю ввода-вывода, операторской панели). Тип внешнего устройства указывается заданием значений параметров модуля (конфигурированием модуля).

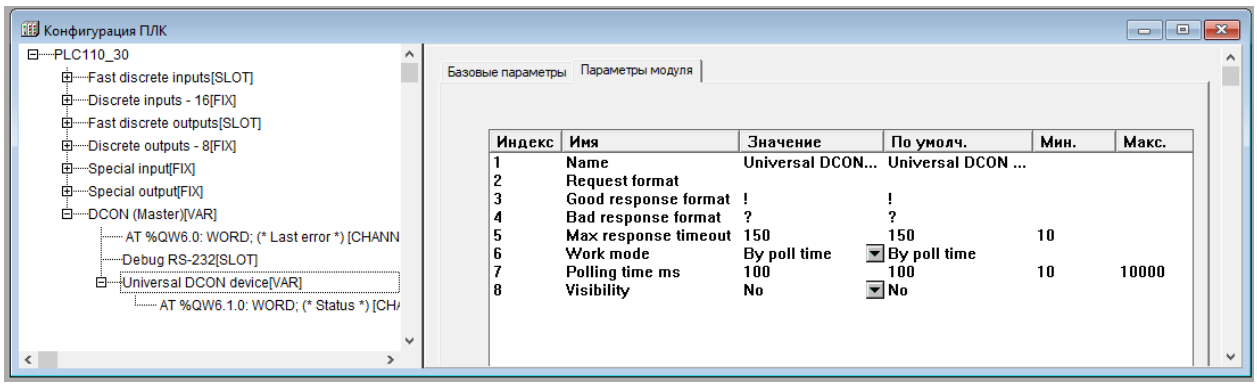


Рисунок 10.34 – Параметры подмодуля Universal DCON device

Параметры подмодуля «Universal DCON device»:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Request format (Формат запроса)** – формат запроса, может быть любым, ограничения не накладываются (см. ниже);
- **Good response format (Формат правильного ответа)** – формат правильного ответа, значение по умолчанию – **«!»** (см. ниже);
- **Bad response format (Формат неправильного ответа)** – формат неправильного ответа, значение по умолчанию – **«?»** (см. ниже);
- **Max response timeout (Максимальное время ответа)** – время, за которое опрашиваемый прибор должен ответить на запрос **DCON (Master)**. Если в течение этого времени прибор не отвечает, то считается, что он отключен или произошел обрыв линии связи. Информация об этом заносится в переменную **«Код последней ошибки» (Last error)**. Значение сверху не ограничено, может быть любым, в том числе дробным, но не меньше 10 мс, значение по умолчанию – 150 мс;
- **Work mode (Режим работы)** – режим работы модуля **«DCON (Master)»** при опросе внешних устройств. Значения выбираются из списка (значение по умолчанию – «By poll time»):
  - **By poll time (по времени)** – контролируемые устройства опрашиваются с периодичностью, заданной в параметре **«Период опроса устройства» (Polling time)**;
  - **By value change (по изменению значения переменных)** – модуль **DCON (Master)** генерирует запрос устройству при изменении значений выходных переменных модуля;
  - **Both (оба варианта)** – опрос производится с временным интервалом, заданным в параметре **Polling time** и когда изменяются значения выходных переменных;
  - **By command (по команде)** – производится однократная посылка запроса, когда в канал **Статус (Status)** модуля **Универсальное устройство DCON** записывается значение **0x00FF**.
- **Polling time (Период опроса устройства, в мс)** – диапазон значений от 10 до 10000, значение по умолчанию – 100;
- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

ПРИМЕЧАНИЕ
В Мастере, когда он работает в режиме **«По изменению значения переменных»** или **«По команде»** , нельзя ставить значение параметра **Polling time** слишком маленьким. По умолчанию его значение 100 мс. Однако, если на реальном проекте будет замечено, что Мастер при загрузке программы или при **Login** формирует лишние пакеты и/или запросы, которых не должно быть, значение параметра увеличивают (до 200, 300 и т. д.) до предотвращения появления ложных пакетов.

Параметры «Формат…»

В протоколе **DCON** при организации опроса устройств создается строка запроса. При ее посылке опрашиваемое устройство может вернуть два варианта ответа: ответ правильный (команда распознана, данные есть) – один формат, и ответ неправильный (не распознана команда, нет данных и/или пр.) – другой формат.

Строки «Формат запроса» (Request format), «Формат правильного ответа» (Good response format), «Формат неправильного ответа» (Bad response format) используются для задания формата запроса DCON (Master) и разбора правильного/неправильного ответа.

Строки формата ответа могут не задаваться, если устройство не отвечает на запрос.

Строка формата представляет собой строку, содержащую символы и спецкоманды:

- **символы** – любой символ, кроме служебных, к которым относятся символы **«$»** (знак доллара), **«[»** и **«]»** (открывающая и закрывающая квадратные скобки);

  ПРИМЕЧАНИЕ
  При необходимости вывести служебный символ в качестве обычного, он вводится в строку два раза подряд.

- **спецкоманда** имеет формат: **[{модификатор}действие]**:

  – **модификатор** – количество символов, обрабатываемых действием. Представляет собой десятичное целое число. Может быть у всех действий, кроме вычисления контрольных сумм. Наличие модификатора необязательно, значение по умолчанию = 1;

  – **действие** – отображается в строке спецкоманды одним из символов – **D**, **H**, **F**, **S**, *****, **+**, **%**. Регистр символов значения не имеет.

  Символы соответствуют следующим видам действий:

  – **D** – представляет передаваемую переменную в ASCII-символах в десятичном формате (без знака) или преобразует ASCII-строку из десятичного формата (без знака) в принимаемую переменную. Количество символов задается модификатором;

  – **H** – представляет передаваемую переменную в ASCII-символах в шестнадцатеричном формате или преобразует ASCII-строку из шестнадцатеричного формата в принимаемую переменную. Количество символов задается модификатором;

  – **F** – представляет передаваемую переменную в ASCII-символах в десятичном формате со знаком, разделителем целой и дробной части числа (точкой). Строка имеет фиксированное число символов, заданное модификатором. Для принимаемых переменных производит обратное преобразование из ASCII-строки в число;

  – **S** – осуществляет прямое копирование из передаваемой строковой переменной в строку запроса числа символов, заданного **модификатором** или обратное копирование из строки ответа в принимаемую переменную строкового типа;

  – ***** – задает в строке ответа набор символов, которые надо пропустить. Количество символов может быть задано модификатором;

  – **+** – вставляет в строку запроса контрольную сумму или получает ее в строке ответа. Контрольная сумма вычисляется путем сложения с переполнением по модулю 256. Данное действие не может иметь модификатора;

  – **%** – вставляет в строку запроса контрольную сумму или получает ее в строке ответа. Контрольная сумма вычисляется по восьмибитному полиному (DOW-CRC). Данное действие не может иметь модификатора.

  ПРИМЕЧАНИЕ
  Во время работы ПЛК по протоколу DCON есть три варианта работы: без расчета контрольных сумм, с расчетом контрольных сумм путем сложения значений всех символов и с расчетом контрольных сумм восьмибитовых. Вариант работы пользователь выбирает в соответствии с тем, какой вариант расчета контрольной суммы используется в опрашиваемом приборе.

  Используются следующие алгоритмы преобразования:

  – **при формировании запроса** – все символы вне спецкоманд копируются в строку запроса без изменения, спецкоманды заменяются на значения передаваемых (выходных) переменных. Значения переменных кодируются в формате, заданном **действием**, число символов соответствует **модификатору**;

  – **при разборе ответа** – все символы вне спецкоманд сравниваются с соответствующими позициями ответа и, при нахождении различия, вырабатывается сообщение об ошибке. Данные в позициях ответа, соответствующих спецкомандам, преобразуются и сохраняются в соответствующих принимаемых (входных) переменных.

  Если запрос жестко фиксированный, т. е. в строке не содержатся изменяемые данные, то строка набивается без каких-либо команд, в таком виде отсылается. В строку может быть добавлена контрольная сумма.

Аналогично с ответом: если приходит строка, не содержащая каких-либо данных (в конце может быть контрольная сумма), это означает, что прибор работает, реагирует и факт получения ответа от прибора уже является информацией.

Настройка входов и выходов подмодуля

После задания значений параметров подмодуля «Universal DCON Device» к нему требуется подключить каналы, задающие входные параметры (параметры, значение которых Мастер запрашивает у Slave-устройств) и выходные параметры (значения, которые Мастер передает – записывает – в Slave устройства) подмодуля.

Каналы добавляются выбором команды **Добавить подэлемент → <Наименование подэлемента>** контекстного меню строки подмодуля «Universal DCON Device».

В подмодуль «Universal DCON Device» можно добавить каналы из таблицы ниже.

Каналы (переменные) могут принадлежать следующим типам: REAL (32 бит, целое беззнаковое, с плавающей точкой или строка), STRING (16 байтовая строка), 8, 16 или 32 бит.

Таблица 10.6 – Типы каналов модуля «Universal DCON Device»

Канал	Размер в памяти	Направление пересылки данных
8-bit input	8 бит	Чтение
16-bit input	16 бит	
32-bit input	32 бита	
Float input		
String input	16 байт (строка)	
8-bit input	8 бит	Запись
16-bit input	16 бит	
32-bit input	32 бита	
Float input		
String input	16 байт (строка)	

Тип и порядок расположения входных и выходных переменных в модуле должны соответствовать строкам команд в полях **«Request format»** и **«Good response format»** (см. выше).

Примеры настройки модуля **DCON (Master)** для опроса устройств ввода-вывода представлены в приложении Примеры настройки опроса для модуля «DCON (Master)» .

10.7.5 Модуль «Owen (Master)»

Модуль «Owen (Master)» используется для работы контроллера в режиме Мастера сети, т. е. для опроса и контроля других устройств, работающих в сети по протоколу ОВЕН в подчиненном режиме (Slave) – например, модули ввода-вывода, панели оператора, частотные преобразователи и т. д. (протокол предназначен для описания процесса обмена информацией между приборами компании «ОВЕН» и между приборами и ПК на базе сети RS-485).

ПРИМЕЧАНИЕ
При случайном отключении питания в процессе работы ПЛК последние (текущие) значения переменных сохраняются в энергонезависимой памяти и восстанавливаются при возобновлении работы прибора.

Во время установки модуля «Owen (Master)» следует выбрать коммуникационный интерфейс для обмена данными с другими устройствами, добавить и настроить требуемые переменные.

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Max response delay, ms (Максимальное время задержки ответа, мс)** – время, за которое опрашиваемый прибор должен ответить на запрос Owen (Master). Если прибор не отвечает, то считается, что прибор отключен или произошел обрыв линии связи. Диапазон значений от 0 до 32000, значение по умолчанию – 50;
- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

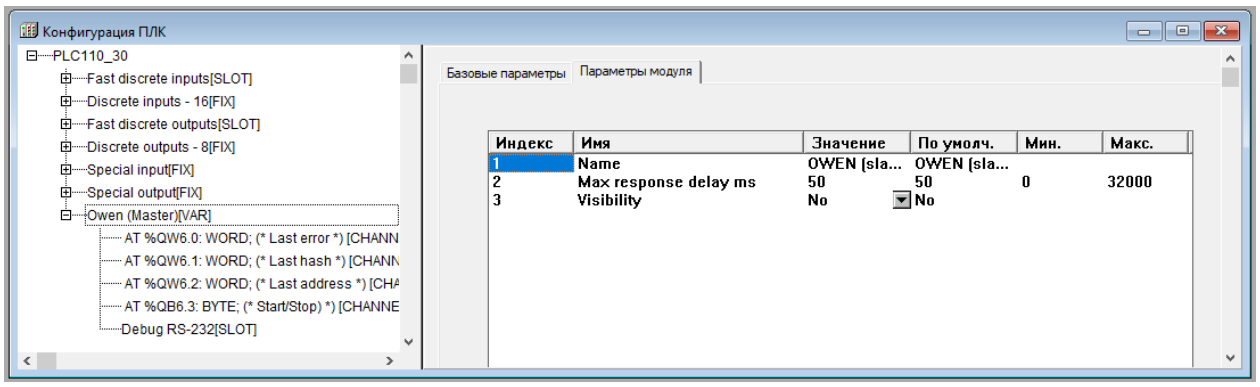


Рисунок 10.35 – Каналы и параметры модуля «Owen (Master)»

Во время опроса модулем «Owen (Master)» подчиненных устройств информация о ходе обмена записывается в соответствующих каналах его переменных.

Каналы модуля:

- **Last address (Последний адрес)** – адрес последнего прибора, по которому обращался Owen (Master). При использовании 8 бит адреса – восьмибитный адрес при отображении умножается на 2^3 (т. е. на 8), при 11 – не умножается;
- **Last error (Код последней ошибки)** – код ошибки, которая произошла при последнем опросе;
- **Last Hash (Последний Hash-код)** – hash-код параметра, который фигурировал в последнем опросе;
- **Start/Stop (Старт/Стоп)** – используется для управления включением/выключением работы модуля мастера: если в канал записывается значение **0x00FF**, то происходит старт работы модуля, если же в канал записано значение **0x00FE**, то происходит его остановка и прекращение всех посылок в сеть. Если модуль уже запущен, то повторная запись в канал значения **0x00FF** приводит к внеочередному запросу очередной переменной протокола ОВЕН.

Коды ошибок работы ПЛК и пояснения к ним представлены в приложении Сообщения об ошибках в ПЛК .

Настройка коммуникационных интерфейсов

Во время добавлении модуля «OWEN (Master)» в конфигурацию ПЛК в состав модуля уже подключен порт Debug RS-232 (см. рисунок 10.36). Если необходимо использовать другой коммуникационный интерфейс, порт по умолчанию можно заменить требуемым последовательным портом или модемом (выбрать команду **Заменить элемент** → **<Элемент>** контекстного меню строки «Debug RS-232»).

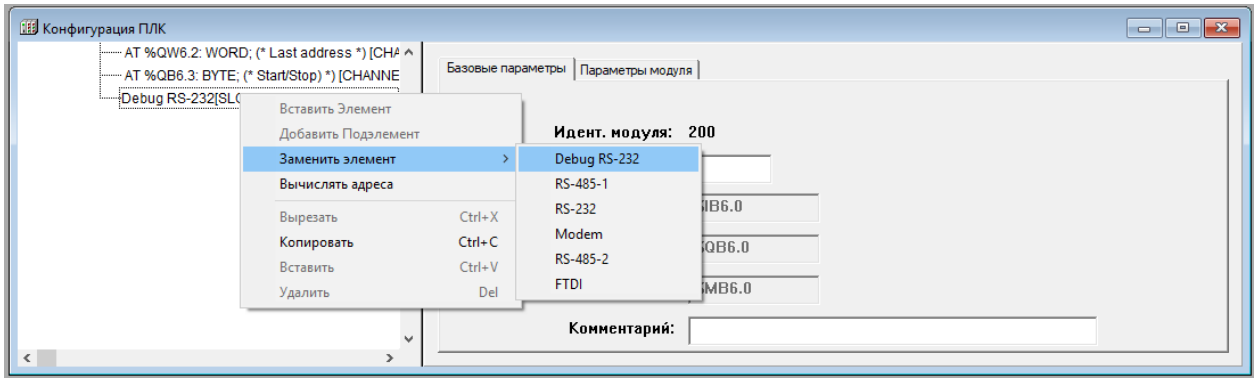


Рисунок 10.36 – Выбор коммуникационных интерфейсов

Настройка входов и выходов

После задания значений параметров модуля «OWEN (Master)» к нему требуется подключить каналы, задающие входные параметры (параметры, значение которых Мастер запрашивает у Slave-устройств) и выходные параметры (значения, которые Мастер передает – записывает в Slave устройства) модуля.

Каналы добавляются выбором команды **Добавить подэлемент** → **<Наименование подэлемента>** контекстного меню строки «OWEN (Master)».

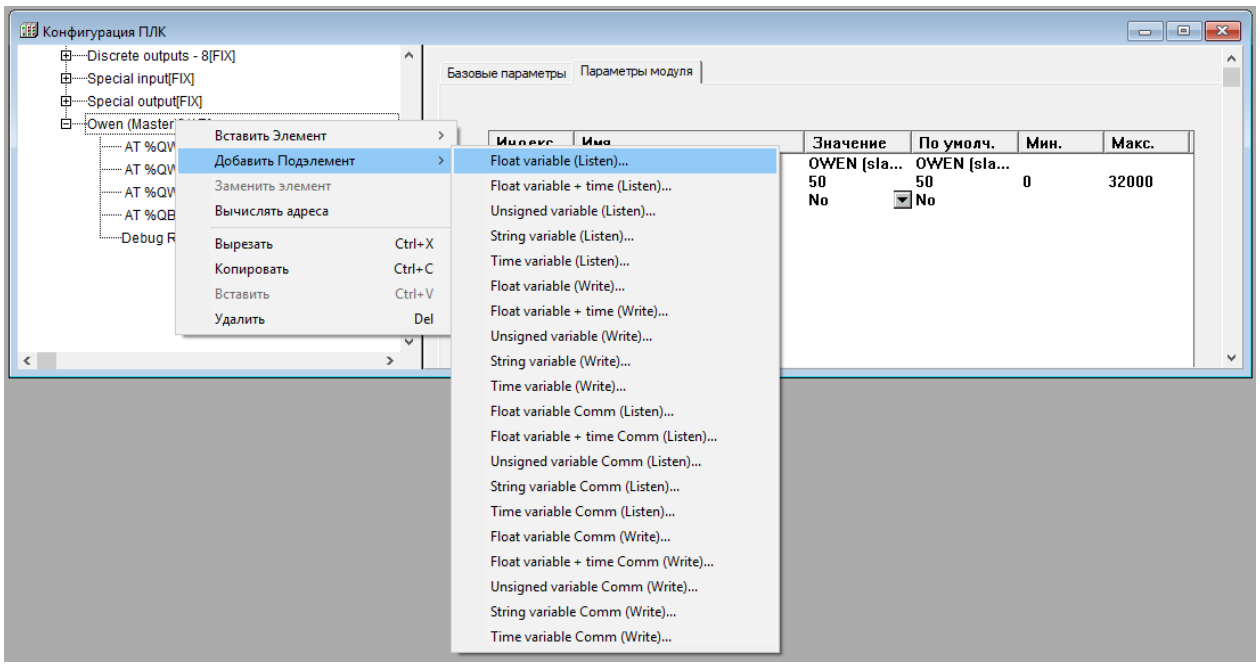


Рисунок 10.37 – Добавление каналов ввода-вывода

При добавлении канала в «OWEN (Master)» в дерево конфигурации добавляется подкаталог, содержащий внутри себя канал, в котором и отображаются полученное/передаваемое по сети значение.

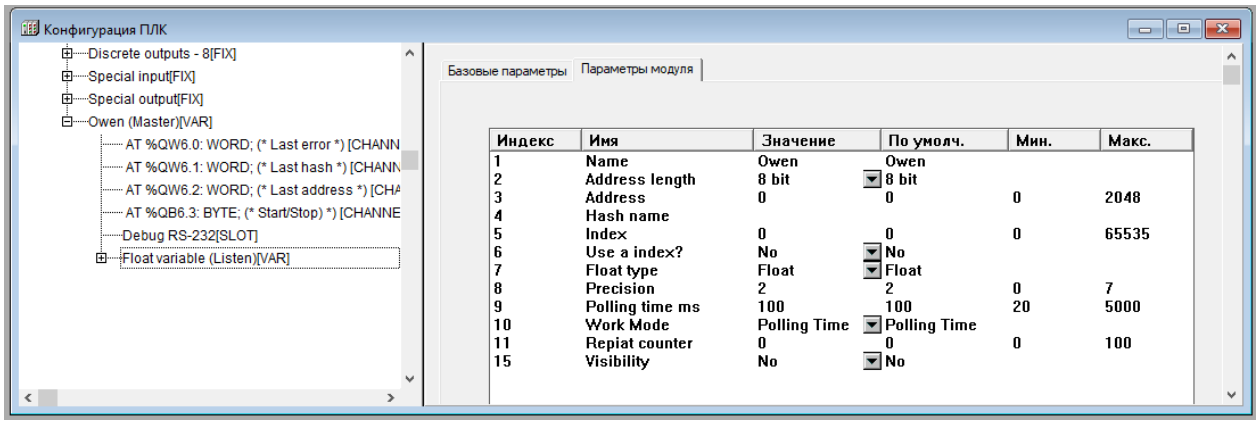


Рисунок 10.38 – Каналы и параметры переменной

В модуле« Owen (Master)» могут быть использованы следующие типы переменных:

- для чтения («Listen»);
- для записи («Write»);
- для чтения («Listen») с дополнительным командным управляющим каналом («Comm»);
- для записи («Write») с дополнительным командным управляющим каналом («Comm»).

Переменные различаются по типу данных:

- **Float variable** – число с плавающей точкой;
- **Float variable + time** – число с плавающей точкой с модификатором времени;
- **Unsigned variable** – целочисленная переменная;
- **Unsigned variable + time** – целочисленная переменная с модификатором времени;
- **String variable** – строковая переменная, максимальная длина – 15 символов, в соответствии со стандартом протокола ОВЕН;
- **String variable + time** – строковая переменная с модификатором времени;
- **Time variable** – позволяет передать время, в протоколе ОВЕН при передаче времени данные имеют следующий формат: «**год:месяц:день:час:минута:секунда:миллисекунда»**.

Особенности переменных типа Unsigned

Переменные типа «Unsigned» позволяют пользователю передавать любые данные в любом произвольном (в том числе собственном специализированном) формате. В переменную типа «Unsigned» командой **Добавить подэлемент (Insert element)** контекстного меню вставляются

переменные длиной 1, 2 или 4 байта. Допускается вставить не более четырех переменных суммарной длинной не более 16 байт.

Параметры переменных протокола ОВЕН

Параметры переменных протокола ОВЕН (общие для всех типов переменных):

- **Name** – имя элемента, см. раздел 10.6.1;
- **Address Length (Длина адреса устройства)** – значения выбираются из списка **«8 bit»** и **«11 bit»**, значение по умолчанию – **«8 bit»**;
- **Address (Адрес устройства)** – диапазон значений от 0 до 255 или от 0 до 2048, в зависимости от размера адреса, значение по умолчанию – 0;
- **Hash name (Сетевое имя переменной)** – сетевое имя переменной. Имена переменных ведомых приборов указываются в руководствах по эксплуатации этих приборов. Вводимое имя преобразуется в ПЛК в hash-код, который используется при обмене по сети RS-485;
- **Index (Индекс прибора)** – индекс прибора. В параметре **«Use a index? (Использовать индекс?)»** задают использование индекса. В совокупности параметры применяются для управления конфигурационными параметрами ПЛК, определяют наличие линейного индекса у параметра и задают значение индекса. Диапазон значений от 0 до 65535, значение по умолчанию – 0;
- **Use a index? (Использовать индекс?)** – значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**;
- **Polling time, ms (Период опроса устройства, мс)** – диапазон значений от 20 до 5000, значение по умолчанию – 100;

  ПРИМЕЧАНИЕ
  Следует учитывать физические ограничения сети: скорость информационного обмена в сети ограничена, и если задается большое количество переменных, значения которых часто запрашиваются, то информация будет поступать к прибору с запаздыванием. Поэтому требуется заранее просчитать пропускную способность сети, и, соответственно, уменьшить частоту опроса или опрашивать по разным линиям. При работе в режимах «Value change (По изменению значения переменных)» и «By Command (По команде)» (режим задается значением параметра «Work mode (Режим работы)», см. ниже), не следует задавать значение параметра «Polling time» слишком маленьким. По умолчанию его значение 100 мс. Если на реальном проекте будет замечено, что модуль «Owen (Master)» при загрузке программы или при выборе команды «Login» формирует лишние пакеты и/или запросы, то значение параметра следует увеличить (до 200, 300 и более мс), вплоть до прекращения появления лишних пакетов.

- **Work mode (Режим работы)** – режим работы модуля «Owen (Master)» при опросе внешних устройств (для переменных различных типов список допустимых значений различен, см. примечание ниже):
  - **Polling time (По времени)** – контролируемые устройства опрашиваются с периодичностью, заданной в параметре **«Polling time (Период опроса устройства)»**;
  - **Value change (По изменению значения переменных)** – модуль «Owen (Master)» генерирует запрос устройству при изменении значений выходных переменных модуля;
  - **Both (Оба варианта»** – опрос производится с временным интервалом, заданным в параметре «Polling time» и тогда, когда изменяются значения выходных переменных;
  - **By Command (По команде)** – производится однократная посылка запроса, когда в командный канал («Command») переменной, имеющей такой канал, записывается значение **0x00FF**.

    ПРИМЕЧАНИЕ
    Для считываемых переменных (тип «Listen») доступен только режим «Polling time (По времени)».
    Для записываемых переменных (тип «Write») доступны режимы «Polling time» (По времени), «Value change» (По изменению значения переменных) и «Both» (По времени и по изменению значения переменных).
    Для переменных с командным каналом (тип «Command») доступен режим «Command (По команде)». В этом режиме управление осуществляется следующим образом: первая посылка значения **0x00FF** в командный канал включает функционирование этой переменной, повторная посылка значения **0x00FF** инициирует проведение опроса. Аналогично опрос инициируется для переменных с командным каналом при работе в других режимах. При посылке в командный канал значения **0x00FE** переменная выключается из цикла опроса мастера.

- **Repiat Counter** – число повторов запроса при ошибке связи;

- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Для переменных типов «Float variable» (число с плавающей точкой), «Float variable + time» (число с плавающей точкой с модификатором времени) всех модификаций («Listen» – предназначенные для чтения, «Write» – предназначенные для записи, «Command» – имеющие дополнительный командный управляющий канал) – задаются параметры «Float Type» (Тип числа с плавающей точкой), который уточняет вид переменной типа Float и «Precision», который задает точность (определяет положение десятичной точки):

- **Float Type (тип числа с плавающей точкой)** – значение выбирается пользователем из списка (значение по умолчанию – «Float»):
  - **Float** – число с плавающей точкой в формате IEEE, обычно используемое в программировании, в CODESYS называется Real, имеет длину 4 байта;
  - **Float-Pic** – переменная размером в 3 байта, и один байт из мантиссы удаляется, т. е. число с меньшей точностью, но и размер переменной (количество байт) меньше;
  - **Fix point binary** – число с фиксированной точкой в двоичном виде; например, число 3 будет записано в виде «0011». Положение десятичной точки для параметров с фиксированной точкой задается параметром «Precision (Точность)»;
  - **Fix point BCD** – число с фиксированной точкой в двоично-десятичном виде (двоично-десятичный код – форма записи целых чисел, когда каждый десятичный разряд числа записывается в виде его четырехбитного двоичного кода. Например, число 311 будет записано в виде «0011 0001 0001». Положение десятичной точки для параметров с фиксированной точкой задается параметром «Precision» (Точность).
- **Precision (Точность)** – точность (определяет положение десятичной точки) для параметров с фиксированной точкой: если выбирается значение точности «2», то число «10,12» так и будет передано. При значении «1» – «10,1», при значении «3» – «10,120». Диапазон значений от 0 до 7, значение по умолчанию – 2.

Отличие переменной **Float** с модификатором времени состоит в том, что при тех же параметрах в поле данных, кроме собственно значения, присутствует еще и время (в сотых долях секунды).

Примеры задания параметров для различных случаев применения ПЛК представлены в приложении Примеры настройки опроса переменных по протоколу ОВЕН .

10.7.6 Модуль «Owen (Slave)»

Модуль «OWEN (Slave)» используется для работы контроллера в сети в подчиненном режиме (slave), т. е. для ответа на запросы устройства, работающего в режиме Master. Во время установки модуля «OWEN (Slave)» следует выбрать коммуникационный интерфейс для обмена данными с другими устройствами, добавить и настроить требуемые переменные. Модуль «OWEN (Slave)» обеспечивает обмен информацией по протоколу ОВЕН (протокол предназначен для описания процесса обмена информацией между приборами компании «ОВЕН» и между приборами и ПК на базе сети RS-485).

ПРИМЕЧАНИЕ
При случайном отключении питания в процессе работы ПЛК последние (текущие) значения переменных сохраняются в энергонезависимой памяти и восстанавливаются при возобновлении работы прибора.

Модуль «OWEN (Slave)» – составной и имеет в своем составе подмодуль «**OWEN (FIX)»** (см. раздел 10.7.6.1).

Параметры модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Slave Name (Имя прибора)** – имя ПЛК в сети протокола ОВЕН. Значение по умолчанию – **«max 8 sym»** (т. е. максимально 8 символов);
- **Address Length (Длина адреса устройства)** – длина адреса в битах. Приборы разработки компании «ОВЕН» поддерживают два варианта адресов – 8-ми и 11-ти битовые (задается тот же вариант адреса, что выставлен на управляющей стороне: значение длины адреса Master и Slave-устройств должны совпадать). Значения выбираются из списка **«8 bit»** и **«11 bit»**, значение по умолчанию – **«8 bit»**;
- **Address (Адрес устройства)** – адрес прибора, по которому посылаются запросы. Параметр имеет значения в диапазоне от 0 до 2048 значение по умолчанию – 1;

- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

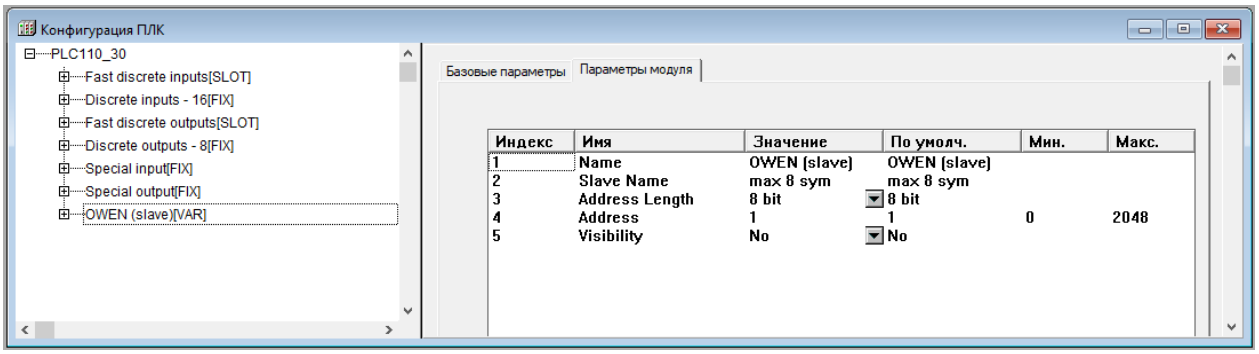


Рисунок 10.39 – Параметры модуля «Owen (Slave)»

10.7.6.1 Подмодуль «OWEN (FIX)»

Настройка коммуникационных интерфейсов

Во время добавления модуля «OWEN (Slave)» в конфигурацию ПЛК в состав модуля уже подключен подмодуль «OWEN (FIX)», к которому, в свою очередь, подключается коммуникационный интерфейс (см. рисунок 10.40).

В ПЛК предусмотрена возможность обмена данными по следующим интерфейсам:

- RS-232;
- RS-485;
- TCP (Ethernet).

Для работы с разными коммуникационными интерфейсами в ПЛК предусмотрены соответствующие подмодули (подэлементы). Подэлементы подключаются выбором команды **Добавить подэлемент → <Имя подэлемента>** контекстного меню (см. рисунок 10.41).

Во время работы ПЛК в режиме **«Ведомый (slave)»** возможно использование нескольких разных портов, т. е. опрос может вестись по разным интерфейсам. Таким образом, подключая несколько разных портов, можно один модуль соединить с разными Мастерами по разным физическим линиям (и интерфейсам).

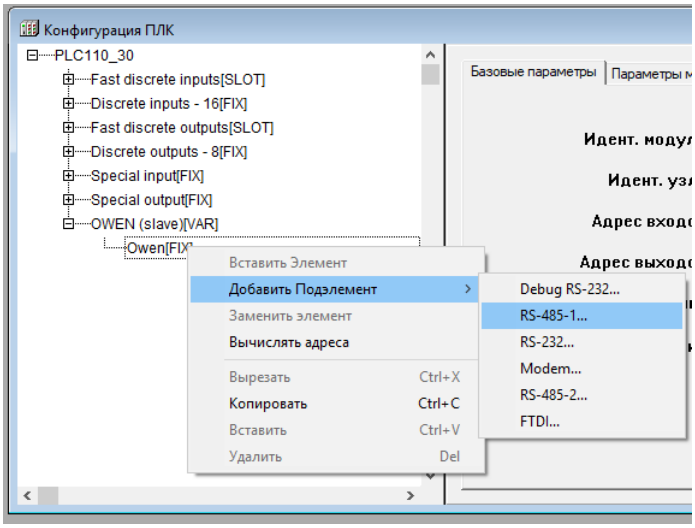


Рисунок 10.40 – Выбор интерфейса связи

Количество подключаемых портов ограничено конструкцией ПЛК.

Пример

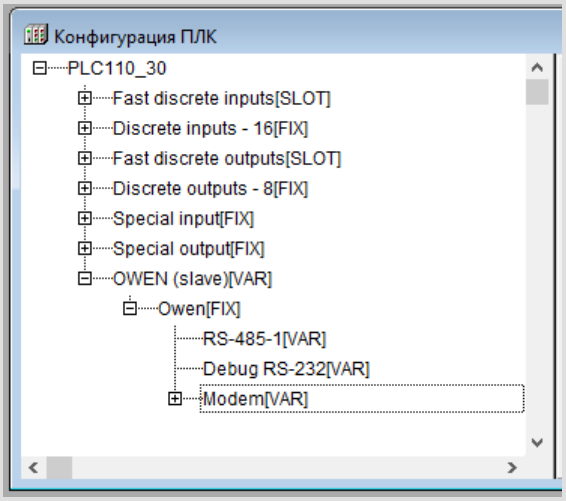


Рисунок 10.41 – Подключение нескольких портов

Настройка входов/выходов подмодуля

После задания значений параметров подмодуля «OWEN (Slave)» к нему следует подключить каналы входа/выхода.

Каналы добавляются выбором команды **Добавить подэлемент** → **<Наименование подэлемента>** контекстного меню строки «OWEN (Slave)».

В модуле «Owen (Slave)» могут быть использованы только переменные для чтения (обозначены «**Listen**»), использующие следующие типы данных:

- **Float variable** – число с плавающей точкой;
- **Float variable + time** – число с плавающей точкой с модификатором времени;
- **Unsigned variable** – целочисленная переменная;
- **Unsigned variable + time** – целочисленная переменная с модификатором времени;
- **String variable** – строковая переменная, максимальная длина – 15 символов, в соответствии со стандартом протокола ОВЕН;
- **Time variable** – позволяет передать время; в протоколе ОВЕН при передаче времени данные имеют следующий формат: **год:месяц:день:час:минута:секунда:миллисекунда**.

Переменные, которыми будет обмениваться ПЛК по протоколу ОВЕН, выбираются командой **Добавить Подэлемент (Append Subelements)** контекстного меню строки «OWEN (slave)».

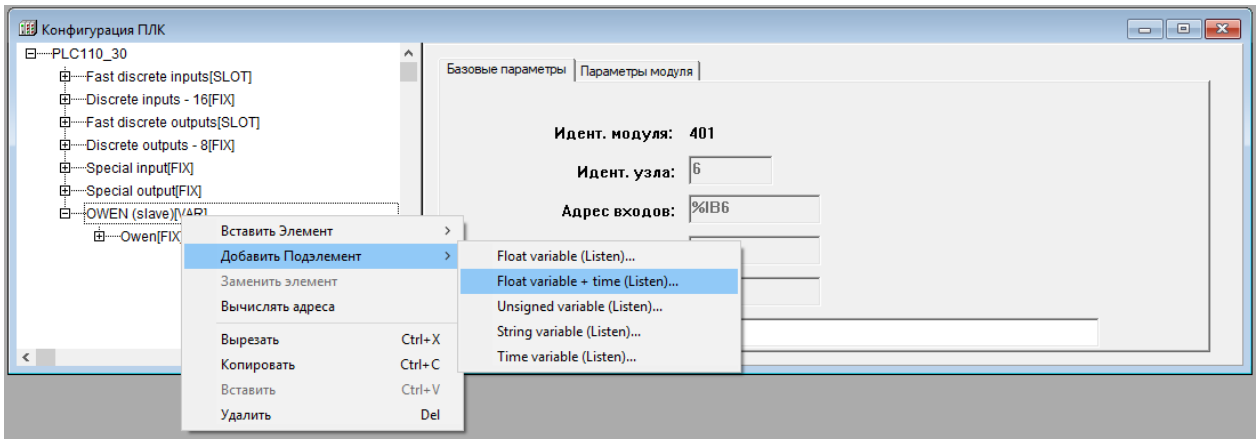


Рисунок 10.42 – Выбор переменных для обмена ПЛК по протоколу ОВЕН

Параметры переменных протокола ОВЕН

Параметры переменных протокола ОВЕН, общие для всех типов переменных:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Address Length (длина адреса устройства)** – значения выбираются из списка **«8 bit»** и **«11 bit»**, значение по умолчанию – **«8 bit»**;

- **Address (адрес устройства)** – диапазон значений от 0 до 255 или от 0 до 2048, в зависимости от размера адреса, значение по умолчанию – 0;
- **Hash name (сетевое имя переменной)** – сетевое имя переменной. Имена переменных ведомых приборов указываются в руководствах по эксплуатации этих приборов. Вводимое имя преобразуется в ПЛК в hash-код, который используется при обмене по сети RS-485;
- **Index (индекс прибора)** – индекс прибора, в параметре **«Use a index? (использовать индекс?)»** задают использование индекса. В совокупности параметры применяются для управления конфигурационными параметрами ПЛК, определяют наличие линейного индекса у параметра и задают значение индекса. Диапазон значений от 0 до 65535, значение по умолчанию – 0;
- **Use a index? (использовать индекс?)** – значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**;
- **Polling time, ms (период опроса устройства, мс)** – диапазон значений от 20 до 5000, значение по умолчанию – 100;

  ПРИМЕЧАНИЕ
  Следует учитывать физические ограничения, накладываемые характеристиками сети: скорость информационного обмена в сети ограничена, и если задается большое количество переменных, значения которых часто запрашиваются, то информация будет поступать к прибору с запаздыванием. Поэтому требуется заранее просчитать пропускную способность сети, и, соответственно, уменьшить частоту опроса, или производить опрос по разным линиям и/или др. При работе в режимах «Value change» (По изменению значения переменных) и «By Command (По команде)» (режим задается значением параметра «Work mode» (Режим работы), см. ниже), нельзя задавать значение параметра «Polling time» слишком маленьким. По умолчанию его значение 100 мс. Если на реальном проекте будет замечено, что модуль «Owen (Master)» при загрузке программы или при выборе команды «Login» формирует лишние пакеты и/или запросы, то значение параметра следует увеличить (до 200, 300 и более мс), вплоть до прекращения появления лишних пакетов.

- **Work mode (режим работы)** – режим работы модуля «Owen (Master)» при опросе внешних устройств (для переменных различных типов список допустимых значений различен, см. примечание ниже):
  - **Polling time (по времени)** – контролируемые устройства опрашиваются с периодичностью, заданной в параметре **«Polling time» (Период опроса устройства)**;
  - **Value change (по изменению значения переменных)** – модуль «Owen (Master)» генерирует запрос устройству при изменении значений выходных переменных модуля;
  - **Both (оба варианта»** – опрос производится с временным интервалом, заданным в параметре «Polling time» и тогда, когда изменяются значения выходных переменных;
  - **By Command (по команде)** – производится однократная посылка запроса, когда в командный канал («Command») переменной, имеющей такой канал, записывается значение **0x00FF**.

    ПРИМЕЧАНИЕ
    Для считываемых переменных (тип «Listen») доступен только режим «Polling time (По времени)».
    Для записываемых переменных (тип «Write») доступны режимы «Polling time (По времени)», «Value change (По изменению значения переменных)» и «Both (По времени и по изменению значения переменных)».
    Для переменных с командным каналом (тип «Command») доступен режим «Command (По команде)». В этом режиме управление осуществляется следующим образом: первая посылка значения **0x00FF** в командный канал включает функционирование этой переменной, повторная посылка значения **0x00FF** инициирует проведение опроса. Аналогично опрос инициируется для переменных с командным каналом при работе в других режимах. При посылке в командный канал значения **0x00FE** переменная выключается из цикла опроса мастера.
- **Repiat Counter** – число повторов запроса при ошибке связи;
- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Для переменных типов «Float variable» (число с плавающей точкой), «Float variable + time» (число с плавающей точкой с модификатором времени) всех модификаций («Listen» – предназначенные для чтения, «Write» – предназначенные для записи, «Command» – имеющие дополнительный командный управляющий канал) – задаются параметры «Float Type (Тип числа с плавающей

точкой)», который уточняет вид переменной типа Float и «Precision», который задает точность (определяет положение десятичной точки):

- **Float Type (тип числа с плавающей точкой)** – значение выбирается пользователем из списка (значение по умолчанию – «Float»):
  - **Float** – число с плавающей точкой в формате IEEE, в CODESYS называется Real, имеет длину 4 байта;
  - **Float-Pic** – переменная размером в 3 байта, и один байт из мантиссы удаляется, т. е. число с меньшей точностью, но и размер переменной (количество байт) меньше;
  - **Fix point binary** – число с фиксированной точкой в двоичном виде. Например, число 3 будет записано в виде «0011». Положение десятичной точки для параметров с фиксированной точкой задается параметром **Precision (Точность)**;
  - **Fix point BCD** – число с фиксированной точкой в двоично-десятичном виде (двоично-десятичный код – форма записи целых чисел, когда каждый десятичный разряд числа записывается в виде его четырехбитного двоичного кода. Например, число 311 будет записано в виде «0011 0001 0001». Положение десятичной точки для параметров с фиксированной точкой задается параметром **Precision (Точность)**.
- **Precision (Точность)** – точность (определяет положение десятичной точки) для параметров с фиксированной точкой: если выбирается значение точности «2», то число «10,12» так и будет передано. При значении «1» – «10,1», при значении «3» – «10,120». Диапазон значений от 0 до 7, значение по умолчанию – 2.

Отличие переменной **Float** с модификатором времени состоит в том, что, при тех же параметрах, в поле данных, кроме собственно значения, присутствует еще и время (в сотых долях секунды).

Примеры задания параметров для различных случаев применения ПЛК представлены в приложении Примеры настройки опроса переменных по протоколу ОВЕН .

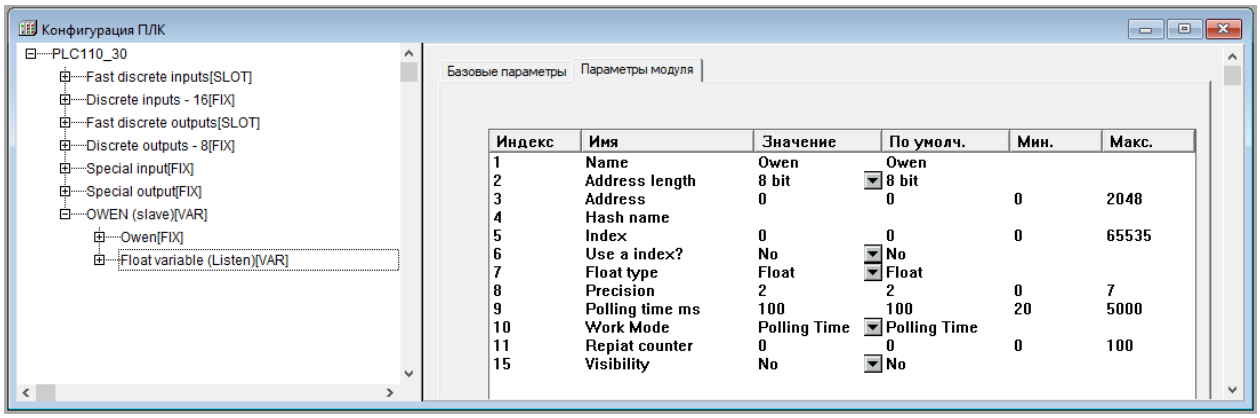


Рисунок 10.43 – Параметры переменных протокола ОВЕН

10.7.7 Модуль «Owen (Spy)»

Модуль «OWEN (Spy)» используется для мониторинга информационных обменов в сети, в которую включен ПЛК. Модуль не отвечает на запросы Мастера сети, а только прослушивает обмен данными в сети RS-485 и может быть настроен таким образом, что в ходе опроса Мастером сети какого-либо устройства, ответ этого устройства прослушивается модулем «OWEN (Spy)» и записывается им во встроенную переменную. Этот механизм может быть использован, если ПЛК требуется интегрировать в существующую сеть: получать из нее данные для последующей обработки и выполнения заданных действий. Использование модуля «OWEN (Spy)» позволяет не останавливать работу системы в ходе интеграции: ПЛК прослушивает требуемые данные и выполняет запрограммированные действия.

Во время установки модуля «OWEN (Spy)» следует выбрать коммуникационный интерфейс для обмена данными с другими устройствами, добавить и настроить требуемые переменные. Модуль «OWEN (Spy)» обеспечивает обмен информацией по протоколу ОВЕН (протокол предназначен для описания процесса обмена информацией между приборами компании «ОВЕН» и между приборами и ПЭВМ на базе сети RS-485).

ПРИМЕЧАНИЕ
При случайном отключении питания в процессе работы ПЛК последние (текущие) значения переменных сохраняются в энергонезависимой памяти и восстанавливаются при возобновлении работы прибора.

Модуль «OWEN (Spy)» – составной и имеет в своем составе подмодуль «OWEN [FIX]» (см. раздел 10.7.7.1).

Модуль имеет единственный параметр «Visibility (Видимость)», который задает видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

10.7.7.1 Подмодуль «OWEN (FIX)»

Настройка коммуникационных интерфейсов

При добавлении модуля «OWEN (Spy)» в конфигурацию ПЛК, в состав модуля уже подключен подмодуль «OWEN (FIX)», к которому подключается коммуникационный интерфейс.

В ПЛК предусмотрена возможность обмена данными по интерфейсам **RS-232** и **RS-485**.

Для работы с разными коммуникационными интерфейсами в ПЛК предусмотрены соответствующие подмодули (подэлементы). Подэлементы подключаются выбором команды **Добавить подэлемент → <Имя подэлемента>** контекстного меню (см. рисунок 10.44). Допустимо использование нескольких разных портов, т. е. опрос может вестись по разным интерфейсам.

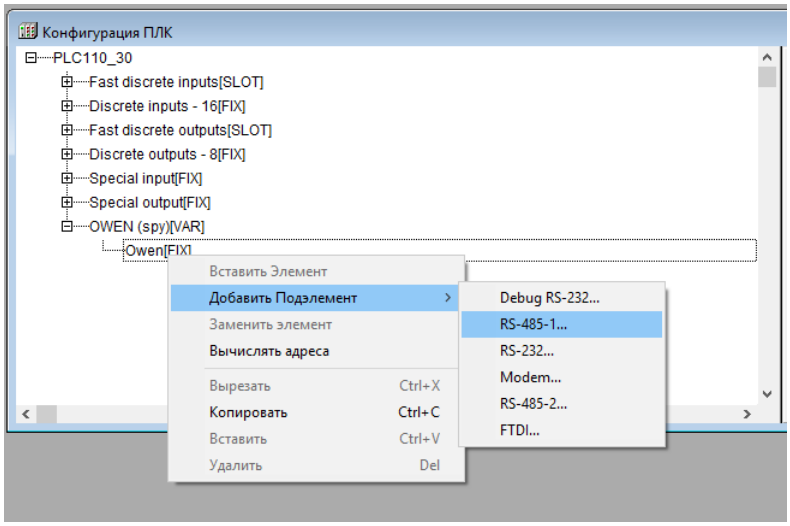


Рисунок 10.44 – Выбор интерфейса связи

Количество подключаемых портов ограничено конструкцией ПЛК.

Пример

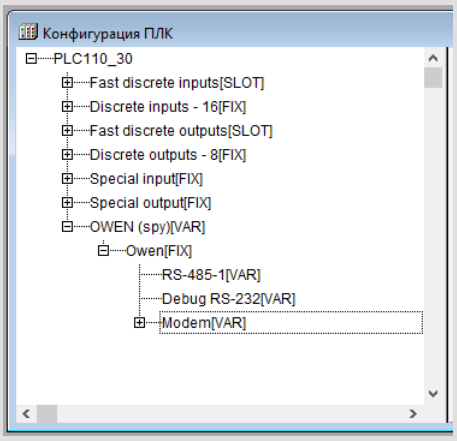


Рисунок 10.45 – Подключение нескольких портов

Настройка входов и выходов модуля

После задания значений параметров подмодуля «OWEN (Spy)» к нему следует подключить каналы, задающие входные параметры (параметры, значение которых модуль запрашивает у сети) модуля.

В модуле «Owen (Spy)» могут быть использованы только переменные для чтения (обозначены «Listen»), использующие следующие типы данных:

- **Float variable** – число с плавающей точкой;
- **Float variable + time** – число с плавающей точкой с модификатором времени;
- **Unsigned variable** – целочисленная переменная;
- **Unsigned variable + time** – целочисленная переменная с модификатором времени;
- **String variable** – строковая переменная, максимальная длина – 15 символов, в соответствии со стандартом протокола ОВЕН;
- **Time variable** – позволяет передать время, в протоколе ОВЕН при передаче времени данные имеют следующий формат: «год:месяц:день:час:минута:секунда:миллисекунда».

Переменные, которыми будет обмениваться ПЛК по протоколу ОВЕН, выбираются командой **Добавить Подэлемент (Append Subelements)** контекстного меню строки «OWEN (Spy)».

При добавлении канала в «OWEN (Spy)» в дерево конфигурации добавляется подкаталог, содержащий внутри себя канал, в котором отображаются полученное/передаваемое по сети значение (см. рисунок 10.46).

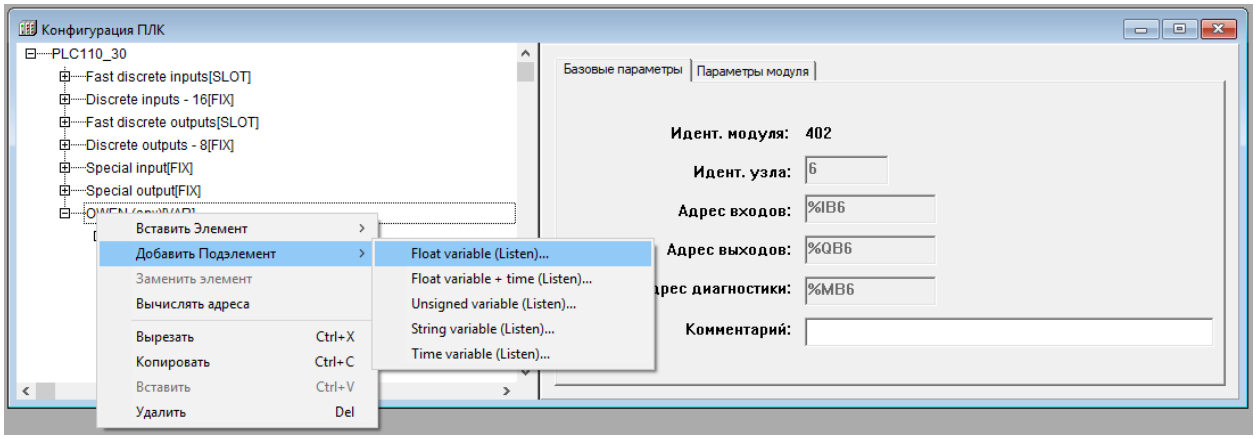


Рисунок 10.46 – Добавление каналов ввода/вывода

Параметры переменных протокола ОВЕН

Параметры переменных протокола ОВЕН, общие для всех типов переменных:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Address Length (Длина адреса устройства)** – размер адреса ведомого устройства в битах, ответ которого необходимо прослушать, значения выбираются из списка **«8 bit»** и **«11 bit»**, значение по умолчанию – **«8 bit»**;
- **Address (Адрес устройства)** – адрес ведомого устройства, ответ которого требуется прослушать, диапазон значений от 0 до 255 или от 0 до 2048, в зависимости от размера адреса, значение по умолчанию – 0;
- **Hash name (Сетевое имя переменной)** – сетевое имя переменной ведомого устройства, опрашиваемого Мастером сети. Имена переменных ведомых приборов указываются в руководствах по эксплуатации этих приборов. Вводимое имя преобразуется в ПЛК в hash-код, который используется при обмене по сети RS-485;
- **Index (Индекс прибора)** – индекс прибора, в параметре **«Use a index? (Использовать индекс?)»** задают использование индекса. В совокупности параметры применяются для управления конфигурационными параметрами ПЛК, определяют наличие линейного индекса у параметра и задают значение индекса. Диапазон значений от 0 до 65535, значение по умолчанию – 0;
- **Use a index? (Использовать индекс?)** – значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**;
- **Polling time, ms (Период опроса устройства, мс)** – модулем «OWEN (Spy)» не используется;
- **Work mode (Режим работы)** – режим работы модуля при опросе внешних устройств: «Polling time (По времени)»;
- **Repiat Counter** – число повторов запроса при ошибке связи;
- **Visibility (Видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Для переменных типов «Float variable» (число с плавающей точкой), «Float variable + time» (число с плавающей точкой с модификатором времени) всех модификаций («Listen» – предназначенные для чтения, «Write» – предназначенные для записи, «Command» – имеющие дополнительный

командный управляющий канал) – задаются параметры «Float Type» (тип числа с плавающей точкой), который уточняет вид переменной типа Float и «Precision», который задает точность (определяет положение десятичной точки):

- **Float Type (тип числа с плавающей точкой)** – значение выбирается пользователем из списка (значение по умолчанию – «Float»):
  - **Float** – число с плавающей точкой в формате IEEE, в CODESYS называется Real, имеет длину 4 байта;
  - **Float-Pic** – переменная размером в 3 байта, и один байт из мантиссы удаляется, т. е. число с меньшей точностью, но и размер переменной (количество байт) меньше;
  - **Fix point binary** – число с фиксированной точкой в двоичном виде. Например, число 3 будет записано в виде «0011». Положение десятичной точки для параметров с фиксированной точкой задается параметром **Precision (Точность)**;
  - **Fix point BCD** – число с фиксированной точкой в двоично-десятичном виде (двоично-десятичный код – форма записи целых чисел, когда каждый десятичный разряд числа записывается в виде его четырехбитного двоичного кода. Например, число 311 будет записано в виде «0011 0001 0001». Положение десятичной точки для параметров с фиксированной точкой задается параметром **Precision (Точность)**.
- **Precision (Точность)** – точность (определяет положение десятичной точки) для параметров с фиксированной точкой: если выбирается значение точности «2», то число «10,12» так и будет передано. При значении «1» – «10,1», при значении «3» – «10,120». Диапазон значений от 0 до 7, значение по умолчанию – 2.

Отличие переменной **Float** с модификатором времени состоит в том, что, при тех же параметрах, в поле данных, кроме собственно значения, присутствует еще и время (в сотых долях секунды).

Примеры задания параметров для различных случаев применения ПЛК представлены в приложении Примеры настройки опроса переменных по протоколу ОВЕН .

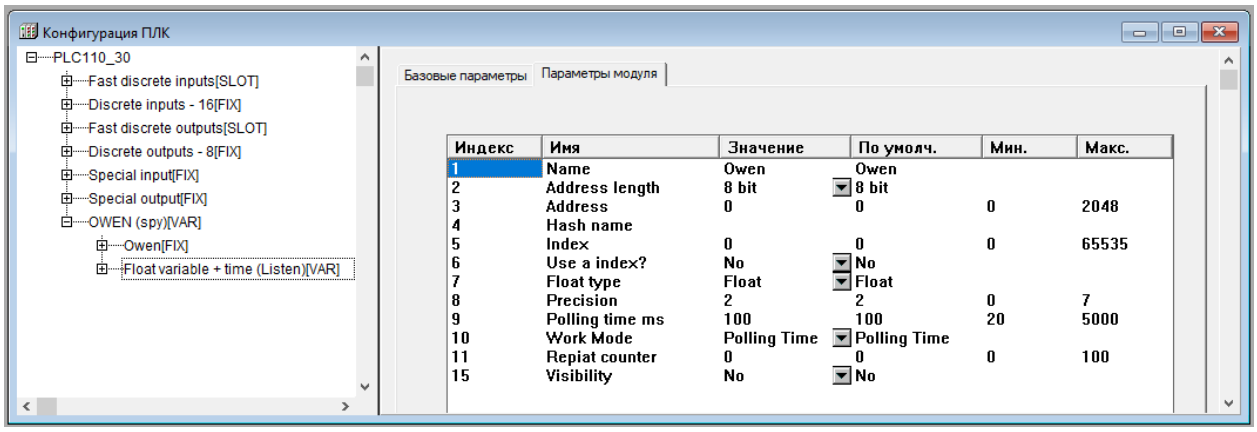


Рисунок 10.47 – Параметры переменных протокола ОВЕН

10.7.8 Модуль статистики Statistic

Модуль статистики (Statistic) предназначен для выдачи в пользовательскую программу информационные данные о функционировании ПЛК.

Модуль не имеет параметров и содержит следующие каналы (см. рисунок 10.48):

- **Last cycle time in mks (Значение последнего цикла работы ПЛК в мкс)** – позволяет пользователю оценить объем вычислительных ресурсов, который требуется для работы написанной им программы. Если цикл оказывается больше, заданного в параметрах работы ПЛК параметра **MinCycleLength**, то это означает, что пользовательская программа слишком требовательна к ресурсам и параметр **MinCycleLength** желательно увеличить, чтобы циклы не перекрывались;
- **Temp inside PLC (Температура внутри ПЛК)** – температура, замеренная датчиком внутри корпуса ПЛК (у разных моделей ПЛК температура может измеряться на разных платах, определяется интенсивностью нагрева конкретных плат). Характеристика косвенно свидетельствует о рабочем состоянии ПЛК;
- **Power status (Состояние питания)**– логическая переменная, имеющая значение TRUE при наличии питания от сети и значение FALSE в случае отсутствия питания или аварии по питанию;

ПРИМЕЧАНИЕ
Сейчас переменная **Power status** всегда имеет значение **TRUE** и не настраивается. Эта переменная оставлена в таргет-файлах для обратной совместимости с уже существующими пользовательскими проектами.

- **CPU is overloaded, optimize your program or increase PLC cycle (центральный процессор перегружен, оптимизируйте вашу программу или увеличьте время цикла ПЛК)** – логическая переменная, имеющая значение TRUE, если цикл оказывается больше, заданного в параметрах работы ПЛК параметра **MinCycleLength**, в случае выполнения программы в рамках заданного времени цикла переменная имеет значение FALSE;
- **Free processor resource mks in 1 cycle (свободное процессорное время за один цикл в мкс)** – значение этой переменной соответствует времени в цикле исполнения программы, не занятому исполнением программы.

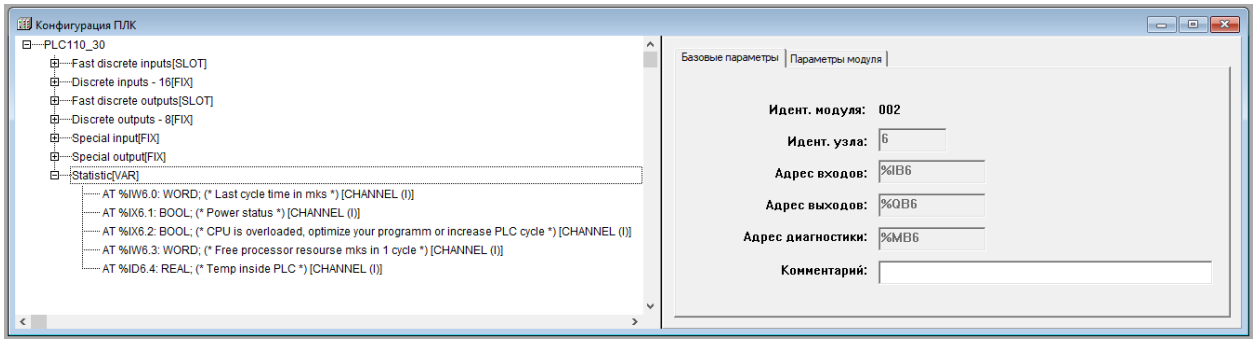


Рисунок 10.48 – Каналы модуля статистики (Statistic)

10.7.9 Модуль «Universal network module» (Универсальный сетевой модуль)

Универсальный сетевой модуль предназначен для организации универсального коммуникационного интерфейса ПЛК, выполняющего прием/передачу последовательности байт через встроенные порты контроллера (RS-232/RS-485/Ethernet). Подключение модуля резервирует область памяти ввода-вывода ПЛК, с которой будут выполняться функции, включенные в специализированную библиотеку дополнительных программных модулей UNM.lib (Universal Network Module).

Особенностью данной библиотеки (и, соответственно, модуля конфигурации ПЛК) является возможность работать одновременно с протоколами Modbus, DCON и ОВЕН на одном физическом интерфейсе.

Одновременная работа с разными протоколами позволяет создать модуль опроса устройства стандартными командами и в то же время выдавать в интерфейс и получать из интерфейса в нужное время произвольную последовательность байт. Например, если к порту RS-232 подключен модем, то до начала работы стандартного модуля опроса пользовательская программа позволит установить связь с удаленным устройством, используя «AT»-последовательности. После этого начнет работать модуль опроса устройства через стандартный протокол.

Описание функций библиотеки UNM. lib (Universal Network Module) см. в документе *Библиотека UNM* на странице CODESYS V2 на сайте owen.ru.

Модуль имеет один канал (Debug RS-232[SLOT]) и один параметр **Видимость (Visibility)**, задающий видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Во время добавления модуля «Universal network module» (Универсальный сетевой модуль) в конфигурацию ПЛК в состав модуля уже подключен подмодуль «Debug RS-232[SLOT]», к которому подключается коммуникационный интерфейс (см. рисунок 10.49).

В ПЛК предусмотрена возможность обмена данными по интерфейсам:

- RS-232;
- RS-485;
- TCP (Ethernet).

Для работы с разными коммуникационными интерфейсами в ПЛК предусмотрены соответствующие подмодули (подэлементы). Подэлементы подключаются выбором команды **Заменить подэлемент → <Имя подэлемента>** контекстного меню.

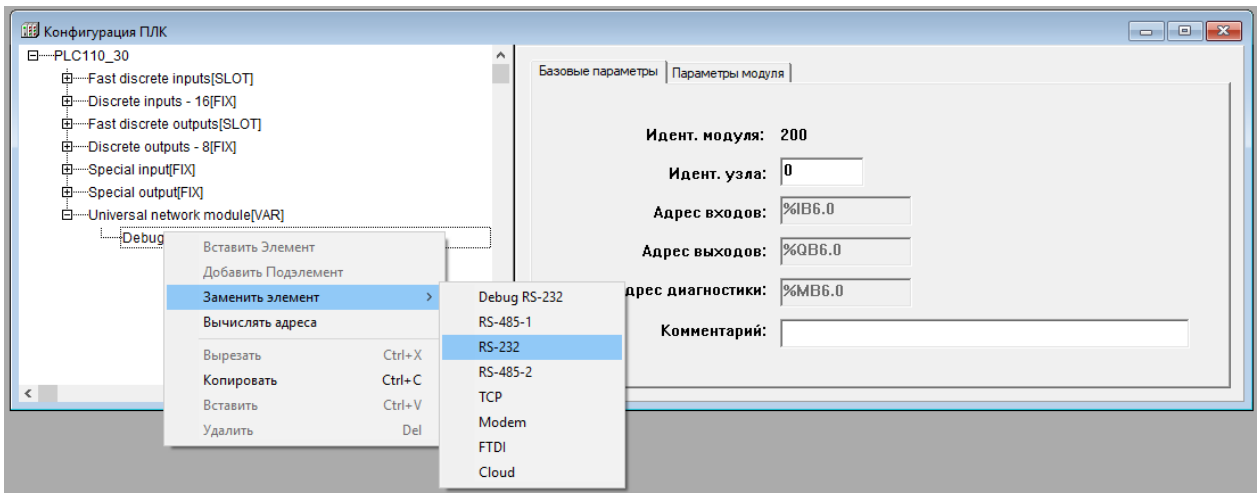


Рисунок 10.49 – Параметры и контекстное меню замены подэлементов Универсального сетевого модуля (Universal network module)

10.7.10 Модуль «Archiver» (Архиватор)

Модуль «Archiver» (Архиватор) используется для архивирования требуемых данных. Архивируемые данные могут храниться на Flash-диске ПЛК и извлекаться оттуда при необходимости, или выводиться через коммуникационный интерфейс (например, данные могут быть распечатаны на принтере, подключенном к ПЛК через последовательный интерфейс). К модулю «Archiver» (Архиватор) может быть добавлен подмодуль архивации информации в файл «File Output» (см. раздел 10.7.10.1).

Перечень архивируемых переменных (внесение переменных в список для последующего архивирования) создается вызовом команды **Добавить подэлемент (Append Subelement)** контекстного меню строки «Archiver».

В список могут быть добавлены переменные следующих типов:

- 8-битная;
- 16-битная;
- 32-битная;
- число с плавающей точкой («Float»);
- текстовая строка («String») – максимум 15 символов + завершающий нуль.

После добавления переменной любого типа в ее параметрах следует задать имя архивируемой переменной – **Variable Name**. Этим же именем переменная будет именоваться в архивном файле (см. рисунок 10.51).

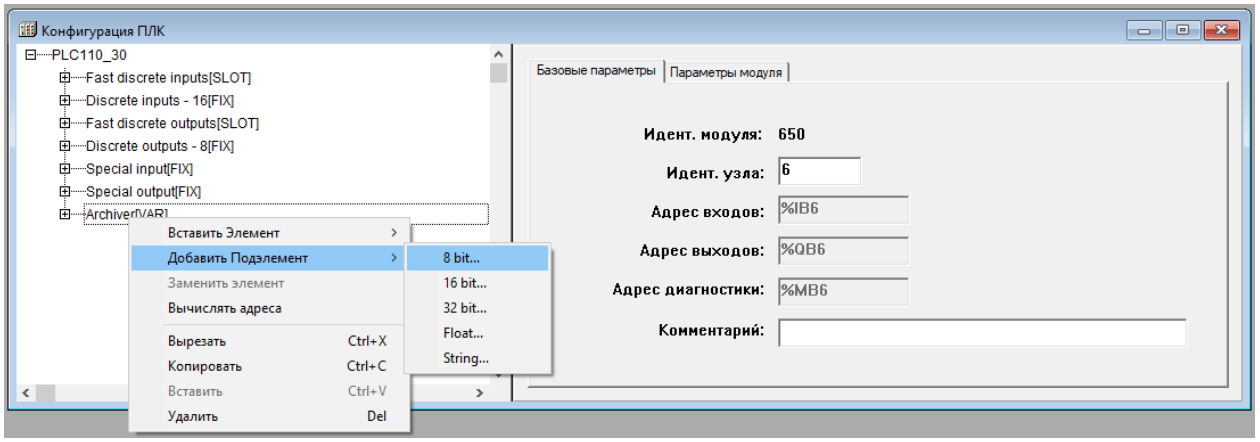


Рисунок 10.50 – Внесение переменных в список архивируемых

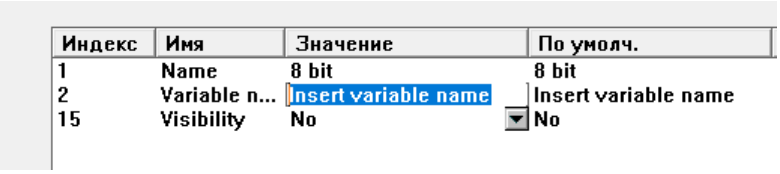


Рисунок 10.51 – Именование переменных в список архивируемых

Параметры передачи архивных данных (режим передачи, формат данных) задаются в параметрах модуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Archive Mode (Режим проведения архивации)** – режим архивации. Значения выбираются из списка:
  - **By timer (по таймеру)** – данные записываются в архив с заданным периодом архивации, значение по умолчанию;
  - **By change value (по изменению значений)** – если какая-то переменная, включенная в список архивации, меняет свое значение, то происходит ее архивация, причем только этой переменной. Изменения могут записываться не чаще, чем раз в пять секунд;
  - **By command (по команде)** – если в переменной **Status** модуля архивации записана специальная команда, то происходит старт архивации, либо ее остановка (**0x00FE – «стоп»**, **0x00FF – «старт»**).

    ПРИМЕЧАНИЕ
    Действия архиватора в ответ на команды сводятся к следующему:
    - ♦ во время загрузки проекта модуль архивации находится в режиме «Работает»;
    - ♦ повторная команда «Старт» приводит к немедленному акту архивации;
    - ♦ если архиватор остановлен, то команда «Старт» запустит его.
- **Type of archive (тип данных архивации)** – тип записи архивируемых данных. Значения выбираются из списка:
  - **ASCII only** – данные выдаются в текстовом виде, удобном для чтения пользователя, для печати и т. п. (значения по умолчанию);
  - **Mixed** – данные выдаются в смешанном виде: запись имеет заголовок архива с именами переменных, временные данные в удобном для чтения виде, а все архивируемые переменные записываются в бинарном виде.
- **Period of Archiving (период архивации, с)** – периодичность обновления данных архива при работе модуля в режиме **«по таймеру»**. Диапазон значений от 5 до 65535 секунд, значение по умолчанию – 60;
- **Archive Name (имя архива)** – имя архива, которое записывается в начале файла;
- **Comment (комментарий архива)** – текст комментария к архиву. Здесь может быть введена информация, позволяющая в будущем идентифицировать конкретный архив по дополнительным признакам;
- **Start time (время начала архивации)** – время старта архивации;
- **Stop time (время остановки архивации)** – время остановки архивации;

  ВНИМАНИЕ
  Параметры «Start time (Время начала архивации)» и «Stop time (Время остановки архивации)», задающие временные рамки процесса архивирования, независимы друг от друга, т. е. один или оба параметра могут быть не заданы. Для параметров определен формат, в котором они должны задаваться – **чч:мм:сс**, – с обязательным использованием полноформатного задания величин и разделителя «двоеточие». При неполном формате и/или использовании иного разделителя программа проигнорирует информацию, как ошибочную.
- **On Sunday (воскресенье)…On Saturday (суббота)** (всего семь параметров) – назначается день (дни) недели, когда будет производиться архивация. Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«yes»**;
- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

ВНИМАНИЕ
Между всеми условиями старта и остановки архивирования установлен приоритет . Главный приоритет имеет переменная **File Status**: если в ней записана команда **«стоп»**, то операция в любом случае прекратится, если **«старт»** – она будет выполнена минимум один раз. Следующим по приоритету идет день недели и, далее, время старта и время останова процедуры архивации.

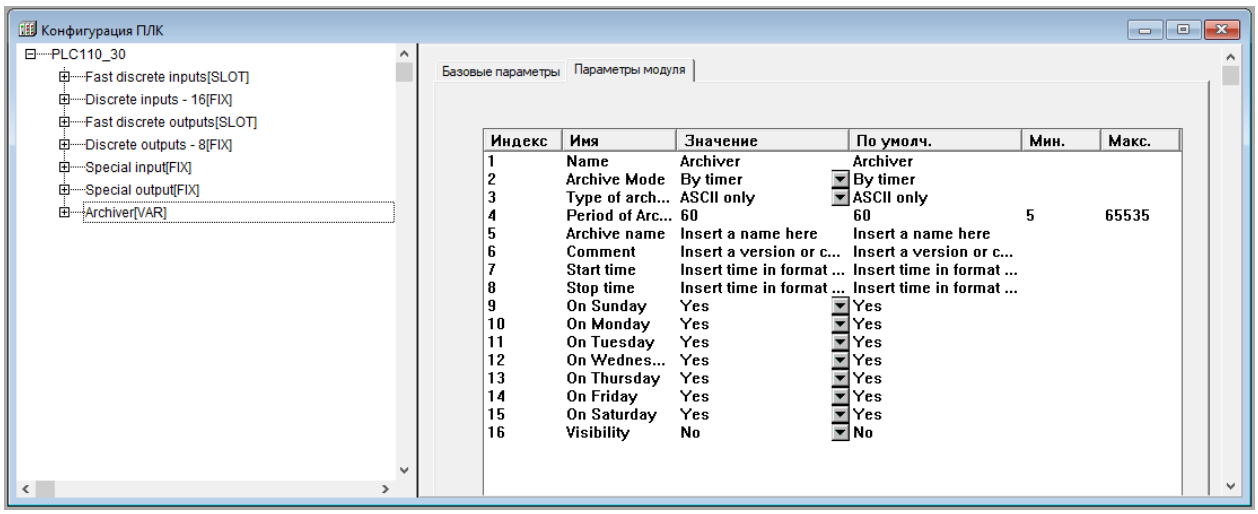


Рисунок 10.52 – Параметры модуля

Настройка коммуникационных интерфейсов

По умолчанию к модулю «Archiver (Архиватор)» подключен подмодуль интерфейсного порта, через который передаются архивные данные. Интерфейсный порт может быть заменен на требуемый выбором команды «Replace element» (Заменить элемент) контекстного меню (см. рисунок 10.53).

Настройка последовательных интерфейсов, интерфейса TCP, модема, и выгрузка архива в файл (порт «File Output») описана в разделе 10.7.10.1.

В модуле «Archiver (Архиватор)» имеется канал «Status», который отображает статус архива и коды возникающих ошибок. Коды ошибок ПЛК представлены в приложении Сообщения об ошибках в ПЛК .

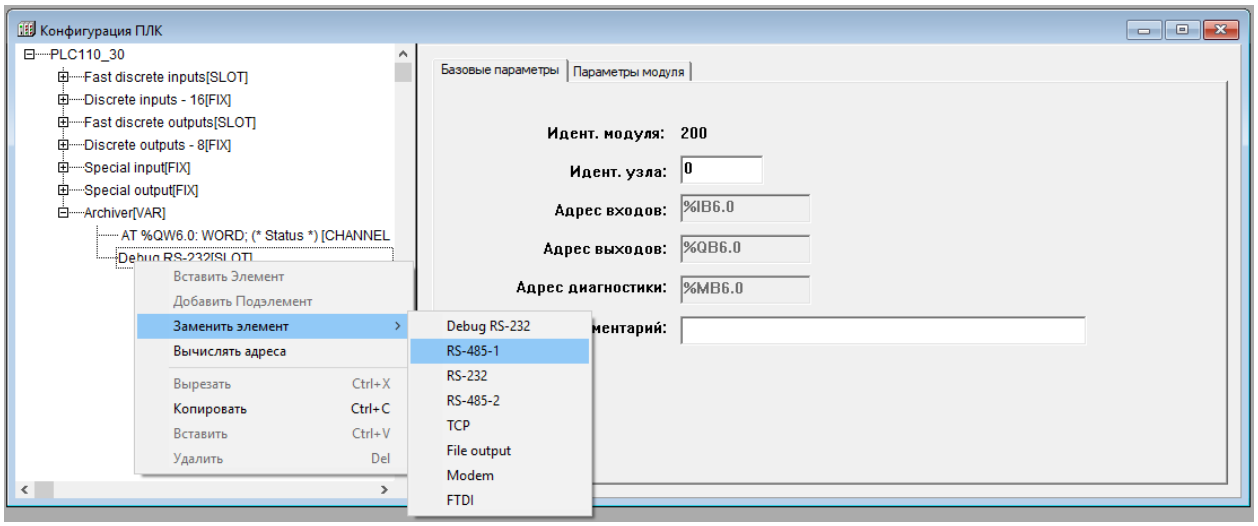


Рисунок 10.53 – Смена интерфейсного порта

10.7.10.1 Подмодуль архивации информации в файл («File output»)

Подмодуль интерфейсного порта «File output» задает параметры архивации информации в файл и является подчиненным подмодулем модуля «Архиватор».

Подмодуль «File output» имеет собственную переменную «File Status», в которой сохраняется информация о работе подмодуля. Коды ошибок ПЛК представлены в приложении Сообщения об ошибках в ПЛК .

Параметры подмодуля:

- **Name** – имя элемента, см. раздел 10.6.1;
- **File name (имя файла)** – имя файла, в который будет записываться архивная информация. Значение по умолчанию – **«File_name.log»**;
- **Mode (режим работы модуля)** – вариант архивации информации:
  - **Append to end (добавить в конец)** – информация добавляется в конец файла, и, как только файл переполняется, запись прекращается. Файл имеет ограничение по размеру (в байтах) или по количеству записей (задаваемому в параметре **Max file size**), значение по умолчанию;
  - **Rewrite on start (перезапись при старте)** – старый файл стирается во время старта ПЛК или загрузки новой конфигурации и начинается запись файла с самого начала;
  - **Rewrite on oversize (перезапись старого файла при превышении заданного размера)** – файл стирается при достижением им заданного размера, и запись начинается сначала;
  - **Shift Mode (режим сдвига)** – вариант работы, при котором, при достижении файлом заданного размера, вторая (более поздняя по времени записи) половина файла переносится в начало, запись продолжается, дописывается, т. е. остаются самые последние записи.
- **Type (тип)** – способ подсчета размера файла:
  - **Text (текстовый режим)** – по количеству записей, каждая запись заканчивается символом перевода каретки;
  - **Binary (цифровой или двоичный режим)** – по размеру файла в байтах.
- **Max file size (размер записи)** – ограничение размера записываемого файла, размер определяется в зависимости от типа (**Type**). Диапазон значений от 100 до 320000, значение по умолчанию – 500;
- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

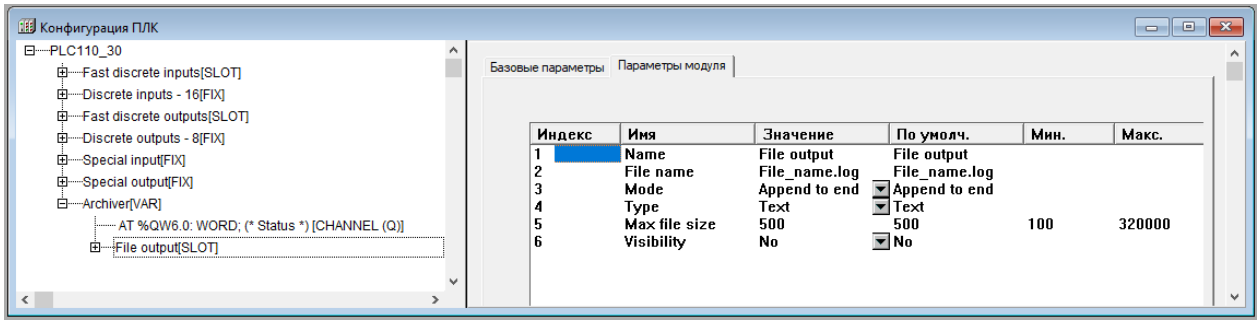


Рисунок 10.54 – Параметры подмодуля «File output»

10.7.11 Модуль «Extended settings» (Расширенные настройки)

Модуль «Расширенные настройки» (Extended settings) служит для управления режимом работы подтяжки линий RS-485 интерфейсов (Физический мастер — физическое устройство расширения). При загрузке конфигурации значение записывается в канал модуля и может быть считано в программу ПЛК. Управление возможно как при старте, так и в ходе работы программы ПЛК. По умолчанию оба интерфейса RS-485 работают в режиме «Физический мастер».

Также в модуле «Extended settings» индицируется состояние батареи питания часов ПЛК.

Параметры модуля:

- **«Состояние батареи» (Batery discharged)** – в зависимости от состояния батареи данный параметр принимает состояние: **False (0)** – батарея заряжена, **True (1)** – до окончания срока работы батарее не более 5 месяцев. Поле в модуле дублирует индикатор на лицевой панели и включается при снижении напряжения на батарее до 2,7 В. Часы ПЛК сохраняют работоспособность при снижении напряжения питания на батарее до 1,4 В. После включения индикатора «батарея разряжена» работа часов при нормальных условиях обеспечивается ещё на протяжении 6 месяцев;
- **«Подтягивающий резистор интерфейса RS 485-1» (RS 485-1 master mode)** – значения выбираются из списка **«master device»** и **«terminal device»**, значение по умолчанию –

«master device». Значение выбирается в зависимости от режима работы по данному порту: **«master device»** – режим master и **«terminal device»** – режим slave на данном порте;

- **«Подтягивающий резистор интерфейса RS 485-2» (RS 485-2 master mode)** – значения выбираются из списка **«master device»** и **«terminal device»**, значение по умолчанию – **«master device»**. Значение выбирается в зависимости от режима работы по данному порту: **«master device»** – режим master и **«terminal device»** – режим slave на данном порте;

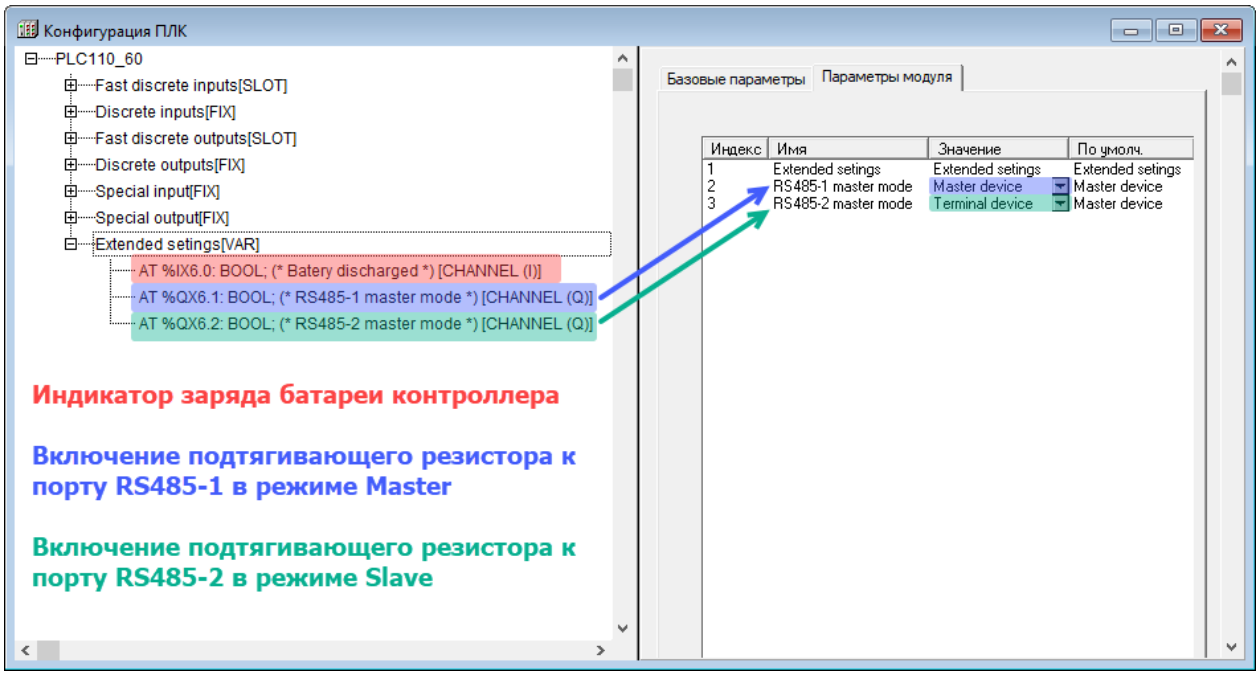


Рисунок 10.55 – Параметры модуля «Extended settings»

10.7.12 Настройка коммуникационных интерфейсов модуля

У большинства модулей есть возможность подключения подмодуля интерфейсного порта, через который будут передаваться и/или считываться данные. Интерфейсный порт можно заменить на другой с помощью команды «Replace element» (Заменить элемент) контекстного меню.

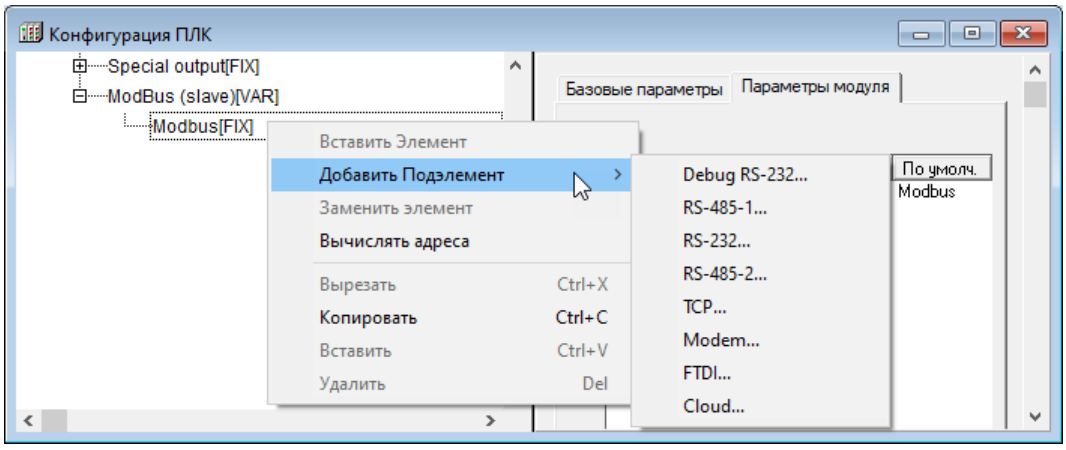


Рисунок 10.56 – Смена интерфейсного порта модуля «ModBus (slave)»

Подмодули последовательных портов

Параметры последовательных портов Debug RS-232, RS-232, RS-485-1 и RS-485-2, используемых в ПЛК, идентичны (см. рисунок 10.57). Для конфигурирования меняется только название и физический порт, в котором происходит работа.

Таблица 10.7 – Нумерация последовательных портов для программировании

Последовательный порт	RS-485-1	RS-232	RS-485-2	Debug RS-232
Программный номер порта	COM0	COM1	COM2	COM4

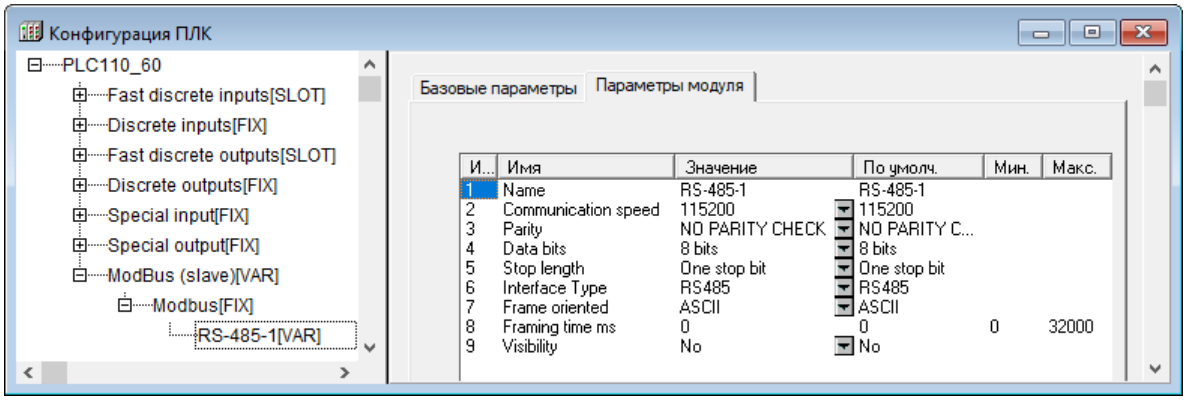


Рисунок 10.57 – Параметры последовательных портов

Параметры последовательного порта:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Communication speed (скорость передачи информации)** – скорость передачи информации через последовательный порт. Значения выбираются из списка (2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, 115200), значение по умолчанию – 115200;
- **Parity (проверка четности)** – наличие бита четности и его значение (четность, нечетность). Значения выбираются из списка (EVEN, ODD, SPACE, MARK) значение по умолчанию – «NOPARITYCHECK» (отсутствие проверки четности);
- **Data Bits (количество бит данных)** – количество значащих бит в одном байте посылке. Значения в диапазоне от 5 до 8 бит, значение по умолчанию – 8;
- **Stop Length (количество стоп-битов)** – количество стоп-битов. Значения выбираются из списка (один, полтора или два стоп-бита), значение по умолчанию – один стоп-бит (Onestopbit);
- **Interface Type (тип интерфейса)** – тип последовательного интерфейса, по которому осуществляется информационный обмен. Задается при выборе подэлемента (**RS-232** или **RS-485**);
- **Frame Oriented (тип протокола обмена)** – значения выбираются из списка «ASCII» и «RTU», значение по умолчанию – ASCII;

ПРИМЕЧАНИЕ

В ПЛК используются следующие типы протоколов обмена: ориентированный на передачу текстовых символов ASCII и ориентированный на передачу потока байтов RTU.

В ASCII-режиме информация передается последовательностью символов, и начало и окончание посылки имеют четко обозначенные специальные символы, обычно это – символы решетки, перевода строки и др.

В RTU-режиме иная структура передачи информации: передаются байты, несущие полезную информацию, без какого-либо указания начальных и/или конечных границ (заголовочных и конечных байтов). Сама посылка и ее границы определяются по наличию разрыва. Если время разрыва превышает определенное время (например, для Modbus – время передачи 3,5 символов), устройство определяет, что посылка закончилась, началась другая посылка. Таким образом, посылки отделяются друг от друга и их можно идентифицировать.

- **Framing time (время, на которое необходимо задерживать ответ на запрос в мс)** – временная задержка между последним байтом принятого пакета и первым байтом, передаваемым в ответ. Задержка бывает необходима для работы с устройствами с низкими скоростями информационного обмена. Рекомендуемый диапазон значений от 0 до 50 мс (для работы в режиме «slave»), значение по умолчанию – 0 (для работы в режиме «master»);
- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

Порт TCP

Помимо последовательного порта в системе устройств может использоваться порт TCP.

Параметры порта TCP:

- **Name** – имя элемента, см. раздел 10.6.1;
- **Remote Port (удаленный порт)** – адрес удаленного порта. Значения устанавливаются в диапазоне от 0 до 65535, значение по умолчанию – 502;

- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

ПРИМЕЧАНИЕ
Обращение мастера сети к ПЛК по Ethernet происходит по базовым сетевым настройкам ПЛК, заводским или пользовательским. Изменить сетевые настройки ПЛК можно в режиме «ПЛК-Браузер (PLC-Brauser)» (см. раздел 12). В данном случае модуль использует MAC-адрес и IP-адрес контроллера.

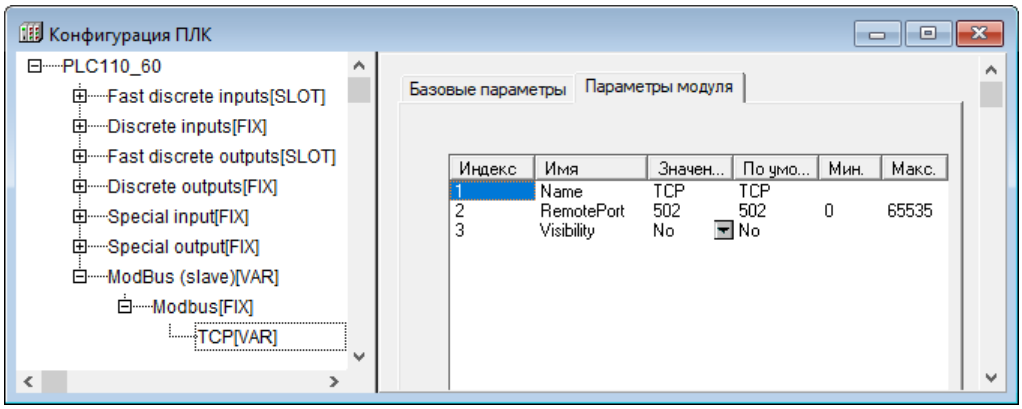


Рисунок 10.58 – Параметры порта TCP

Порт Empty

В ряде модулей, работающих в режиме master, также может использоваться порт Empty. Данный порт используется при передачи данных по Ethernet и для освобождения последовательных портов при работе по TCP.

Подмодуль Cloud

Подмодуль Cloud используется для подключения контроллеров к облачному сервису OwenCloud. Доступ к облачному сервису осуществляется через подключение контроллера к локальной сети с доступом в Интернет. Для передачи данных используется протокол Modbus TCP.

Подробнее о подключении к облачному сервису OwenCloud см. раздел 13.

Параметры подмодуля Cloud:

- **Name** – имя элемента, см. раздел 10.6.1;
- **CloudToken** – уникальный ключ, который вводится в конфигурации устройства для соединения с облачным сервисом. Генерируется при добавлении прибора в OwenCloud;
- **SSLKeyFileName** – параметр не используется, зарезервирован на будущее;
- **Visibility (видимость)** – видимость параметров модуля в протоколе «Gateway» (в частности, в программе «EasyWorkPLC» разработки компании «ОВЕН»). Значения выбираются из списка **«yes»** и **«no»**, значение по умолчанию – **«no»**.

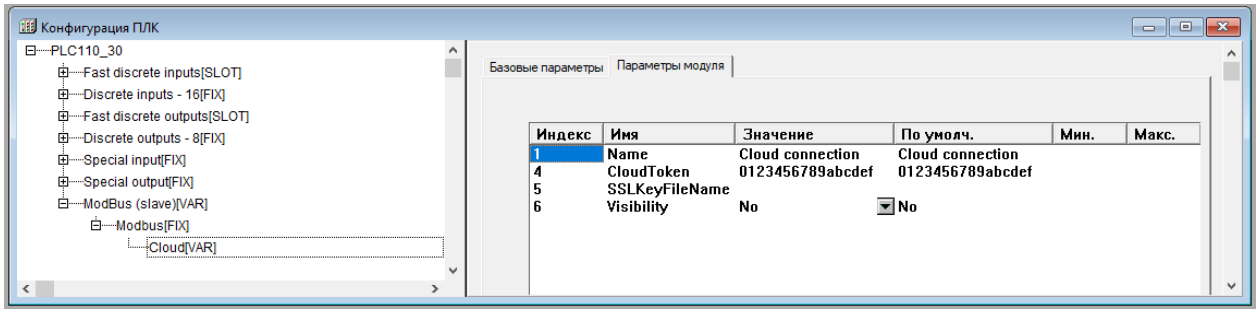


Рисунок 10.59 – Параметры подмодуля Cloud

Настройка модемного подключения

В качестве интерфейса в контроллере может также выступать модем. В зависимости от используемого режима модема указываются настройки.

Особенностью ПЛК является реализация поддержки протоколов стека PPP через модем (GPRS). Поддерживается один сетевой интерфейс PPP (№ 1) и один модем на любом из портов ПЛК.

Маршрутизация пакетов из Ethernet (сетевой интерфейс № 0) в PPP и обратно в автоматическом режиме не поддерживается.

Для настройки PPP соединения следует отредактировать файл *local_addres.dat* в соответствии с настройками порта, типом модема, адресом точки доступа и настройками протокола.

Режим GPRS

Для работы с модемом в режиме GPRS не требуется установка PPP-драйвера, но рекомендуется установить порт Empty (см. рисунок 10.60).

Для корректной работы модема в режиме GPRS требуется скорректировать файл ПЛК *local_addres.dat*. Подробнее о настройке см. раздел *Примеры с GSM/GPRS модемом ПМ01* на странице CODESYS на сайте owen.ru.

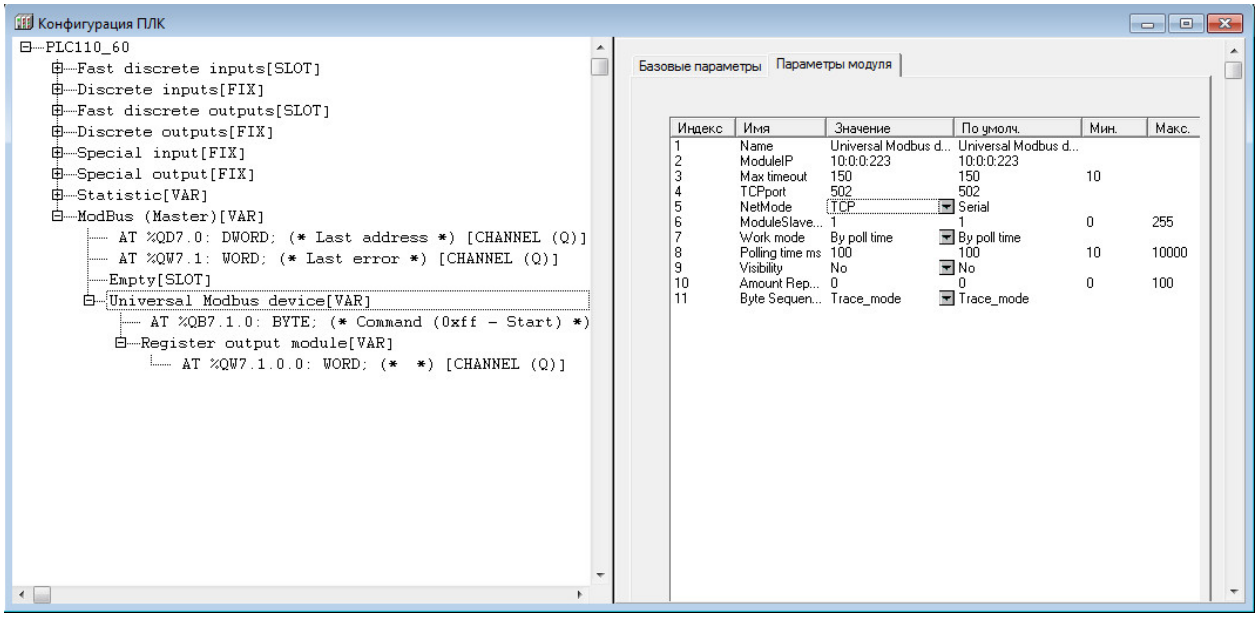


Рисунок 10.60 – Конфигурация модуля при использовании модема

11 Настройка дополнительных функций

ПЛК110 содержат ряд дополнительных функций (часы реального времени и др.), которые настраиваются в режимах «Конфигурация ПЛК (PLC Configuration)» и «ПЛК-браузер (PLC Browser)» CODESYS. В этих же режимах можно настроить поддерживаемые в ПЛК110 интерфейсы или протоколы обмена и конфигурирования.

Подробнее о работе режима «ПЛК-браузер (PLC Browser)» см. в разделе 12.

11.1 Значение часов реального времени

Часы реального времени в ПЛК110 имеют независимый источник питания и продолжают отсчет времени при выключении основного питания контроллера.

ПРИМЕЧАНИЕ
Тип источника питания указан в *Руководстве по эксплуатации* ПЛК.

Время, в течение которого встроенные часы будут идти при отсутствии внешнего питания ПЛК, зависит от состояния встроенного источника питания и условий хранения. Так, при хранении в помещении с комнатной температурой и новом источнике питания время работы часов составит 5 лет. Поэтому значения часов следует задавать однократно, либо после длительного (более 6 месяцев) неиспользования ПЛК.

Значение часов задается в режиме «ПЛК-браузер (PLC Browser)» CODESYS командами **SetTime** (Задание времени), **SetDate** (задание календарной даты).

Команды могут быть введены в командной строке режима вручную или выбором стандартных команд из перечня, вызываемого нажатием кнопки с тремя точками, расположенной у правого края командной строки.

Значения времени и даты просматриваются командой **GetTime**.

ПРИМЕЧАНИЕ
Значение времени можно установить или считать с часов с помощью библиотеки SysLibTime. Подробнее см. в документе *Библиотека SysLibFile.lib* на странице CODESYS V2.3 на сайте owen.ru.

11.2 Порт Ethernet

Для корректной работы порта Ethernet в сети следует задать ему необходимый IP-адрес, а также маску подсети IP-адреса.

Параметры задаются в режиме «ПЛК-браузер (PLC Browser)» командами **SetIP** и **SetMask**. Измененные значения вступают в силу только после перезагрузки ПЛК.

Значения сетевых параметров сохраняются в файле *local_addres. dat* на внутреннем Flash-диске ПЛК110. Файл может быть считан с контроллера на ПК вызовом команды **Онлайн → Читать файл из ПЛК** главного меню.

Текущие настройки интерфейса можно посмотреть в режиме «ПЛК-браузер (PLC Browser)» по команде **PLCInfo**.

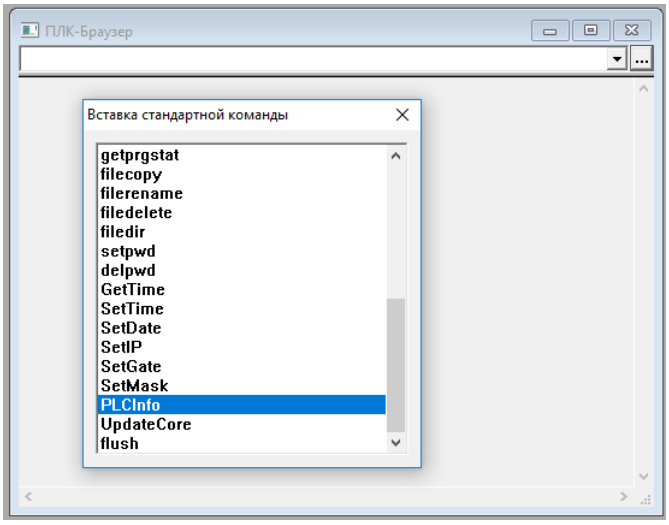


Рисунок 11.1 – Вызов команды текущих настроек интерфейса

11.3 Структура файла local_addres.dat

Значения параметров интерфейса Ethernet сохраняются в файле *local_addres. dat* на внутреннем Flash-диске ПЛК110. Файл может быть считан с контроллера на ПК вызовом команды **Онлайн → Читать файл из ПЛК** главного меню.

Пример

Описание структуры файла и основных настроек.

```
EMAC=e4:1e:0a:00:0c:2c // MAC-адрес ПЛК (находится на корпусе ПЛК)
IP=0A:02:19:5D // IP-адрес ПЛК (10.2.25.93)
GATE=0A:02:01:01 // Адрес шлюза в сети (10.2.1.1)
MASK=FF:FF:00:00 // Маска в сети (255.255.0.0)
DNS=8:8:8:8 // Первый сервер DNS (поддержано до четырех DNS серверов)
DNS=8:8:4:4 // Второй сервер DNS
DNS=0:0:0:0 // Третий сервер DNS
DNS=0:0:0:0 // Четвертый сервер DNS
DHCP=0 // Режим DHCP клиента (0 - отключен, 1 - включен)
```

ВНИМАНИЕ

В файле в качестве разделителя используется **двоеточие** (**:**), а все поля задаются в шестнадцатеричном формате (**HEX**).

ПРИМЕЧАНИЕ

Файл *local_addres.dat* может содержать дополнительные настройки, например, для обмена по каналу GPRS через модем ПМ01. Подробнее см. раздел «Примеры» страницы CODESYS V2 на сайте owen.ru.

12 Режим «ПЛК-Браузер»

Режим «PLC-Browser» (ПЛК-Браузер) предназначен для мониторинга (диагностики) состояния, настройки функционирования ПЛК и позволяет:

- вводить команды в виде текстовых строк;
- передавать команды в ПЛК;
- получать в качестве реакции ПЛК запрошенную информацию или отчет о результатах выполнения команд.

Вход в режим

Работа в режиме «ПЛК-Браузер» (PLC-Browser) возможна только после физического подключения ПЛК к ПК и установки связи с контроллером (связь устанавливается командой **Онлайн** → **Подключение** в главном меню). Для входа в режим «ПЛК-Браузер» (PLC-Browser) следует выбрать строку «ПЛК-Браузер» (PLC-Browser) на вкладку «Ресурсы» (PLC-Browser) организатора объектов.

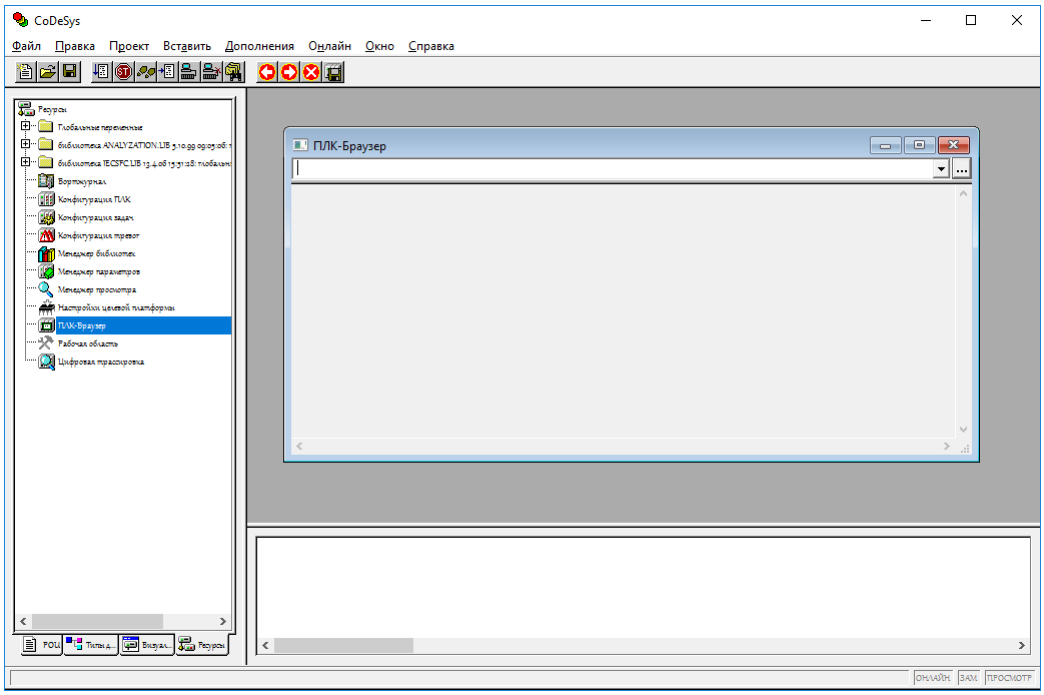


Рисунок 12.1 – Окно режима «ПЛК-Браузер» (PLC-Browser)

Рабочее поле окна режима разделено на две части: в верхней части окна отображается строка вводимых пользователем команд, в нижней части – поле отображения реакции ПЛК на введенную команду.

Кнопкой  в правой части строки команд вызывается раскрывающийся список, содержащий список (стек) всех ранее введенных со времени запуска проекта команд, автоматически дополняемый впервые вводимыми в рамках проекта командами, список сохраняется до закрытия проекта.

Кнопкой с тремя точками, расположенной справа от строки команд, вызывается перечень доступных команд, в котором можно выбрать требуемую команду и нажатием клавиши <Enter> перенести ее в строку команд.

Введенная команда передается в контроллер нажатием клавиши <Enter>. В нижней части рабочего поля отображается реакция ПЛК на введенную команду.

Команды PLC-Browser

Команды режима PLC-Browser обеспечивают выполнение функций манипулирования памятью, файлами, управления программами и информационных функций системы исполнения. Синтаксис команд:

```
<команда><пробел><параметры>
```

Список параметров определяется типом команды. Переданная команда повторяется в окне отображения вместе с ответом контроллера. После открытия проекта список доступных команд режима можно получить, введя команду «?» (знак вопроса).

Команды режима, применяемые в ПЛК, представлены в таблицах ниже.

ПРИМЕЧАНИЕ
Команды перечня стандартных команд PLC-Browser «saveretain» (запись сохраняемых (retain) переменных) и «restoreretain» (чтение сохраняемых (retain) переменных) в ПЛК не используются.

Таблица 12.1 – Перечень команд PLC-Browser, применяемых в ПЛК

Команда	Содержание	Описание
?	show implemented commands	Запрос у системы исполнения актуального списка всех поддерживаемых команд
mem	Memory dump	Hex-дамп области памяти. Синтаксис 1: *mem <start address><end address>*; Синтаксис 2: *mem <start address>-<end address>*. Адрес вводится в виде десятичного или шестнадцатеричного числа (префикс 16#)
memc	Memory dump relative to code-startaddress	Относительный hex-дамп области кода (аналогична команде mem, адрес задается от начала области кода)
memd	Memory dump relative to data-startaddress	Относительный hex-дамп области данных (аналогична команде mem, адрес задается от начала области данных)
reflect	reflect current command (test!)	Возврат строки (для тестирования соединения)
dpt	get data-pointer-table	Чтение таблицы указателей данных
ppt	get POU-table	Чтение таблицы POU
pid	get project ID	Чтение идентификатора проекта
pint	get project info	Чтение информации о проекте
startprg	Start PLC program	Запуск программы ПЛК
stopprg	Stop PLC program	Остановка программы ПЛК
resetprg	Reset PLC program	Сброс программы ПЛК – инициализируются только не энергонезависимые переменные
resetprgcold	Reset PLC program cold	Холодный сброс программы ПЛК инициализируется все, в том числе, и энергонезависимые переменные
resetprgorg	Reset PLC program original	Заводской сброс программы ПЛК – полная очистка областей кода и данных
reload	Reload boot project	Перезапись загрузочного кода проекта
getprgprop	Program properties	Свойства программы
getprgstat	Program status	Статус программы
filecopy	Copy files [from] [to]	Копирование файла [из] [в]
filerename	Rename files [old] [new]	Переименование файла [старое имя] [новое имя]
filedelete	Delete file [filename]	Удаление файла [имя файла]
filedir	display directory list	Файловая команда dir (дает лист перечня файлов)
setpwd	set login password	Установка пароля на контроллер. Синтаксис: setpwd <password> [level], где level может быть «0» (по умолчанию), действительный для подключения системы программирования, или «1», действительный для всех приложений
		Внимание! Пароль должен состоять только из **латинских букв** и/или **цифр**. Запрещено использовать пробел, знаки препинания и другие спецсимволы, так как это приведет к блокировке ПЛК, которую можно будет снять только с помощью сервисного центра.
delpwd	delete login password	Удалить пароль

Таблица 12.2 – Перечень команд разработчика ПЛК

Команда	Содержание	Описание
GetTime	return current time and date	Возврат текущего времени и даты
SetTime	Format [SetTime HH:MM:SS]	Установка времени в формате: часы, минуты, секунды
SetDate	Format [SetDate DD.MM. YYYY]	Установка даты в формате: день, месяц, год
SetIP	Format [SetIP XXX .XXX .XXX .XXX]	Установка IP-адреса в сети Ethernet
SetGate	Format f SetGate XXX .XXX .XXX . XXX]	Установка адреса шлюза в сети Ethernet
SetMask	Format [SetMaskXXX .XXX .XXX . XXX]	Установка маски в сети Ethernet
SetModemCfg	Format [SetModemCfg X]	Настройка конфигурации подключенного модема, где X может быть: • «0» (по умолчанию) – модем не подключен или будет использоваться пользовательским программным обеспечением через PLC-Configuration; • «1» – к порту (порт задается командой SetModemPort) подключен последовательный модем (используется в режиме прямого соединения с другим модемом); • «2» – к порту подключен последовательный модем (используется в режиме соединения с сетью Интернет GPRS-подключение)
PLCInfo	Information about PLC	Информация о типе ПЛК и его настройках
FactoryInfo		Информация об аппаратном обеспечении ПЛК и его серийном номере
UpdateCore	Обновление прошивки	Команда для перепрошивки контроллера
SetupRetainMode		Просмотр активного режима работы с RETAIN переменными
SetCyclicMode	Format [SetCyclicMode X]	Выбор режима циклической записи, где X может быть: • «0» – отключение режима циклической записи и переход к режиму записи по сигналу о пропадании питании контроллера; • от «1» до «1000» – значение периода в секундах, установка режима циклической записи Retain
rebooT		Перезагрузка ПЛК

formatFFS		Форматирование Flash-памяти ПЛК. Из Flash-памяти будут удалены все файлы, в том числе файл сетевых настроек *local_addres.dat*, сетевые настройки порта Ethernet будут сброшены до следующих:
		`IP 10.2.11.119`
		`GATE 10.2.1.1`
		`MASK 255.255.0.0`

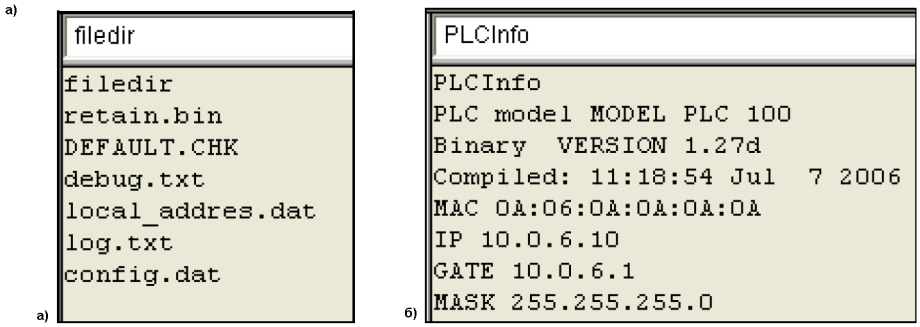


Рисунок 12.2 – Ввод команд (а) и реакция контроллера (б)

Если команда не распознана контроллером (введена с ошибкой), то в области отображения результата появится сообщение «Keyword not found» – ключевое слово не обнаружено.

Вспомогательные команды режима PLC-Browser

В режиме PLC-Browser в пункте «Дополнения» главного меню отображаются вспомогательные команды режима, а в панели инструментов – вспомогательные кнопки:

- команды (и кнопки) просмотра истории «Вперед» (History forward) и «Назад» (History backward) дают возможность «прокрутить» результаты выполненных команд вперед и назад по списку. Запись истории сохраняется до закрытия проекта;
- команда (и кнопка) «Cancel» прерывает начатый запрос;
- команда (и кнопка) «Save history list» сохраняет результаты выполненных команд в файле с расширением **.bhl* (Browser History List);
- команда «Print last command» открывает диалоговое окно печати. На печать будет выведен текущий запрос и его результат.

Настройка ПЛК: изменение сетевых настроек

Использование нескольких контроллеров в одной Ethernet-сети требует, чтобы их IP-адреса были уникальными. В режиме PLC-Browser можно, при работающем программном соединении, узнать имеющиеся сетевые настройки ПЛК и внести в них требуемые изменения. Для изменения физическое и программное соединение со всеми контроллерами компании «ОВЕН» удобнее устанавливать через COM-порт ПК.

Для изменения сетевых настроек следует:

1. Интерфейсным кабелем из комплекта поставки контроллера подключить COM-порт ПК с гнездом (RS-232), расположенным на лицевой панели контроллера (должно быть включено питание контроллер).
2. В CODESYS запустить созданный проект.
3. В главном меню выбрать команду **Онлайн → Параметры связи…**, в открывшемся окне нажать кнопку «New…».
4. В открывшемся окне новому соединению присвоить имя (например, «Owen_RS232») и выбрать из перечня вид соединения: «Serial (RS232)», в параметре «Baudrate» установить скорость 115200 бит/с.
5. В главном меню выбрать команду **Онлайн → Подключение** и подтвердить загрузку программы в контроллер.
6. Открыть окно режима «PLC-Browser» на вкладке «Ресурсы» (Resources) организатора объектов.
7. В командной строке режима «PLC-Browser» ввести команду «PLCInfo» и нажать на клавиатуре кнопку <Enter>. В ответе контроллера отобразится значения действующих параметров для IP-адреса (IP), маски (MASK) и шлюза (GATE) подсети.

8. Для изменения IP-адреса контроллера в командной строке режима «PLC-Browser» ввести команду и нужный адрес, например «SetIP 10.0.6.11», нажать на клавиатуре клавишу <Enter>, в поле отображения реакции ПЛК на введенную команду появится ответ контроллера, подтверждающий исполнение команды.

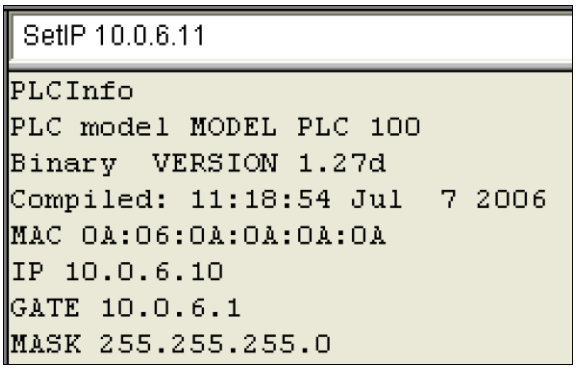


Рисунок 12.3 – Пример ввода команды для изменения IP-адреса контроллера в локальной сети

9. При необходимости изменить значения маски (MASK) и шлюза (GATE) в подсети командами «SetGate» и «SetMask» аналогично предыдущему пункту. После смены значения параметра GATE требуется перезагрузка контроллера. Перезагрузку лучше производить отключением питания контроллера с последующим включением через пять и более секунд.

10. Перезагрузить контроллер переводом переключателя на передней панели контроллера в положение «Сброс» и удержанием его в течение пяти или более секунд, либо отключением питания на время не менее пяти секунд с последующим включением; рекомендуется использовать перезагрузку выключением питания (программное соединение разрывается).

11. Повторно в главном меню выбрать команду **Онлайн** → **Подключение** и подтвердить загрузку программы в контроллер. Предварительно может потребоваться подтверждение выбора соединения с контроллером через COM-порт ПК.

12. В командной строке режима «PLC-Browser» ввести команду «PLCInfo» для проверки проведенных изменений.

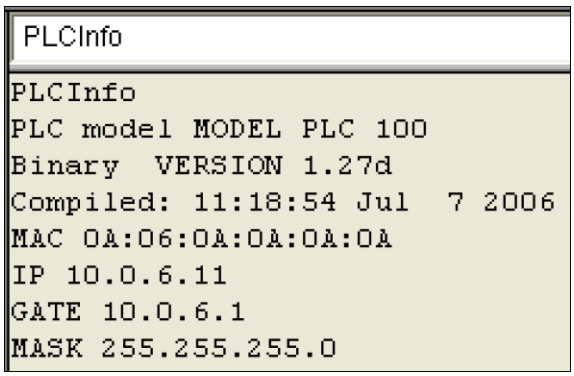


Рисунок 12.4 – Информация о новых настройках контроллера для локальной сети

После проверки связь с контроллером может быть установлена по интерфейсу Ethernet согласно новым сетевым настройкам.

13 Подключение к облачному сервису OwenCloud

13.1 Подключение ПЛК через Ethernet по протоколу Modbus TCP

Для подключения контроллера к облачному сервису OwenCloud не требуется наличие сетевых шлюзов линейки Пх210. Доступ к облачному сервису осуществляется через подключение контроллера к локальной сети с доступом в Интернет. Для передачи данных используется протокол Modbus TCP.

Для ПЛК функционал доступен начиная с версии встроенного ПО v0.3.66 и target-файла v3.12.

ПРИМЕЧАНИЕ
Во время работы с облачным сервисом OwenCloud используется сетевой порт **25502**. Следует убедиться, что порт не закрыт и не заблокирован.

Для подключения ПЛК к OwenCloud через Ethernet по протоколу Modbus TCP следует:

1. Подключиться к ПЛК. В главном меню выбрать команду **Онлайн** → **Читать файл из ПЛК**, в разделе **Имя файла** указать **local_addres.dat** и выбрать директорию на ПК, в которой будет сохранен данный файл.
2. Открыть файл *local_addres.dat* текстовым редактором (например, Notepad++). Файл будет иметь следующую структуру (количество полей может отличаться в зависимости от версии встроенного ПО):

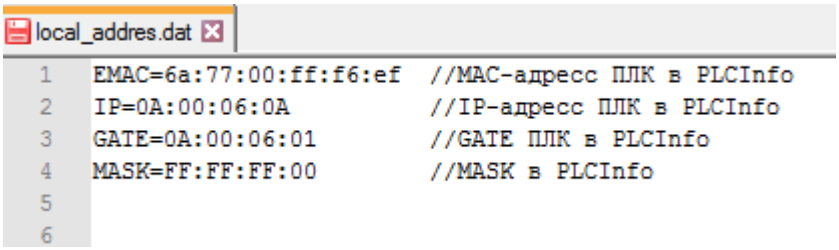


Рисунок 13.1 – Структура файла local_addres.dat

Следует обратить внимание на MAC-адрес (поле EMAC) – он понадобится далее.

3. Отредактировать файл *local_addres.dat* одним из способов:
   - **С использованием DHCP** – если в локальной сети есть DHCP-сервер, то можно переключить ПЛК в режим DHCP-клиента. В этом случае ПЛК при загрузке будет получать сетевые настройки от DHCP-сервера. Для переключения следует добавить в файл строку *DHCP=1*;

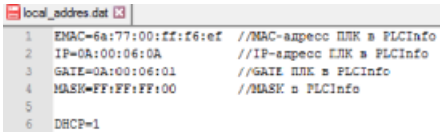


Рисунок 13.2 – Включение режима DCHP-клиента

   - **Статический IP-адрес** – если у ПЛК требуется статический IP-адрес, то его следует прописать в файле адреса **DNS-серверов**.

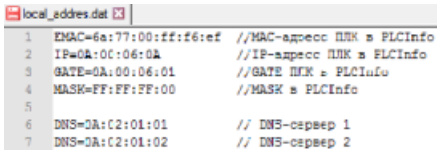


Рисунок 13.3 – Добавление DNS-серверов (пример для сети 10.2.1.x)

Всего может быть указано до четырех DNS-серверов.

ПРИМЕЧАНИЕ
На рисунке 13.3 указан только пример DNS-серверов. Пользователь должен указать адреса DNS-серверов из своей сети или публичных DNS-серверов (например, Google Public DNS: **08:08:08:08**).

В файле *local_addres.dat* используются значения в шестнадцатеричной системе (**HEX**), разделитель между октетами – **двоеточие** (**:**).

4. Сохранить отредактированный файл, не меняя его название. В главном меню вызвать команду **Онлайн** → **Записать файл в ПЛК** и загрузить в ПЛК отредактированный файл *local_addres.dat*.
5. Создать новый проект для ПЛК. На вкладке **Конфигурация ПЛК** добавить элемент **ModBus (Slave)** и задать для него адрес **1**.

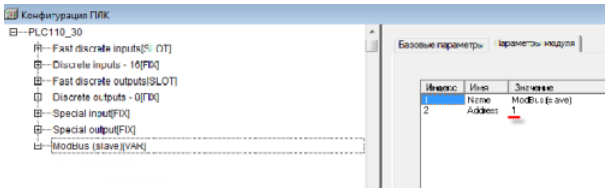


Рисунок 13.4 – Добавление и настройка элемента Modbus (Slave)

6. В элемент **ModBus (Slave)** добавить элемент **Cloud**.

   В параметре CloudToken потребуется ввести токен прибора, генерируемый при добавлении прибора в OwenCloud. На данном этапе токен отсутствует – он будет получен далее.

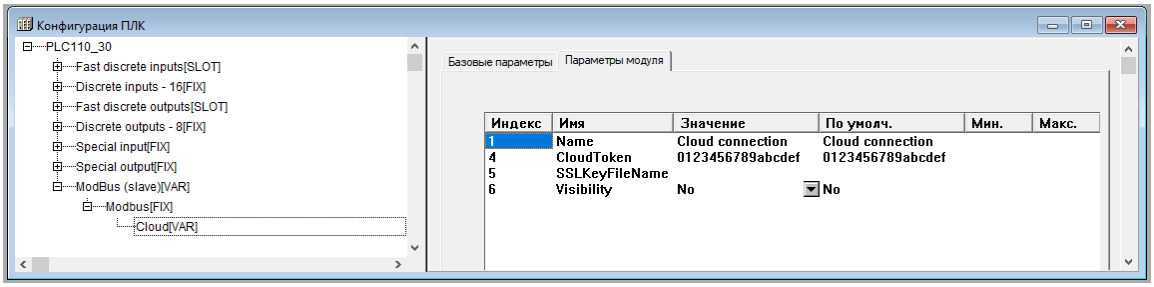


Рисунок 13.5 – Добавление элемента Cloud

7. Добавить в конфигурацию два подэлемента **2 byte** и один подэлемент **Float**. К подэлементам обязательно должны быть привязаны переменные – это является необходимым условием для импорта конфигурации ПЛК в OwenCloud. В результате в контроллере будет сформирована следующая карта регистров:

Таблица 13.1 – Карта регистров для ПЛК

Имя переменной	Тип	Адрес регистра (задается автоматически)	Описание
wVar1	WORD	0	Целочисленное значение
wVar2	WORD	1	Целочисленное значение
rVar	REAL	2–3	Значение с плавающей точкой

ПРИМЕЧАНИЕ
Переменная с плавающей точкой (**rVar**) занимает два регистра в памяти ПЛК (в данном случае, второй и третий). Адрес первого регистра для переменной типа **REAL** должен быть четным из-за особенностей выравнивания памяти ПЛК (подробнее см. в разделе 10.7.3.1).

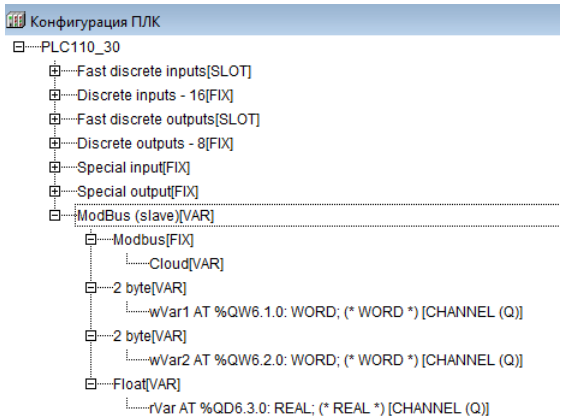
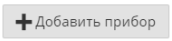


Рисунок 13.6 – Добавление переменных в ModBus (slave)

8. Зайти на главную страницу OwenCloud и авторизоваться. При отсутствии регистрации, зарегистрироваться.

9. На странице **Администрирование** открыть вкладку **Приборы** и нажать кнопку 

. В окне добавления прибора указать следующие настройки:
   - **Идентификатор** – MAC-адрес ПЛК (указан на корпусе ПЛК);
   - **Тип прибора** – тип «ПЛК через Modbus TCP»;
   - **Заводской номер** – заводской номер прибора (необязательно);
   - **Название прибора** – название прибора;
   - **Категории** – категории, к которым будет принадлежать прибор;
   - **Часовой пояс** – часовой пояс, в котором находится прибор.

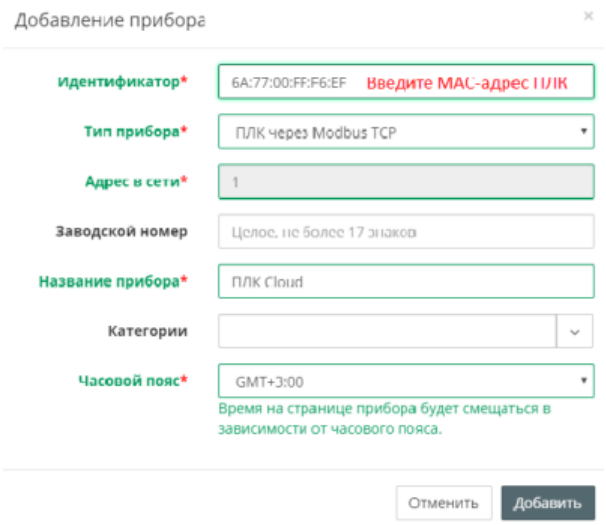


Рисунок 13.7 – Окно добавления прибора

Для завершения нажать кнопку **Добавить**.

10. На вкладке **Общие/Общие настройки** будет отображаться токен ПЛК. Его следует скопировать и вставить в CODESYS в настройках элемента Cloud.

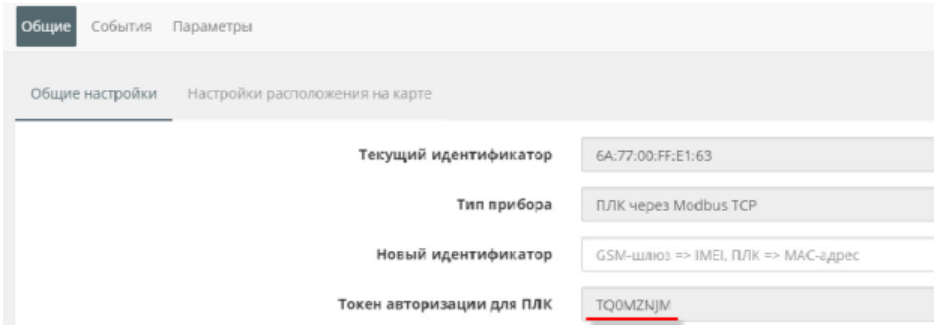


Рисунок 13.8 – Копирование токена из OwenCloud

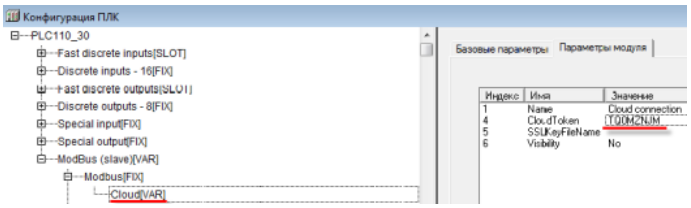


Рисунок 13.9 – Ввод токена OwenCloud в конфигурацию ПЛК

11. В главном меню CODESYS выбрать команду **Проект** → **Экспорт** и сохранить конфигурацию ПЛК в виде файла формата **.exp**.

12. В OwenCloud на вкладке **Параметры/Настройки параметров Modbus** нажать кнопку **Импортировать**, выбрать пункт **Загрузить из Codesys 2.3** и указать путь к созданному файлу в формате **.exp**. В результате в OwenCloud будут автоматически добавлены параметры из конфигурации ПЛК.

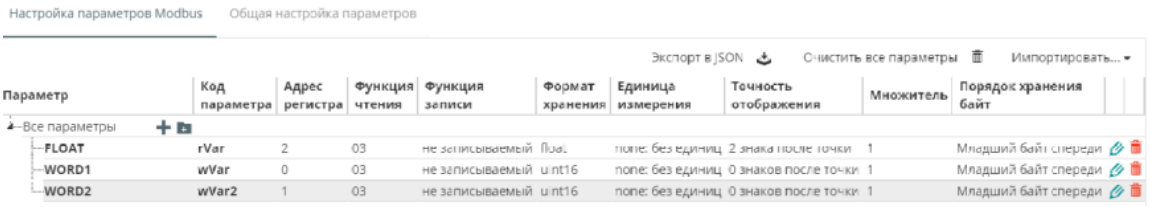


Рисунок 13.10 – Импортированные параметры Modbus

13. Нажать на кнопку , чтобы перейти к редактированию параметра. Снять галочку **Порядок хранения байт: Младший байт спереди** и выбрать нужную функцию записи (для параметров типа **Uint16** – функцию записи **06**, для переменной типа **float** – функцию записи **16**).

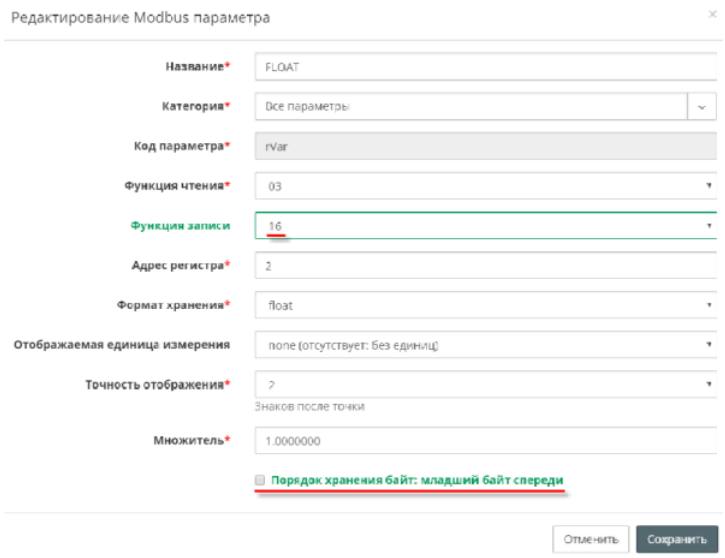


Рисунок 13.11 – Редактирование параметров Modbus

14. Загрузить проект в ПЛК **(Онлайн → Подключение)**. Создать загрузочное приложение **(Онлайн → Создать загрузочное приложение)** и запустить проект **(Онлайн → Старт)**.

15. Подключить ПЛК к локальной сети, которая имеет доступ в Интернет.

16. Нажать на кнопку , чтобы перейти к просмотру значений параметров прибора. Изменить значения переменных в **Codesys 2.3** и наблюдать соответствующие изменения в OwenCloud. Для изменения значения из OwenCloud следует перейти на вкладку **Запись параметров**.

Рисунок 13.12 – Просмотр параметров прибора

13.2 Подключение ПЛК через шлюз Пх210 по протоколу Modbus RTU

Доступ к облачному сервису OwenCloud осуществляется с помощью шлюза Пх210 с доступом в Интернет. Для передачи данных используется протокол Modbus RTU.

ПРИМЕЧАНИЕ
Подробная настройка доступа в Интернет шлюза Пх210 описана в *кратком руководстве* шлюза.

Для подключения ПЛК к OwenCloud через шлюз Пх210 по протоколу Modbus RTU следует:

1. Создать новый проект для ПЛК. На вкладке **Конфигурация ПЛК** добавить элемент **ModBus (Slave)** и задать для него адрес **1**.

Рисунок 13.13 – Добавление и настройка элемента Modbus (Slave)

2. В элемент **Modbus (Slave)** добавить элемент **RS-485-1** (или **RS-485-2** – в зависимости от используемого интерфейса ПЛК) и задать ему следующие настройки:

Таблица 13.2 – Настройки интерфейса

Параметр	Значение
Скорость (Communication Speed)	115 200 бит/с
Четность (Parity)	Нет
Биты данных (Data bits)	8
Число стоп-бит (Stop length)	1
Протокол (Frame oriented)	RTU

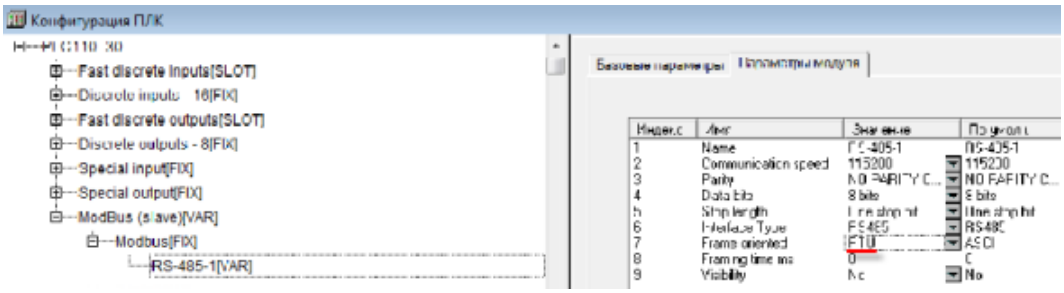


Рисунок 13.14 – Настройки интерфейса RS-485

3. Добавить в конфигурацию два подэлемента **2 byte** и один подэлемент **Float**. К подэлементам обязательно должны быть привязаны переменные – это является необходимым условием для импорта конфигурации ПЛК в OwenCloud. В результате в контроллере будет сформирована следующая карта регистров:

Таблица 13.3 – Карта регистров для ПЛК

Имя переменной	Тип	Адрес регистра (задается автоматически)	Описание
wVar1	WORD	0	Целочисленное значение
wVar2	WORD	1	Целочисленное значение
rVar	REAL	2–3	Значение с плавающей точкой

ПРИМЕЧАНИЕ
Переменная с плавающей точкой (**rVar**) занимает два регистра в памяти ПЛК (в данном случае, второй и третий). Адрес первого регистра для переменной типа **REAL** должен быть четным из-за особенностей выравнивания памяти ПЛК (подробнее см. в разделе 10.7.3.1).

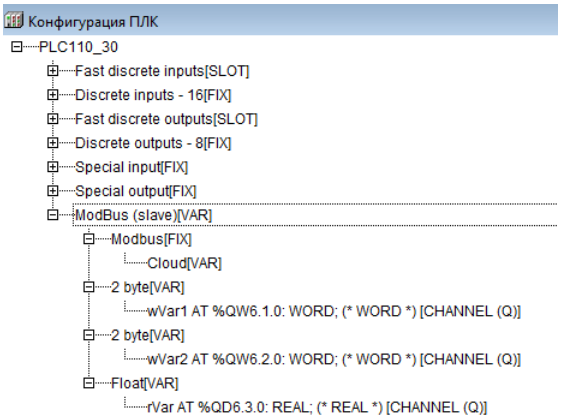
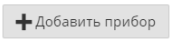


Рисунок 13.15 – Добавление переменных в ModBus (slave)

4. В главном меню CODESYS выбрать команду **Проект** → **Экспорт** и сохранить конфигурацию ПЛК в виде файла формата **.exp**.
5. Загрузить проект в ПЛК **(Онлайн** → **Подключение)**. Создать загрузочное приложение **(Онлайн** → **Создать загрузочное приложение)** и запустить проект **(Онлайн** → **Старт)**.
6. Подключить Пх210 к интерфейсу **RS485-1** или **RS485-2**, в зависимости от выбора. Примеры подключений Пх210 можно найти в кратком руководстве на шлюз.
7. Зайти на главную страницу OwenCloud и авторизоваться. При отсутствии регистрации, зарегистрироваться.

8. На странице **Администрирование** открыть вкладку **Приборы** и нажать кнопку 

. В окне добавления прибора указать следующие настройки:
   - **Идентификатор** – **IMEI** для ПМ210 или **заводской номер** для ПВ210 и ПЕ210;
   - **Тип прибора** – тип «Произвольное устройство Modbus»;
   - **Заводской номер** – заводской номер прибора (необязательно);
   - **Название прибора** – название прибора;
   - **Категории** – категории, к которым будет принадлежать прибор;
   - **Часовой пояс** – часовой пояс, в котором находится прибор.

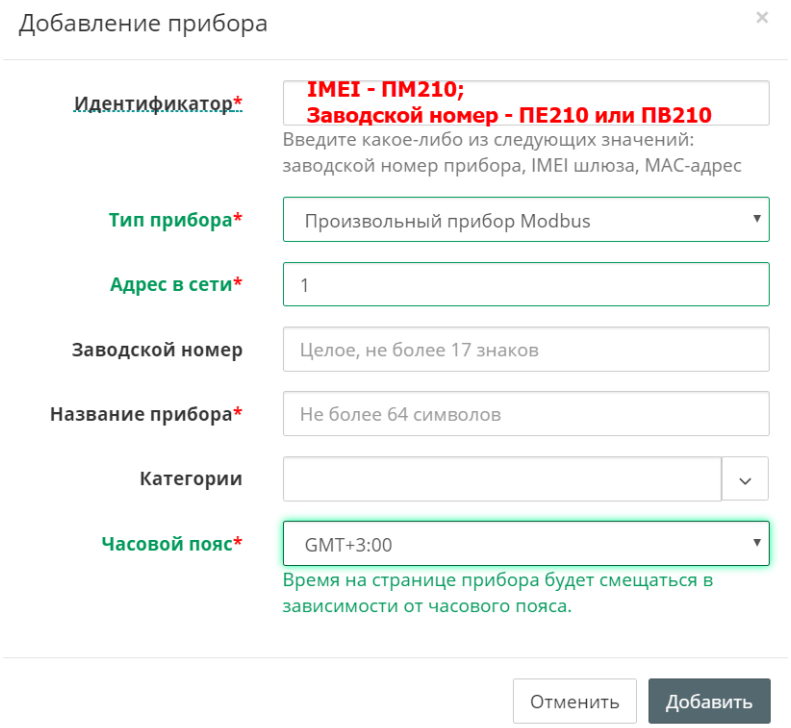


Рисунок 13.16 – Окно добавления прибора

Для завершения нажать кнопку **Добавить**.

9. В OwenCloud на вкладке **Общее/Общие настройки** указать скорость опроса и настройки COM-порта прибора. Нажать кнопку **Сохранить** для применения новых настроек. В случае необходимости настройки можно изменить.

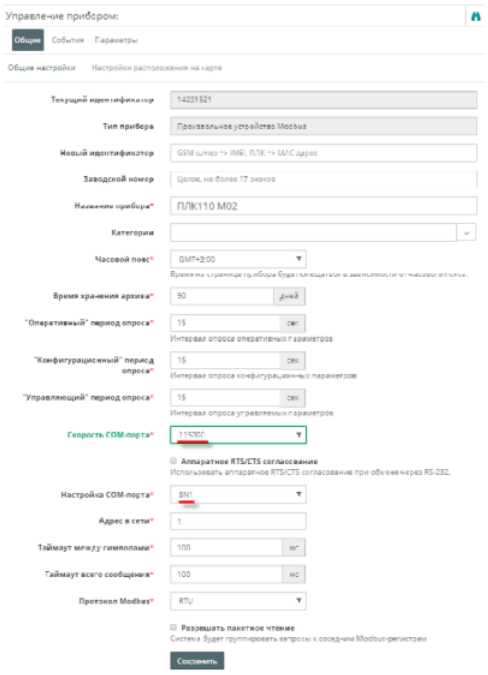


Рисунок 13.17 – Ввод сетевых настроек в OwenCloud

10. В OwenCloud на вкладке **Параметры/Настройки параметров Modbus** нажать кнопку **Импортировать**, выбрать пункт **Загрузить из Codesys 2.3** и указать путь к созданному файлу в формате **.exp**. В результате в OwenCloud будут автоматически добавлены параметры из конфигурации ПЛК.

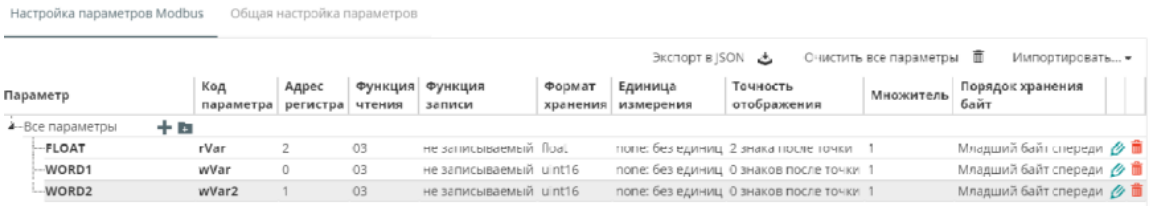


Рисунок 13.18 – Импортированные параметры Modbus

11. Нажать на кнопку , чтобы перейти к редактированию параметра. Снять галочку **Порядок хранения байт: Младший байт спереди** и выбрать нужную функцию записи (для параметров типа **Uint16** – функцию записи **06**, для переменной типа **float** – функцию записи **16**).

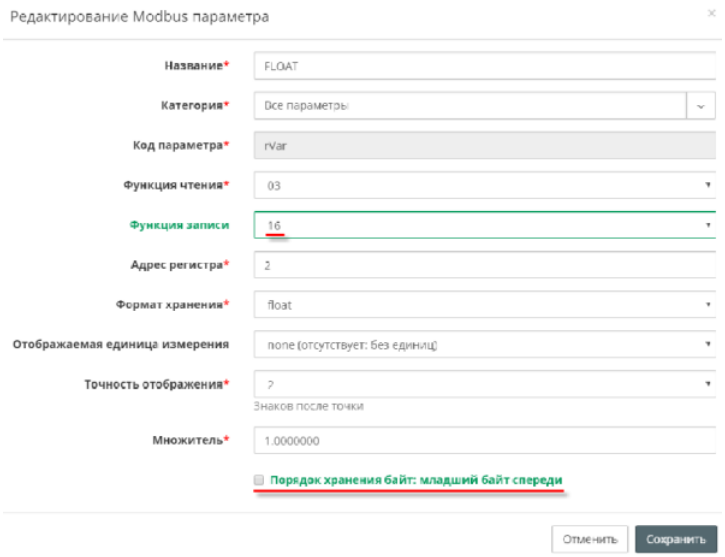


Рисунок 13.19 – Редактирование параметров Modbus

12. Нажать на кнопку , чтобы перейти к просмотру значений параметров прибора. Изменить значения переменных в **Codesys 2.3** и наблюдать соответствующие изменения в OwenCloud. Для изменения значения из OwenCloud следует перейти на вкладку **Запись параметров**.

Рисунок 13.20 – Просмотр параметров прибора

14 Работа с высокочастотным таймером

ПЛК110 имеет встроенный таймер, по прерыванию которого может быть вызвана отдельная программа, не связанная с выполнением основной программы ПЛК. Пример такой программы приведен в файле «hi_timer.pro», который можно скачать в архиве «Работа с быстрыми входами/выходами по прерыванию высокочастотного таймера» в разделе «Примеры» на странице CODESYS V2 на сайте owen.ru. Перед открытием проекта его следует скопировать на жесткий диск ПК.

В программе, вызываемой по прерыванию встроенного таймера, могут обрабатываться состояния «быстрых» входов и выходов ПЛК (Fast inputs & Fast outputs). Подробно о количестве «быстрых» входов и выходов см. *Руководство по эксплуатации* ПЛК.

Такой режим обработки может потребоваться для задач, время обработки которых должно быть существенно меньше времени цикла ПЛК или для автоматизации объектов, критичных ко времени реакции на определенные события. Минимальный период вызовов прерываний таймера составляет 20 мкс и может быть увеличен при вызове функции инициализации (период должен быть кратен 20 мкс).

В качестве примера рассмотрим объект, требующий выключение исполнительного механизма при замыкании дискретного датчика. Датчик подключен к «быстрому» входу 1 ПЛК110, исполнительный механизм управляется «быстрым» выходом 1 того же контроллера. Также требуется вычислять, сколько таких переключений произошло между началами основных циклов ПЛК.

Для создания программы обработки прерывания таймера следует:

1. Создать проект в CODESYS. Процедура создания описана в разделе 6.1. В примере используется контроллер ПЛК110-60.К-М. Основная программа контроллера – PLC_PRG будет написана на языке ST.
2. Перейти на вкладку «POU» организатора объектов.
3. Выбрать команду **Добавить объект…** контекстного меню вкладки «POU» организатора объектов или команду **Проект** → **Объект** → **Добавить** в главном меню.
4. В открывшемся окне задать тип, имя и язык написания POU: добавляемый POU должен быть типа «Программа» (Program), имя и язык написания могут быть любыми, однако, **рекомендуется использовать язык ST**, что повысит скорость выполнения программы и позволит придерживаться минимального периода вызовов прерываний таймера.

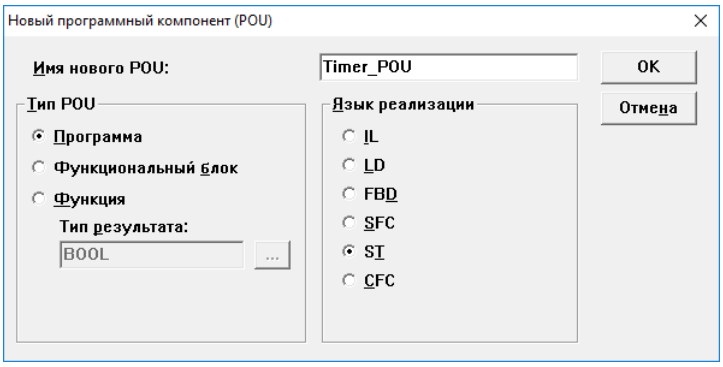


Рисунок 14.1 – Добавление POU

5. Для подключения POU к прерыванию таймера следует перейти в режим «Конфигурация задач» (Task Configuration) на вкладке «Ресурсы» организатора объектов. Откроется окно режима.

   В левой части окна режима следует выделить строку «Системные события» (System events). В правой части окна отобразится доступный для выбранного контроллера список прерываний. В списке следует выбрать прерывание «Timer» (установить флажок в поле соответствующего переключателя) и в колонке «Вызываемый POU» (called POU) указать имя созданного POU (в примере – «Timer_POU»).

   Использовать кнопку «Создать POU» (Create POU) не следует, т. к. она создает POU типа Function, а для работы требуется POU типа Program.

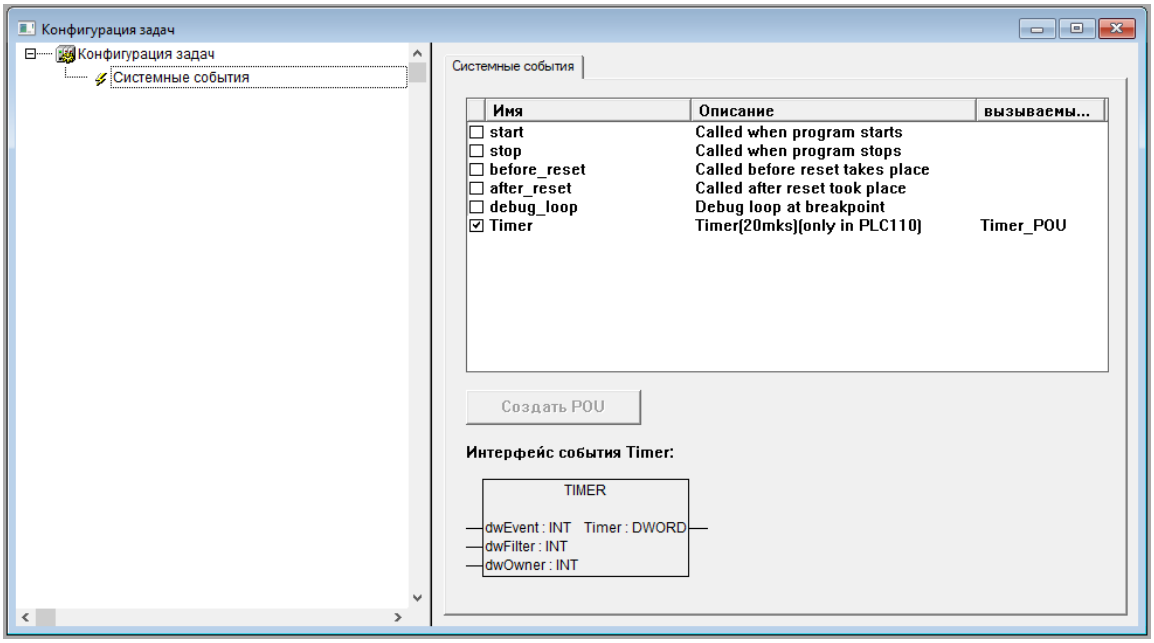


Рисунок 14.2 – Окно режима «Конфигурация задач» (Task Configuration)

6. Перевести «быстрые» дискретные входы и выходы в режим прямого управления (управления из POU), так как прерывание таймера вызывается в несколько десятков раз чаще, чем происходит цикл ПЛК, и использовать в POU обработки прерываний таймера значения из области памяти ввода-вывода не имеет смысла. Для исключения ситуации, когда одним и тем же входом или выходом управляет POU обработки прерывания таймера и основная программа ПЛК, следует перевести «быстрые» входы и выходы в режим прямого управления (Direct Control mode), в котором для «быстрых» входов и выходов не выделяется каналов в памяти ввода-вывода. Для перевода выходов в режим прямого управления следует перейти в режим «Конфигурация ПЛК» (PLC Configuration) и заменить модуль «Fast Discrete Outputs» на модуль «Fast Discrete Outputs – Direct Control» и модуль «Fast Discrete Inputs» – на модуль «Fast Discrete Inputs – Direct Control».

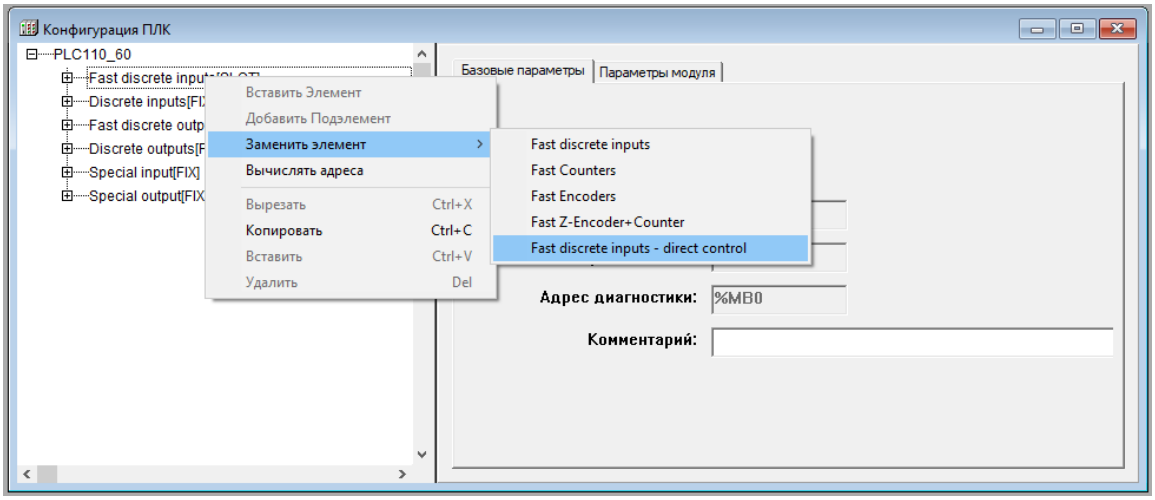


Рисунок 14.3 – Замена модуля

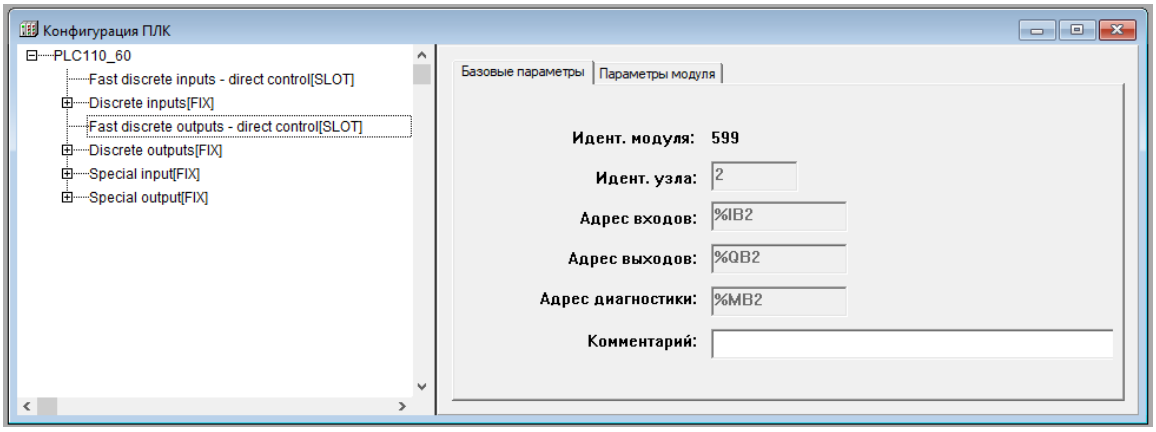


Рисунок 14.4 – Результат замены модулей

7. Подключить требуемые библиотеки. Таймер запускается из основной программы посредством вызова функции, входящей в состав библиотеки Timer.lib (одна из библиотек ОВЕН, поставляемых в комплекте с ПЛК). Работа с «быстрыми» входами и выходами из POU обработки прерываний таймера выполняется не через пространство памяти ввода-вывода, а посредством вызова функций библиотеки SysLibPorts.lib (библиотека CODESYS, поставляется в комплекте с ПЛК). Библиотеки подключаются в режиме «Менеджер библиотек (Library Manager)». Процедура подключения описана в разделе 7.4.

8. Перейти в режим «POU», выбрать в списке POU проекта требуемое окно и написать программу обработки прерывания таймера.

ВНИМАНИЕ
В POU обработки прерывания таймера не допускается выполнения сложных математических действий или вызов ресурсоемких функций, т. к. они могут не успеть завершиться до следующего вызова прерывания таймера.
В описываемом примере программа состоит из чтения состояния «быстрых» входов функцией SysPortIn (из библиотеки SysLibPorts.lib), инвертировании прочитанного значения и передаче его в «быстрые» выходы функцией SysPortOut (см. рисунок 14.5).
Для обеих функций работа ведется с портом 0, с младшими битами. Число ликвидных бит равно числу «быстрых» входов и выходов используемого контроллера.

ВНИМАНИЕ
Во время отладки прерывания таймера запрещается использовать точку остановки (breakpoint), так как это приведет к зависанию контроллера.

Timer_POU (PRG-ST)

```
0001 PROGRAM Timer_POU
0002 VAR
0003     in:BYTE;
0004     last_in:BYTE;
0005     old_res:DWORD;
0006 END_VAR
0007
```

```
0001 (*!! Запрещается использовать breakpoints при отладке в программах, обслуживающих прерывания !!*)
0002
0003 (* Сброс подсчитанного количества срабатываний при начале нового цикла ПЛК – пример чтения из глобальной переменной *)
0004 (* Глобальную переменную t_res в этом обработчике можно только читать, так как значения в нее пишутся в основной программе *)
0005 IF (t_res<>old_res) THEN (* Если в основном цикле ПЛК переменная t_res была изменена, то… *)
0006     old_res:=t_res; (* …запоминаем новое значение t_res в переменную old_res *)
0007     t_count:=0 (* …и обнуляем счётчик срабатываний *)
0008 END_IF
0009
0010 (* Чтение «быстрых» входов и запись «быстрых» выходов *)
0011 (* Не забудьте изменить в окне PLC Configuration модули«быстрых» входов и выходов на модули Direct Control *)
0012 in:=SysPortIn(0); (* Чтение «быстрых» входов из порта 0 *)
0013 SysPortOut(0, NOT(in)); (* Запись «быстрых» выходов в порт 0 инвертированным значением входов *)
0014
0015 (* Подсчёт числа изменений состояний выходов за цикл ПЛК – пример записи глобальных переменных *)
0016 IF (in<>last_in) THEN (* Если состояние входов изменилось, то… *)
0017     t_count=t_count+1; (* …увеличиваем счётчик *)
0018     last_in:=in; (* …и запоминаем текущее состояние *)
0019 END_IF
```

Рисунок 14.5 – Программа обработки высокочастотного таймера («Timer_POU»)

Для передачи информации о количестве срабатываний «быстрых» входов между вызовами циклов ПЛК из обработчика прерывания в основную программу используются глобальные переменные (см. рисунок 14.6). Работа с глобальными переменными ведется в режиме «Глобальные переменные» (Global Variables) организатора объектов.

Во время работы с глобальными переменными следует учитывать, что запись значения в глобальную переменную должна происходить только в POU обработки прерывания или в основной программе.

Global_Variables

```
0001 VAR_GLOBAL
0002     t_count:DWORD; (* Число изменений состояния «быстрых» дискретных входов за цикл ПЛК *)
0003     t_res:DWORD; (* Сигнал сброса счётчика t_count *)
0004 END_VAR
0005
0006
```

Рисунок 14.6 – Глобальные переменные, используемые программой обработки высокочастотного таймера («Timer_POU»)

9. Написать основную программу ПЛК, выбрав в списке объектов основную программу «PLC_PRG». В окне программы (см. рисунок 14.7) записать вызов функции инициализации прерывания таймера SetIRQ (из библиотеки Timer.lib). Аргументом функции является значение периода вызова прерывания в микросекундах (период должен быть кратен 20). Вызов функции требуется делать однократно при старте программы (пример кода представлен в проекте hi_timer.pro).

   Для передачи информации в обработчик прерывания таймера используются глобальные переменные. Для глобальных переменных также действует правило, предписывающее запись переменной только в одном POU.

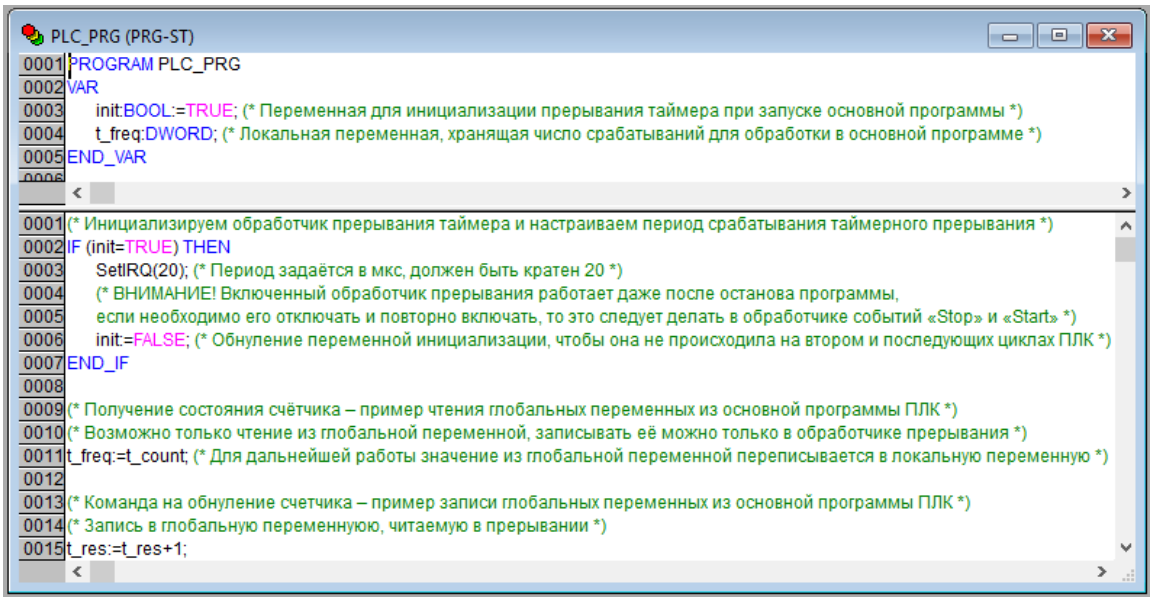


Рисунок 14.7 – Программа ПЛК, которая обращается к функции инициализации прерывания таймера «SetIRQ»

15 Обновление встроенного ПО и target-файлов

Компания ОВЕН совершенствует производимые контроллеры и их ПО, в том числе встроенное ПО контроллера и target-файлы ПЛК. Обновленное ПО контроллера 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 обновлять встроенное ПО не рекомендуется.

Обновленное ПО и соответствующие ему версии target-файлов можно скачать на сайте owen.ru.

ВНИМАНИЕ
Обновленное ПО контроллера и target-файл ПЛК должны иметь соответствующие версии.

Перед обновлением ПО контроллера, следует учесть, что если на ПЛК установлено встроенное ПО версии 2.02.0 или более поздняя, то контроллер подготовлен к возможности перепрограммирования ядра ПЛК в процессе эксплуатации ПЛК без его разборки (без снятия корпуса).

15.1 Определение текущей версии встроенного ПО контроллера

С помощью «гипертерминала»

ПРИМЕЧАНИЕ
Способ доступен для ОС Windows Vista/7/8.

Для определения текущей версии встроенного ПО контроллера с использованием «гипертерминала» следует:

1. Соединить контроллер с ПК через последовательный порт «RS-232 DEBUG».
2. В ОС выбрать команду **Пуск** → **Программы** → **Стандартные** → **Связь** → **HyperTerminal**.
3. В открывшемся окне «Описание подключения» в поле «Название» задать имя нового подключения и нажать кнопку «ОК».

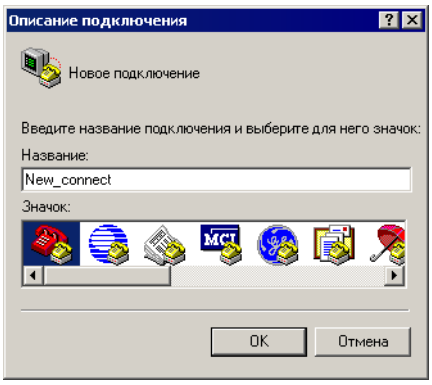


Рисунок 15.1 – Окно «Описание подключения»

4. В открывшемся окне «Подключение» в поле «Подключаться через» выбором из раскрывающегося списка задать COM-порт, к которому подключен ПЛК, и нажать кнопку «ОК».

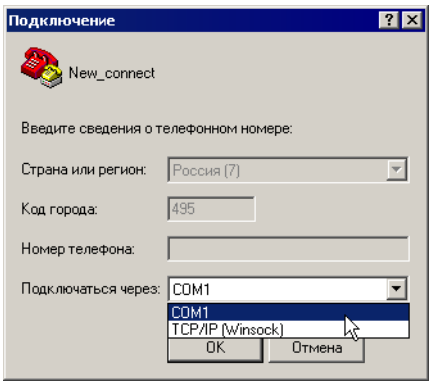


Рисунок 15.2 – Окно «Подключение»

5. В открывшемся окне «Свойства: COM» в поле «Скорость (бит/с» выбором из раскрывающегося списка задать значение «115200», в поле «Управление потоком» выбором из раскрывающегося списка задать значение «Нет», нажать кнопку «Применить», затем – нажать кнопку «ОК».

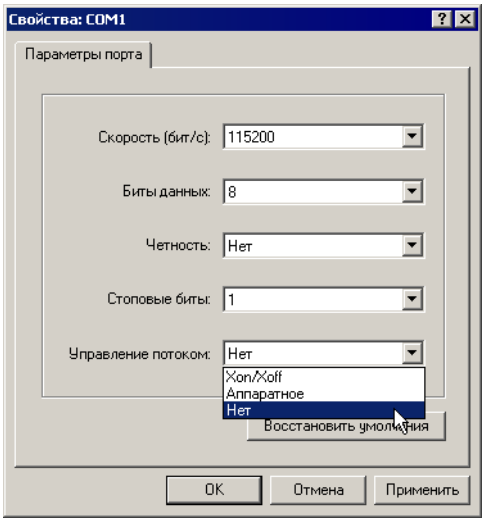


Рисунок 15.3 – Окно «Свойства COM-порта»

6. Нажатием кнопки «Сброс» перезагрузить ПЛК.
7. В открывшемся окне «HyperTerminal» в строке «Binary Version» отобразится номер текущей версии встроенного ПО контроллера.

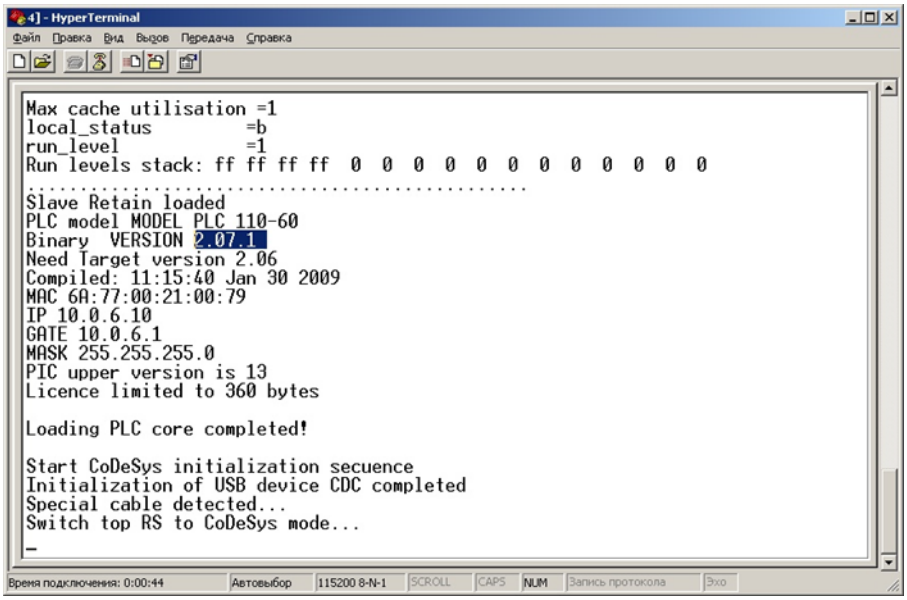


Рисунок 15.4 – Окно «HyperTerminal»

С помощью CODESYS

Для определения текущей версии ПО контроллера с использованием CODESYS следует:

1. Соединить контроллер с ПК через любой из портов для программирования (COM-порт, порт Ethernet или USB).
2. Запустить CODESYS.
3. В CODESYS выбрать команду **Онлайн** → **Подключение** в главном меню.
4. Перейти на вкладку «Ресурсы» организатора объектов и войти в режим «ПЛК Браузер» (PLC-Browser), подробнее о режиме см. раздел 12.

5. Выбрать команду «PLCInfo». В поле отображения реакции ПЛК на введенную команду отобразится информация о ПЛК. В строке «Binary Version» отобразится номер текущей версии встроенного ПО контроллера.

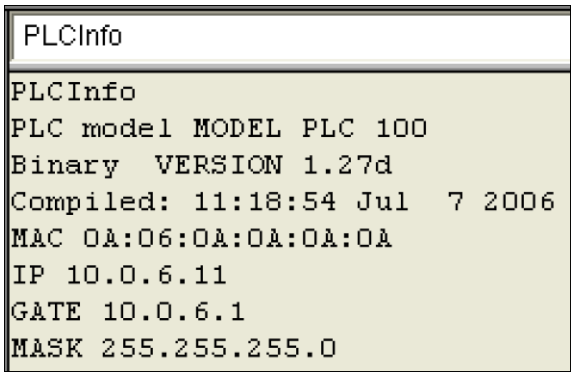


Рисунок 15.5 – Окно «ПЛК Браузер» (PLC-Browser)

15.2 Обновление встроенного ПО контроллера

Встроенное ПО контроллера может быть обновлено:

- с использованием специализированного файла обновленного встроенного ПО контроллера (например, UpdatePLC110_60.bin) и стандартных функций CODESYS;
- с помощью утилиты «Перепрошивка ПЛК <Версия>.exe» (который можно скачать на странице CODESYS V2 на сайте owen.ru). Утилиту рекомендуется использовать, если сбой в работе ПЛК привел к нарушению связи ПЛК и CODESYS.

Обновление встроенного ПО контроллера с использованием CODESYS

ВНИМАНИЕ

Если на ПЛК записана прошивка версии более ранней, чем 2.02.0, то ПЛК следует предварительно подготовить к возможности перепрограммирования ядра:

1. Снять корпус контроллера.
2. Удалить перемычку на средней плате ПЛК.
3. Установить корпус контроллера.

Для обновления встроенного ПО контроллера с использованием CODESYS следует:

1. Записать на жесткий диск ПК файл обновленного встроенного ПО контроллера (например, UpdatePLC110_60.bin).
2. Подключить питание ПЛК.
3. Соединить контроллер с ПК через любой из портов для программирования (COM-порт, порт Ethernet или USB).
4. Запустить CODESYS.
5. Выбрать команду **Онлайн** → **Подключение** в главном меню.
6. Записать в память ПЛК файл обновленного встроенного ПО контроллера выбором команды **Онлайн** → **Записать файл в ПЛК**. Требуемый файл будет опознан CODESYS автоматически.

ПРИМЕЧАНИЕ

Для сохранения в памяти ПЛК файла обновленного встроенного ПО контроллера можно также воспользоваться утилитой «PLC_IO.exe», которую можно скачать на странице CODESYS V2 на сайте owen.ru:

a. Скопировать на жесткий диск ПК утилиту«PLC_IO.exe» и файл «UpdatePLCxxx.bin»для конкретного ПЛК.

b. В командном файле PLC_IO.bat модифицировать IP-адрес или номер COM-порта, по которому утилита будет загружать файл в ПЛК.

c. Выполнить командный файл PLC_IO.bat. После запуска, если ПЛК правильно подсоединен и правильно указаны его настройки в PLC_IO.bat, утилита соединится с контроллером и запишет соответствующий файл обновления.

7. Перейти на вкладку «Ресурсы» организатора объектов и войти в режим «ПЛК Браузер» (PLC-Browser), подробнее о режиме см. раздел 12.
8. Выбрать команду «UpdateCore». В поле отображения реакции ПЛК на введенную команду появится сообщение «UpdateOK».
9. Нажать кнопку «Сброс».

Далее следует перейти к установке target-файла, см. раздел 15.3.

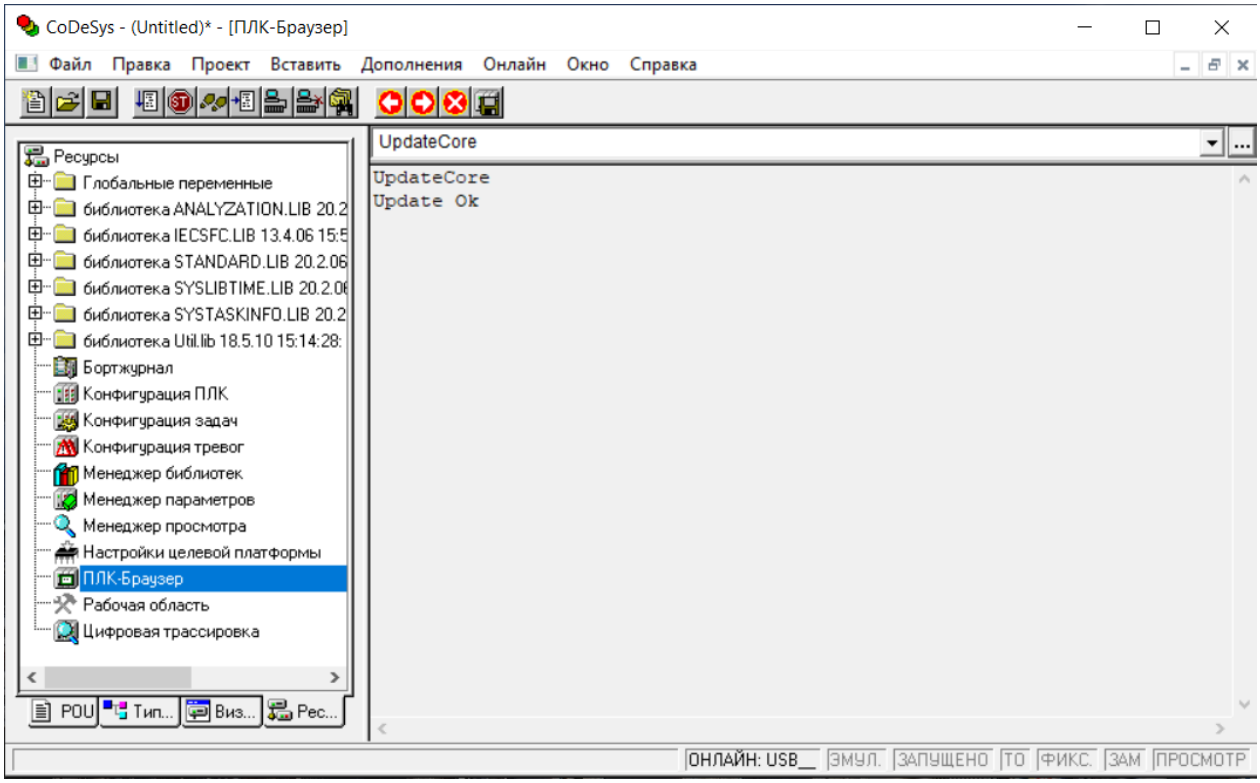


Рисунок 15.6 – Обновление встроенного ПО в окне «ПЛК Браузер» (PLC-Browser)

15.3 Обновление target-файла

Для обновления target-файла ПЛК следует:

1. Скопировать на жесткий диск ПК требуемый target-файл для конкретного ПЛК. Во время выбора target-файла следует обратить внимание на то, что имя target-файла не полностью совпадает с наименованием контроллера: в наименовании контроллера использована кириллица (например, ПЛК110), а в названии target-файла – латиница (например, PLC110). Архивы с target-файлами размещены на странице CODESYS V2 на сайте owen.ru.
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 5.

ВНИМАНИЕ
После обновления встроенного ПО контроллера ПЛК продолжает функционировать в прежнем режиме, т. е. использует настройки старого ПО. Новое встроенное ПО вступит в силу только после перезагрузки ПЛК. ПЛК следует перезагружать отключением питания с повторным включением через пять и более секунд.

16 Использование OPC-сервера

Ниже описывается подключение 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 OPCDA, 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 Microsoft Office, что позволяет получить, например, в таблице Excel набор технологических параметров, изменяющихся в реальном масштабе времени.

OPC-сервер 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. Загрузить проект в CODESYS и проверить подключение ПЛК к ПК. Если подключен, то ПЛК следует отключить выбором команды **Онлайн** → **Отключение** в главном меню.
2. Перейти в режим «Настройки целевой платформы» (Target Settings) на вкладке «Ресурсы» организатора объектов.
3. В открывшемся окне режима «Настройки целевой платформы» (Target Settings) на вкладке «Общие» (General), установить флажок переключателя «Загружать символьный файл» (Download Symbol File) и нажать кнопку «OK».

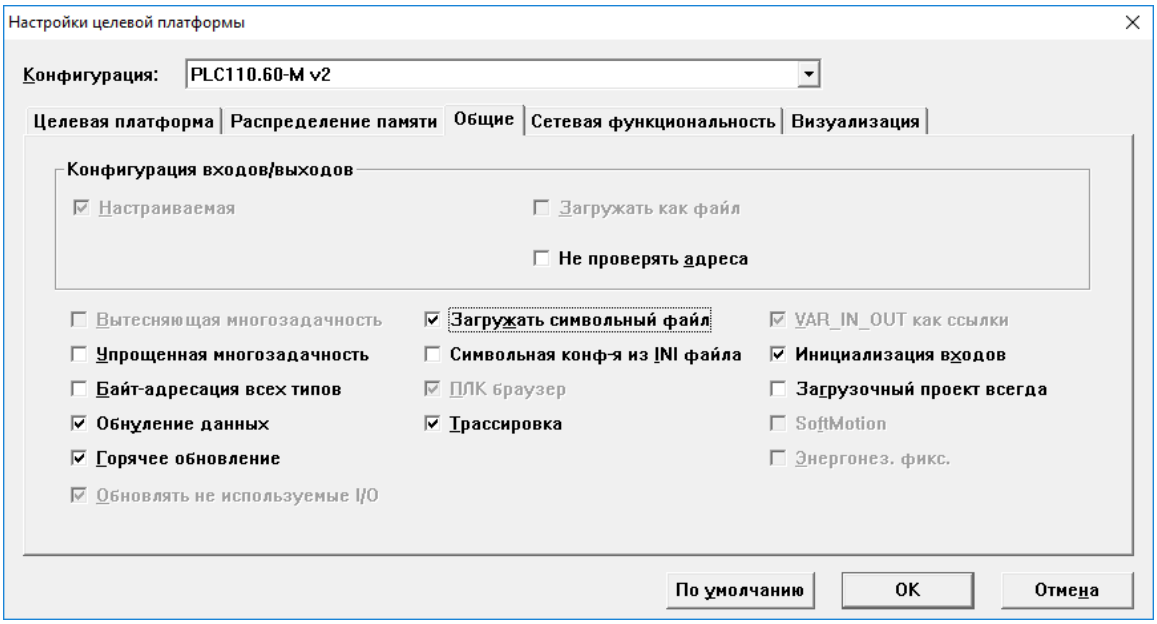


Рисунок 16.1 – Окно режима «Настройки целевой платформы» (Target Settings. Вкладка «Общие» (General)

4. Выбрать команду **Проект** → **Опции** в главном меню.

5. В открывшемся окне режима «Опции» (Options) в списке «Категория» (Category) выделить строку «Символьная конфигурация» (Symbol Configuration). В правой части окна установить флажок в поле переключателя «Создавать описания» (Dump symbol entries) и нажать кнопку «Настроить символьный файл» (Configure symbol file).

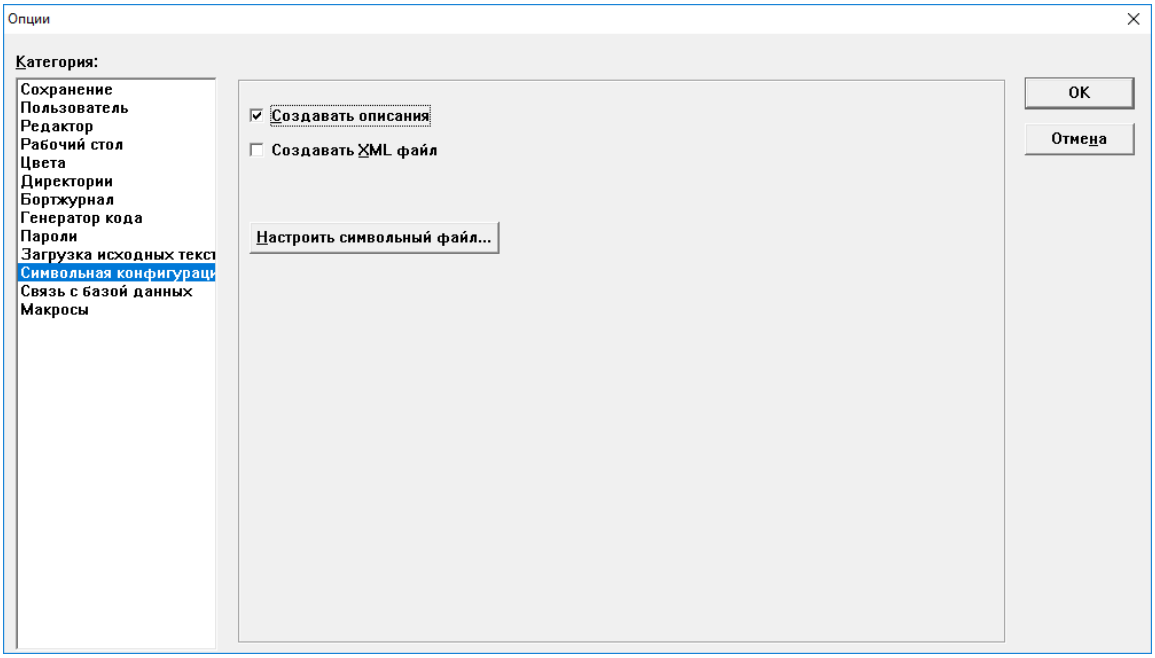


Рисунок 16.2 – Окно режима «Опции» (Options)

6. В открывшемся окне «Установка атрибутов объекта» (Set object attributes) отображается список параметров переменных проекта. В списке следует выбрать объекты проекта, из которых требуется экспортировать переменные – установить для каждого из них флажки в полях переключателей в нижней части окна. Для обеспечения экспорта переменных в пространство имен OPC-сервера следует установить флажок в поле переключателя «Экспорт переменных проекта» (Export variables of object). Если требуется изменять значения переменных, то следует установить флажок в поле переключателя «Доступ по записи» (Write access).

ВНИМАНИЕ
Во время выбора экспортируемых переменных рекомендуется выбирать только требуемые переменные, а не включать в список экспортируемых все переменные проекта.

ПРИМЕЧАНИЕ
Изменение (при необходимости) списка экспортируемых переменных описано в секции «Изменение списка экспортируемых переменных» ниже.

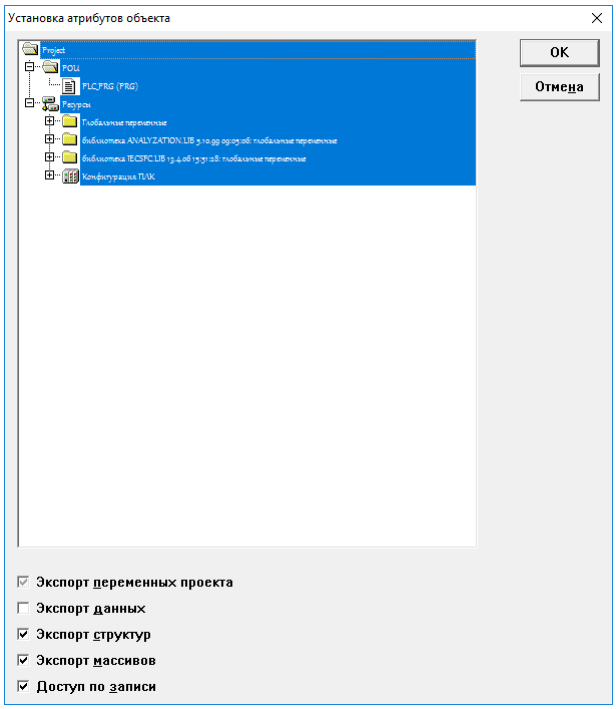


Рисунок 16.3 – Окно «Установка атрибутов объекта» (Set object attributes)

7. Нажать кнопку «OK» окна «Установка атрибутов объекта» (Set object attributes) и кнопку «OK» окна «Опции» (Options).
8. Сохранить проект командой **Файл** → **Сохранить** в главном меню или нажатием кнопки **Сохранить** на панели инструментов.
9. Перекомпилировать проект командой **Проект** → **Компилировать все** в главном меню.
10. Загрузить проект в ПЛК командой **Онлайн** → **Подключение** в главном меню.

11. Запустить ПО «CoDeSys OPC Configurator» на ПК (устанавливается совместно с CODESYS).

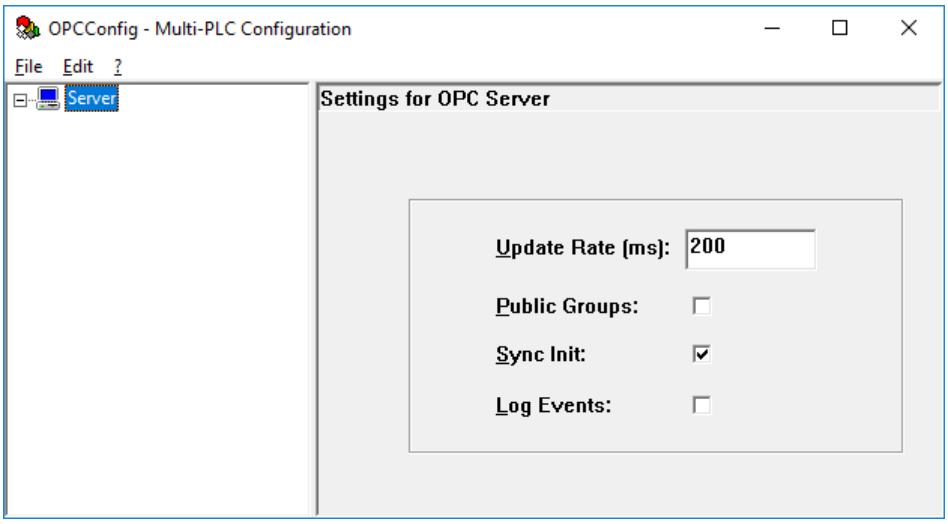


Рисунок 16.4 – Окно ПО «CoDeSys OPC Configurator»

12. В открывшемся окне «CoDeSys OPC Configurator» 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)».

13. В контекстном меню строки «Server» выбрать команду «Append PLC». В иерархический список в левой части окна добавятся строки «PLC1» и «Connection».

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

15. В открывшемся окне «Communication Parameters» установить параметры подключения ПЛК.

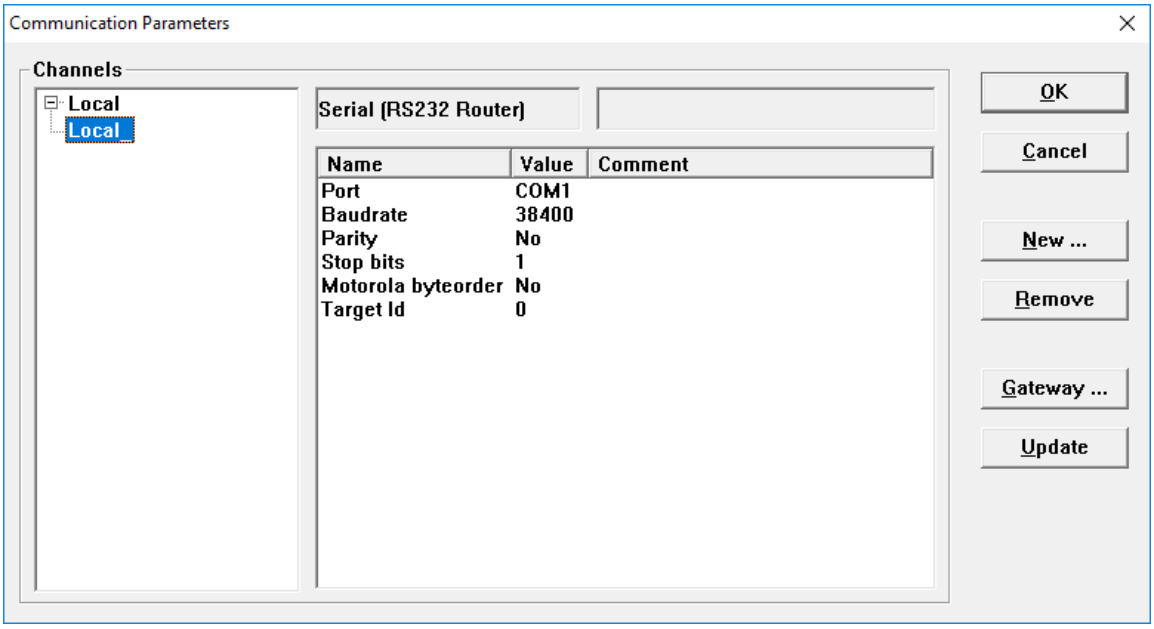


Рисунок 16.5 – Окно «Communication Parameters»

16. Нажать кнопку «OK» в окне «Communication Parameters», в открывшемся окне запроса подтверждения операции («Save changes in Multi-PLC Configuration») нажать кнопку «Да» для подтверждения настроек. OPC-сервер сконфигурирован и готов к работе под управлением SCADA-системы.

Изменение списка экспортируемых переменных

Список экспортируемых переменных сохраняется в памяти прибора, и для изменения списка следует предварительно удалить текущий список из памяти прибора.

Для удаления текущего списка экспортируемых переменных из памяти прибора следует:

1. Загрузить проект в CODESYS и проверить подключение ПЛК к ПК. Если подключен, то ПЛК следует отключить выбором команды **Онлайн** → **Отключение** в главном меню.
2. Выбрать команду **Проект** → **Опции** в главном меню.
3. В открывшемся окне режима «Опции» (Options) в списке «Категория» (Category) выделить строку «Символьная конфигурация» (Symbol Configuration). В правой части окна установить флажок в поле переключателя «Создавать описания» (Dump symbol entries) и нажать кнопку «Настроить символьный файл» (Configure symbol file), см. рисунок 16.2.
4. В открывшемся окне «Установка атрибутов объекта» (Set object attributes) снять флажок переключателя «Экспорт переменных проекта» (Export variables of object), см. рисунок 16.3. Остальные переключатели станут недоступными для редактирования.
5. Нажать кнопку «ОК» окна «Установка атрибутов объекта» (Set object attributes) и кнопку «ОК» окна «Опции» (Options).
6. Очистить сохраненные параметры проекта командой **Проект** → **Очистить все** в главном меню.
7. Перекомпилировать проект командой **Проект** → **Компилировать все** в главном меню.
8. Сохранить проект командой **Файл** → **Сохранить** в главном меню или нажатием кнопки **Сохранить** на панели инструментов.

Память прибора очищена от списка экспортируемых переменных. Процедура создания нового списка переменных описана выше.

OPC-драйверы «ОВЕН»

OPC-драйверы, разработанные компанией «ОВЕН», предназначены для подключения приборов компании «ОВЕН» к системам SCADA. Драйверы реализованы в виде двух модулей: OWEN-RS232 и OWEN-RS485, которые применяются для приборов компании «ОВЕН», поддерживающих сетевой интерфейс «токовая петля» (для преобразования в сеть RS-232 используется адаптер АС2) и сетевой интерфейс RS-485.

Для преобразования в сеть RS-232 или USB можно использовать адаптеры компании «ОВЕН» – АС3, АС3-М, АС4 или адаптеры других производителей.

Для работы могут быть использованы протоколы ОВЕН, Modbus ASCII или Modbus RTU. Перед началом работы следует задать конфигурацию приборов и режим работы порта.

К адаптеру АС2 можно подключить до 8 приборов. К одной сети RS-485 подключается до тридцати двух приборов (шлейфом, без применения повторителя).

Приборы, которые можно подключить к модулям:

- OWEN-RS232:
  - задатчик-регулятор МПР51;
  - измеритель ТРМ0 PiC;
  - измеритель УКТ38-В;
  - измеритель УКТ38-Щ4;
  - измеритель регулятор ТРМ1 PiC;
  - измеритель регулятор ТРМ10 PiC;
  - измеритель регулятор ТРМ12 PiC;
  - измеритель регулятор ТРМ5 PiC;
  - многоканальный регулятор ТРМ32;
  - многоканальный регулятор ТРМ33;
  - многоканальный регулятор ТРМ34;
  - многоканальный регулятор ТРМ38.

ПРИМЕЧАНИЕ
Начиная с версии 1.0.0.5, OPC-сервера OWEN-RS232 добавлен тег, управляющий обменом на внешней шине (флаг активности OPC-сервера). Имя тега «Status/active», тип BOOL. Запись в этот тег 1 (единицы) разрешает обмен по внешней шине, запись 0 (нуля) запрещает обмен.

- OWEN-RS485:
  - многоканальный регулятор ТРМ138;
  - универсальный двухканальный программный ПИД-регулятор ТРМ151;
  - счетчик импульсов СИ8;
  - прибор контроля положения ПКП1;
  - модуль ввода аналоговый МВА8;
  - модуль вывода управляющий МВУ8;
  - ПИД регулятор с универсальным входом ТРМ101;
  - измеритель двухканальный с универсальными входами ТРМ200;
  - измеритель-регулятор одноканальный с универсальным входом ТРМ201;
  - измеритель-регулятор двухканальный с универсальными входами ТРМ202;
  - контроллер приточной вентиляции ТРМ133.

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

1. Запустить установочный файл OwenOPC-setup.exe, который можно скачать на сайте owen.ru.
2. В открывшемся окне инсталлятора нажать кнопку «Далее». В последовательно открывающихся окнах следовать инструкциям мастера установки.

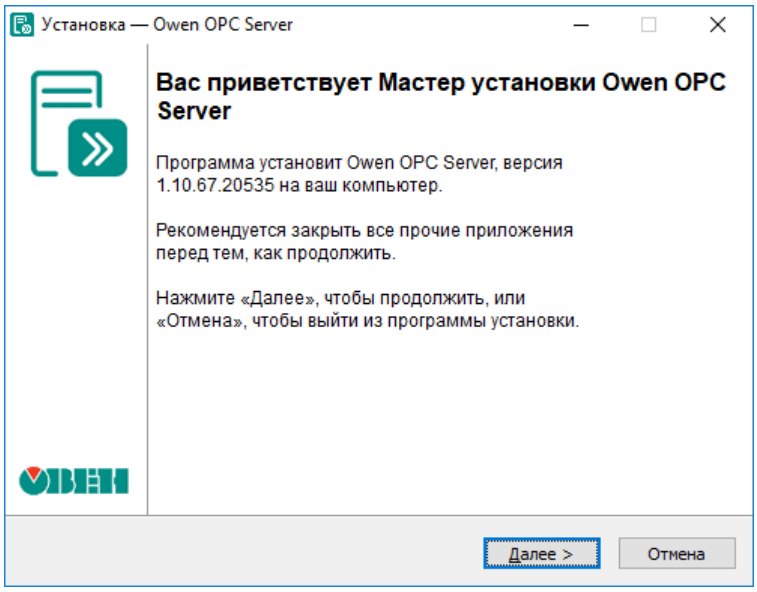


Рисунок 16.6 – Мастер установки OPC-сервера для приборов ОВЕН

Приложение А. Сообщения об ошибках в ПЛК

Модуль «ModBus (Master)»

В модуле «ModBus (Master)» используются следующие каналы для отображения статуса Мастера сети и возникающих ошибок:

- **Last Address** – адрес последнего опрошенного мастером устройства (адрес последовательного устройства или IP-адрес, в зависимости от режима работы универсального устройства odBus);
- **Last error** – содержит код ошибки из таблицы ниже.

Таблица А.1 – Ошибки работы модуля «ModBus (Master)»

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
OK	0x0000	0	Нет ошибок
NO_DEVICE	0x0051	81	Превышен тайм-аут ожидания ответа
NO_SOCKET	0x0054	84	Нет свободного сокета для устройства TCP/IP
SOCKET_ERROR	0x0055	85	Ошибка при приеме/передаче по сети TCP/IP

Модуль «Owen (Master)»

В модуле «Owen (Master)» используются следующие каналы для отображения статуса Мастера сети и возникающих ошибок:

- **Last error** – содержит код ошибки (см. таблицу 1);
- **Last Address** – содержит адрес последнего опрошенного Мастером сети устройства;
- **Last HASH** – содержит hash-код переменной, которая была опрошена последней.

Таблица А.1 – Ошибки работы модуля «Owen (Master)»

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
OK	0x0000	0	Нет ошибок
NO_DEVICE	0x0051	81	Превышен тайм-аут ожидания ответа
N_ERR	0x0057	87	В ответе устройства содержится hash-код ошибки **n.Err (0x0233)**. Уточнение ошибки содержится в старшем байте кода ошибки (см. таблицу 2)
BAD_HASH	0x4000	16384	Hash-код в ответе не соответствует ожидаемому*
BAD_ADDRESS	0x8000	32768	Адрес в ответе не соответствует ожидаемому*

ПРИМЕЧАНИЕ
* Коды ошибок могут быть вызваны совместной работой нескольких мастеров в одной сети.

Таблица А.2 – Коды ошибок приборов в сети ОВЕН

Краткое наименование	Код ошибки в старшем байте		Причины ошибок
	(Hex)	(Dec)	
Определение констант ошибок приема			
OK	0	0	Безошибочный прием кадра
Ошибки записи параметров и атрибутов функцией modific			
PDOT	2	2	Задано положение точки, превышающее 3
EROM	3	3	Попытка модификации ROM-параметра
ESTR	4	4	Не целое число при записи индекса строки или времени

Продолжение таблицы А.2

Краткое наименование	Код ошибки в старшем байте		Причины ошибок
	(Hex)	(Dec)	
EDOT	5	5	Неверно задано положение точки (при фиксированной точке)
ERNG	6	6	Значение мантиссы превышает ограничения дескриптора
Ошибки записи атрибутов функциями modAllPermis() и modEditPermis()			
EOWNER	7	7	Несанкционированная попытка редактирования атрибутов (попытка изменить атрибут пользователем, не являющимся хозяином параметра).
EPERM	8	8	У запрошенного параметра отсутствуют признаки
Стандартные ошибки, присущие протоколу обмена			
AFE	0x21	33	Аппаратная ошибка кадрирования
B8E	0x22	34	Ошибка в восьмом бите посылки
B9E	0x23	35	Ошибка в девятом бите посылки
SBE	0x24	36	Ошибка приема стоп-байта (стоп пришел не вовремя)
OVB	0x25	37	Ошибка переполнения буфера
ERS	0x26	38	Принят недопустимый символ
CRCE	0x27	39	Неверная контрольная сумма кадра
EDESC	0x28	40	Не найден дескриптор
NFNC	0x29	41	Не найдена сетевая функция, хотя дескриптор найден. В нормально функционирующем приборе эта ошибка встречаться не должна
Стандартные ошибки, общие для всех модулей			
EDGT	0x30	48	Мантисса двоично-десятичного параметра содержит ошибку
SZE	0x31	49	Размер поля данных не соответствует ожидаемому
EASK	0x32	50	Значение бита запроса не соответствует ожидаемому
EACC	0x33	51	Редактирование параметра запрещено индивидуальным атрибутом
IDXOVF	0x34	52	Недопустимо большой линейный индекс
IDXLIM	0x35	53	Индекс параметра превышает ограничитель индекса
EXTROM	0x36	54	Индекс параметра превышает ограничитель индекса
RESERVED	0x37	55	Данный код не используется
«Запись запрещена групповым атрибутом уровня»			
LEVGRATT	0x38	56	Запрещающий групповой атрибут находится на уровне 0 (в корне)
LEVGRATT1	0x39	57	Запрещающий групповой атрибут находится на уровне 1
LEVGRATT2	0x3A	58	Запрещающий групповой атрибут находится на уровне 2
LEVGRATT3	0x3B	59	Запрещающий групповой атрибут находится на уровне 3
LEVGRATT4	0x3C	60	Запрещающий групповой атрибут находится на уровне 4
LEVGRATT5	0x3D	61	Запрещающий групповой атрибут находится на уровне 5
LEVGRATT6	0x3E	62	Запрещающий групповой атрибут находится на уровне 6
LEVGRATT7	0x3F	63	Запрещающий групповой атрибут находится на уровне 7
Состояния COMMON-сегмента			
__LOCKSEG	0x41	65	Выполняется другая задача (сегмент COMMON занят)
__FREESEG	0x42	66	Задача еще не запущена (сегмент COMMON свободен)

Продолжение таблицы А.2

Краткое наименование	Код ошибки в старшем байте		Причины ошибок
	(Hex)	(Dec)	
__READYSEG	0x43	67	Запрошенная задача уже выполняется
__DEBUGSEG	0x44	68	Программе неизвестна запрошенная функция
__NOWHATCOM	0x45	69	В программе стоит заглушка функции WhatCOMState()
__NORUNCOM	0x46	70	В программе стоит заглушка функции RunCOMTask()
	0x47	71	Недопустимое сочетание значений параметров (изменение параметра было запрещено функцией Valid)
	0x48	72	Ошибка при чтении EEPROM
Ошибки при редактировании графиков			
	0x49	73	Нарушена упорядоченность узлов X по возрастанию
	0x4A	74	Попытка записи X при ненулевом числе узлов графика
	0x4B	75	Ошибка выполнения функции PrevWriteActions()
Ошибки мостов и ретрансляторов			
GATE _OVR	0x50		Переполнение буфера моста или ретранслятора
GATE_DERR	0x51	81	Превышение тайм-аута ответа, потеря пакета в дочерней сети (сети, в которую ретранслируется пакет)
GATE_NONET	0x52	82	Запрошенная дочерняя подсеть не доступна (в случае ретрансляции в одну из нескольких дочерних подсетей)
GATE_MERR	0x53	83	Ответ из дочерней сети не может быть ретранслирован в материнскую сеть

ПРИМЕЧАНИЕ
Коды ошибок приборов в сети ОВЕН приведены в соответствие с описанием протокола приборов ПО ОВЕН по RS-485.

Модуль «DCON (Master)»

В модуле «DCON (Master)» используется канал **Last error** для отображения статуса Мастера и возникающих ошибок, который содержит код ошибки из таблицы ниже.

Таблица А.1 – Ошибки работы модуля «DCON (Master)»

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
OK	0x0000	0	Нет ошибок
NO_DEVICE	0x0051	81	Превышен тайм-аут ожидания ответа

Информация о работе каждого отдельного устройства DCON выводится в поле «Status» универсального устройства DCON. Коды ошибок приведены в таблице ниже.

Таблица А.2 – Ошибки работы универсального устройства DCON

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
NOT_INITIALIZED	0x0000	0	Модуль универсального устройства DCON не был проинициализирован корректно
REQUEST	0x0001	1	Послан запрос к устройству
BAD_REQUEST_ FORMAT	0x0041	65	Неправильный формат строки запроса

Продолжение таблицы А.2

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
BAD_REQUEST_DATA	0x0081	129	Данные для запроса не соответствуют по формату строке запроса
NO_DEVICE	0x0051	81	Превышен тайм-аут ожидания ответа
UNINDENTIFIED_ ANSVER	0x0021	33	Ответ не распознан
OK_ANSVER	0x0003	3	Пришел ответ, соответствующий строке формата для случая положительного ответа
OK_ANSVER_BAD_ FORMAT	0x0043	67	Неправильный формат строки разбора положительного ответа
OK_ANSVER_BAD_ DATA	0x0083	131	Данные для разбора положительного ответа не соответствуют по формату строке разбора
NEG_ANSVER	0x0023	35	Пришел ответ, соответствующий строке формата для негативного ответа
NEG_ANSVER_BAD_ FORMAT	0x0063	99	Неправильный формат строки разбора негативного ответа
NEG_ANSVER_BAD_ DATA	0x00A3	163	Данные для разбора негативного ответа не соответствуют по формату строке разбора

ПРИМЕЧАНИЕ
Переменная **Status** предназначена также для управления работой мастера DCON при настройке универсального устройства DCON в режиме **«Опрос по команде» (Work mode= «By command»)**. Для однократного запуска опроса следует записать в переменную значение **0xFF**.

Подмодуль «Modem»

В подмодуле «Modem» используется канал **Modem fault** для отображения возникающих ошибок, который содержит код ошибки из таблицы ниже.

Таблица А.1 – Ошибки работы подмодуля «Modem»

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
OK	0x0000	0	Нет ошибок, модем исправен
MODEM_FAULT	0x0001	1	Подмодуль «Modem» зафиксировал отказ подключенного модема или его отсутствие

Модуль «Archiver»

В модуле «Archiver» используется канал **Status** для отображения возникающих ошибок, который содержит код ошибки из таблицы ниже.

Таблица А.1 – Ошибки работы модуля «Archiver»

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
STANDBY	0x0000	0	Модуль «Archiver» остановлен
RUN	0x0001	1	Модуль «Archiver» запущен и работает без ошибок
DEVICE_ERROR	0x0002	2	Ошибка при записи в устройство
NO_DEVICE	0x0004	4	Устройство вывода отсутствует или неправильно проинициализировано

ПРИМЕЧАНИЕ
Во время записи команды в переменную **Status** архивирование может быть запущено или остановлено. Коды команд даны в таблице 2.

Таблица А.2 – Коды команд управления модулем «Archiver»

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
ARCHIVING_STARTED	0x00FF	255	Запуск архивирования
ARCHIVING_STOPED	0x00FE	254	Остановка архивирования

Подмодуль архивирования информации в файл

В подмодуле архивирования информации в файл в поле **File Status** выводится информация о работе подмодуля. Коды ошибок приведены в таблице ниже.

Таблица А.1 – Ошибки работы модуля архивирования информации в файл

Краткое наименование	Код ошибки		Причины ошибок
	(Hex)	(Dec)	
OK	0x0000	0	Модуль функционирует нормально
STOPED	0x0001	1	Вывод в файл запрещен
CANT_OPEN	0x0002	2	Невозможно открыть файл
TOO_LARGE	0x0003	3	Размер записи превышает размер файла. Следует увеличить файл или разбить данные на несколько файлов
FILE_FULL	0x0004	4	Файл заполнен. Перезапись или сдвиг запрещены
GENERAL_ERROR	0x0005	5	Ошибка инициализации

Приложение Б. Примеры настройки опроса для модуля «DCON (Master)»

Ниже представлены примеры настройки модуля «DCON (Master)» для опроса устройств ввода/вывода.

Опрос модулей аналоговых входов IPC-7033

Пример опроса первых трех входов нескольких модулей аналоговых входов IPC-7033.

Описание формата обмена с модулем IPC-7033.

Формат запроса:
`#AA[CRC][CR]`

где # – разделитель;

AA – адрес прибора;

CRC – контрольная сумма;

CR – перевод строки.

Формат ответа:
`>+0255.12+013.45+150.11[CRC][CR]`

где > – разделитель в случае положительного ответа;

+025.12 – данные одного канала (5 чисел + знак + точка), всего семь символов;

CRC – контрольная сумма;

CR – перевод строки.

Окно конфигурирования модуля «DCON (Master)» с подключенным к нему устройством **Universal DCON device**, настроенным для последовательного опроса нескольких модулей IPC-7033 с различными адресами, представлено на рисунке ниже.

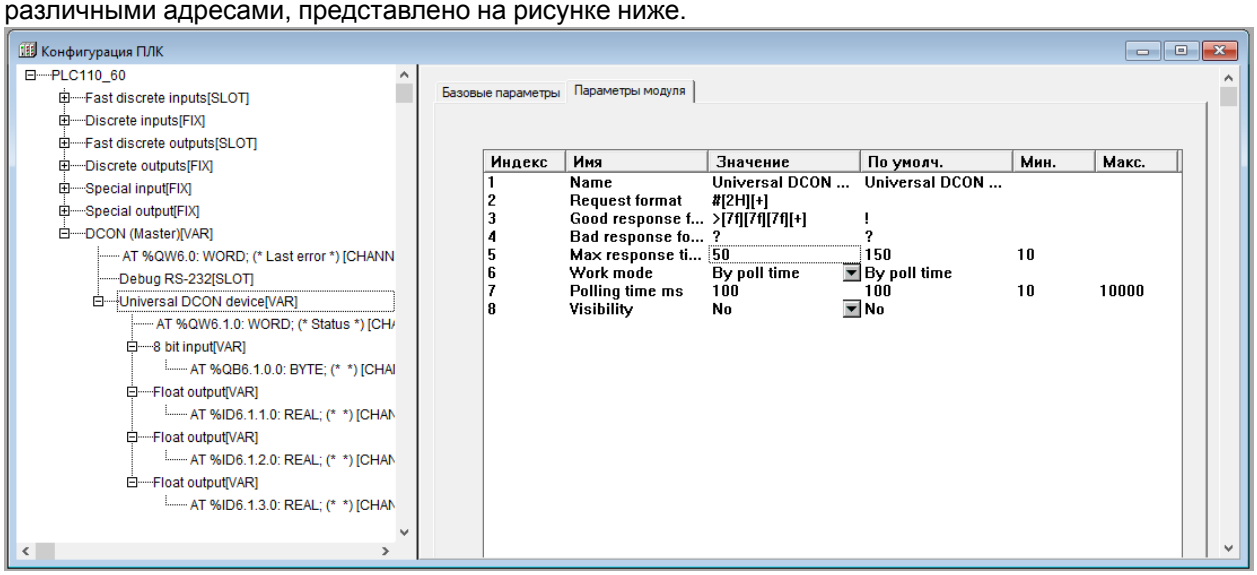


Рисунок Б.1 – Параметры подмодуля «Universal DCON device» модуля «DCON (Master)» для опроса входов модулей IPC-7033

Для задания адреса опрашиваемого прибора используется восьмибитовая входная переменная **Address**. Значения со входов опрашиваемого модуля IPC-7033 отображаются в трех выходных переменных типа **Float (REAL)**.

Настройки подмодуля «Universal DCON device»:

- Настройки подмодуля «Universal DCON device»:
- **Request format** – формат строки запроса – **#[2h][+]**,
  - где **#** – символ разделителя команды опроса входов;
  - **[2h]** – спецкоманда, указывающая, что в это место запроса подставляется шестнадцатеричный двухсимвольный адрес, значение которого должно быть взято из входной переменной;

– **[+]** – спецкоманда подсчета и добавления в конец запроса контрольной суммы «по модулю 256».

ВНИМАНИЕ
Символ возврата каретки вставляется автоматически.

- **Good response format** – формат положительного ответа – **>[7f][7f][7f][+]**,
  - где **>** – символ разделителя в случае положительного ответа;
  - **[7f]** – спецкоманда, указывающая на то, что семь символов ответа должны быть преобразованы в число с плавающей точкой и результат преобразования должен быть помещен в первую выходную переменную, которая имеет формат **Float**;

    ВНИМАНИЕ
    Для следующих спецкоманд **[7f]** применяется то же правило преобразования, но результаты помещаются во вторую и третью выходные переменные, соответственно.

  - **[+]** – спецкоманда, указывающая на то, что должна быть проанализирована правильность контрольной суммы в принятой посылке. Результат записывается в переменную **Status**;
  - **CR** – перевод строки.
- **Bad response format** – формат отрицательного ответа – **?**,
  - где **?** – начальный символ строки отрицательного ответа. В рассматриваемом случае отрицательный ответ не содержит значащей информации, для его идентификации достаточно одного первого символа.
- **Max response timeout** – максимальное время ожидания ответа – 50 мс. Задается в соответствии с рекомендациями производителя прибора.
- **Work mode** – режим работы – **by change value** (по изменению значения одной из входных переменных). Режим позволяет генерировать запросы при изменении адреса опрашиваемого прибора. Для генерации одного запроса следует записать значение, отличающееся от текущего, во входную переменную **Adress**. После этого по значению переменной **Status** определяется окончание обмена данными с опрашиваемым прибором и корректность данных в выходных переменных.

  ПРИМЕЧАНИЕ
  Чтение переменной **Status** и ее анализ должны производиться на следующем цикле работы ПЛК после записи нового адреса. Остальные параметры в данном режиме работы не существенны.

Установка выходного значения модуля IPC-7021

Пример установки выходного значения модуля аналогового вывода IPC-7021 с периодичностью 1 секунда и при необходимости изменения значения следующим образом – записью одного выходного значения у модуля IPC-7021 с шестнадцатеричным адресом 18.

Описание формата обмена с модулем IPC-7021.

Формат запроса:
`#AA(данные)[CRC][CR]`

где **#** – разделитель;

AA – адрес прибора;

(данные) – выходное значение (5 чисел + знак + точка), всего семь символов;

CRC – контрольная сумма;

CR – перевод строки.

Формат ответа:
`![CRC][CR]`

где *!* – разделитель в случае положительного ответа;

CRC – контрольная сумма;

CR – перевод строки.

Параметры модуля «DCON (Master)» с подключенным к нему подмодулем «Universal DCON device», настроенным для периодической записи выходных значений в модуль IPC-7021, изображены на рисунке ниже.

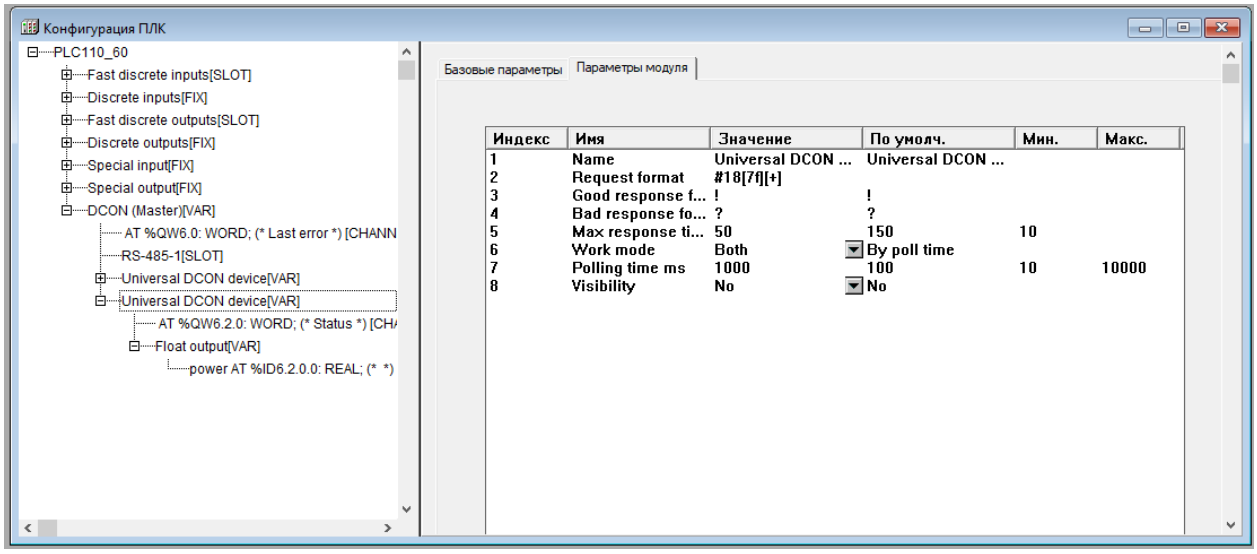


Рисунок Б.2 – Параметры подмодуля «Universal DCON device» модуля «DCON (Master)» для периодической записи выходных значений в модуль IPC-7021

Данные, посылаемые в модуль IPC 7021, задаются во входной переменной **power** типа **Float (REAL)** подмодуля «Universal DCON device».

Настройки подмодуля «Universal DCON device»:

- Настройки подмодуля «Universal DCON device»:
- **Request format** – формат строки запроса – **#18[7f][+]**,
  - где **#** – символ разделителя команды опроса входов;
  - **18** – адрес прибора в шестнадцатеричном формате (**для букв используется верхний регистр**);
  - **[7f]** – спецкоманда, указывающая на то, что семь символов запроса должны быть сформированы в виде числа в формате **[знак]число..число.число..число**. Данные должны быть взяты из входной переменной, которая должна иметь формат **float**;
  - **[+]** – спецкоманда подсчета и добавления в конец запроса контрольной суммы «по модулю 256».

    ВНИМАНИЕ
    Символ возврата каретки вставляется автоматически.
- **Good response format** – формат положительного ответа – **!**,
  - где **!** – начальный символ строки положительного ответа. В рассматриваемом случае положительный ответ не содержит значащей информации, для его идентификации достаточно одного первого символа.
- **Bad response format** – формат отрицательного ответа – **?**,
  - где **?** – начальный символ строки отрицательного ответа. В рассматриваемом случае отрицательный ответ не содержит значащей информации, для его идентификации достаточно одного первого символа.
- **Max response timeout** – максимальное время ожидания ответа – 50 мс. Задается в соответствии с рекомендациями производителя прибора.
- **Work mode** – режим работы – Both (по времени опроса и смене значения одной из входных переменных). Этот режим позволяет генерировать запросы по таймеру (параметр Polling time) и при изменении значения входной переменной модуля;
- **Polling time** – время опроса – 1000 мс. Задает период записи значения в модуль IPC-7021.

Приложение В. Примеры настройки опроса переменных по протоколу ОВЕН

Ниже представлены примеры настройки опроса переменных модуля «Owen (Master)» для некоторых наиболее часто встречающихся случаев применения ПЛК. Примеры могут быть использованы и при настройке модулей «Owen (Slave)» и «Owen (Spy)».

ПРИМЕЧАНИЕ
Во время настройки модулей «Owen (Spy)» и «Owen (Slave)» не используется параметр «Polling Time» (период опроса). При настройке модуля «Owen (Slave)» также не используются параметры «Address Type» (тип адреса) и «Address» (адрес).

Пример

Таблица В.1 – Описание переменной

Характеристика	Значение
Прибор	Все приборы с аналоговыми входами
Тип переменной	Число с плавающей точкой и циклическим временем
Имя параметра	Read
Описание	Значение с измерителя
Диапазон	В соответствии с диапазоном аналогового входа
Диапазон индекса	0…6

Таблица В.2 – Настройки модуля Owen (Master)

Характеристика	Значение
Тип переменной	Float variable + time
Направление опроса	Чтение
Address Length	8/11 бит в зависимости от настроек сети
Address	Адрес прибора + номер канала с нуля
Hash name	Read
Index	Индекс содержится в адресе прибора
Use a index?	No
Float type	Float
Precision	Не используется
Polling time	В мс, в зависимости от необходимого периода опроса
Вставленные подмодули	Отсутствуют

Пример

Таблица В.1 – Описание переменной

Характеристика	Значение
Прибор	ТРМ133, МВА8, ТРМ151, ТРМ148
Тип переменной	Тип данных с числом с фиксированной точкой
Имя параметра	itrL
Описание	Период опроса датчика
Диапазон	0,1…30,0
Диапазон индекса	0…6

Таблица В.2 – Настройки модуля Owen (Master)

Характеристика	Значение
Тип переменной	Float variable
Направление опроса	Чтение/Запись (в зависимости от потребностей)
Address Length	8/11 бит в зависимости от настроек сети
Address	Адрес прибора
Hash name	itrL
Index	0–6 (в зависимости от номера канала измерения)
Use a index?	Yes
Float type	Fix point binary
Precision	1
Polling time	В мс, в зависимости от необходимого периода опроса
Вставленные подмодули	Отсутствуют

Пример

Таблица В.1 – Описание переменной

Характеристика	Значение
Прибор	ТРМ133, МВА8, ТРМ151, ТРМ148
Тип переменной	Тип данных со знаковым/беззнаковым целым значением
Имя параметра	in-t
Описание	Тип датчика
Диапазон	Не ограничен
Диапазон индекса	0…6

Таблица В.2 – Настройки модуля Owen (Master)

Характеристика	Значение
Тип переменной	Unsigned variable
Направление опроса	Чтение/Запись (в зависимости от потребностей)
Address Length	8/11 бит в зависимости от настроек сети
Address	Адрес прибора
Hash name	in-t
Index	0…6 (в зависимости от номера канала измерения)
Use a index?	Yes
Polling time	В мс, в зависимости от необходимого периода опроса
Вставленные подмодули	
1	8 bits

Пример

Таблица В.1 – Описание переменной

Характеристика	Значение
Прибор	Любой прибор ОВЕН
Тип переменной	Тип данных с текстовой информацией
Имя параметра	Dev
Описание	Название прибора
Диапазон	Строка
Диапазон индекса	Нет индекса

Таблица В.2 – Настройки модуля Owen (Master)

Характеристика	Значение
Тип переменной	String variable
Направление опроса	Чтение (константное значение)
Address Length	8/11 бит в зависимости от настроек сети
Address	Адрес прибора
Hash name	dev
Index	Не используется
Use a index?	No
Polling time	В мс, в зависимости от необходимого периода опроса
Вставленные подмодули	Отсутствуют

Пример

Таблица В.1 – Описание переменной

Характеристика	Значение
Прибор	ТРМ133
Тип переменной	Тип данных с информацией о времени
Имя параметра	t.val
Описание	Время полного хода трехпозиционного ИМ, мин:с
Диапазон	1…999
Диапазон индекса	0…7

Таблица В.2 – Настройки модуля Owen (Master)

Характеристика	Значение
Тип переменной	Time variable
Направление опроса	Чтение/Запись (в зависимости от потребностей)
Address Length	8/11 бит в зависимости от настроек сети
Address	Адрес прибора
Hash name	t.val
Index	0–7 (в зависимости от номера канала)
Use a index?	Yes
Polling time	Seconds
Вставленные подмодули	В мс, в зависимости от необходимого периода опроса
Вставленные подмодули	
1	Seconds
2	Minutes

Приложение Г. Перенос проекта между несовместимыми версиями встроенного ПО контроллера

Если проект создан для контроллера со встроенным ПО, версия которого несовместима с имеющейся на доступном контроллере, можно вручную, с помощью двух открытых экземпляров CODESYS, преобразовать проект к нужной версии встроенного ПО. Фактически, такое преобразование есть поэтапное копирование различных частей проекта из одного его экземпляра, для несовместимой версии, в другой, вновь созданный, где target-файл выбран верно.

Во время конвертирования следует учесть, что причины несовместимости могут быть следующими:

- различие имен и допустимых диапазонов значений переменных конфигурации, одинаковых по своему назначению;
- различия в обозначении адресов соответствующих входов и выходов;
- разница в наличии необходимых входов и выходов, в том числе служебного назначения;
- разница в объемах доступной памяти.

Перечисленные причины влекут за собой возможную необходимость исправления исходных текстов POU внутри нового проекта, или даже нереализуемость некоторых функций (в случае отсутствия в конфигурации). Проект, который нужно перенести, далее будет называться «*старый проект*», а проект, в который происходит копирование – «*новый проект*».

Для переноса проекта между несовместимыми версиями встроенного ПО контроллера следует:

1. Перед переносом установить target-файлы для тех версий встроенного ПО, для которых существует старый вариант проекта и будет сформирован его новый вариант.
2. Открыть два экземпляра CODESYS. В обоих открыть старую версию проекта, одно окно далее будет использоваться как источник данных по проекту («старый проект»), второй – как приемник («новый проект»). Рекомендуется в окне с проектом-приемником сохранить проект под другим именем.
3. В новом проекте изменить конфигурацию оборудования, то есть выбрать target-файл, соответствующий контроллеру и версии прошивки, которые планируется использовать далее. Target-файл выбирается на вкладке «Ресурсы» в режиме «Настройки целевой платформы», нажатием на кнопку селектора «Конфигурация».

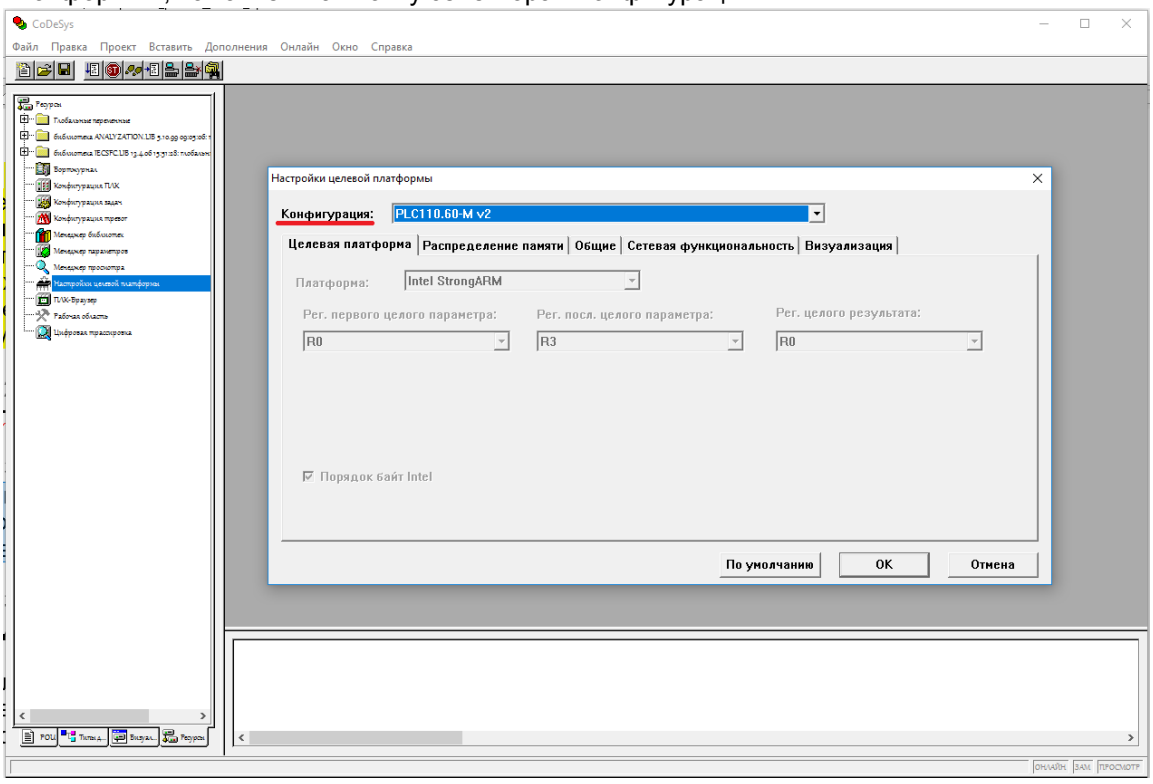


Рисунок Г.1 – Смена target-файла в режиме «Настройки целевой платформы»

4. В новом проекте выбрать **Дополнения** → **Стандартная конфигурация** в главном меню и подтвердить сброс конфигурации проекта до стандартной.

5. Последовательно, сохраняя порядок, создать в новом проекте все те модули, что были в старом проекте. Модули следует добавлять в том, порядке, в котором они перечислены сверху вниз в старом проекте.
6. Задать модулям параметры в соответствии со старым проектом, с учетом различий в требуемых значениях временных и иных параметров в новой версии конфигурации. Во время конфигурирования следует учитывать, что некоторые параметры могут отсутствовать, а у других параметров может смениться способ их представления или диапазоны значений, в частности по-другому могут задаваться временные параметры модуля ШИМ.

   ПРИМЕЧАНИЕ
   Например, могут отсутствовать некоторые параметры портов передачи данных RS-232, RS-485, или дискретных входов и выходов, например, в некоторых случаях, такой параметр, как Visibility.

   ПРИМЕЧАНИЕ
   К параметрам, у которых может измениться способ их представления или диапазоны значений, относится постоянная времени фильтра быстрых входов-выходов. Постоянная времени фильтра может уменьшится в связи с увеличением быстродействия фильтра.
7. Задать каналам ввода-вывода имена в соответствии со старым проектом. Возможность задавать каналам ввода-вывода имена и использовать их в разработанной программе, избегая таким образом использования адресов, существенно облегчает перенос проекта между платформами. Если в старом проекте использовались имена каналов, то следует задать их в настройках объектов точно в том же виде и на тех же местах, как это было в старом проекте.
8. Проверить распределение памяти входов-выходов. Конфигурация входов и выходов могла измениться. Новая версия контроллера ПЛК110 [M02] (target-файл версии 3.04) имеет следующие изменения:
   - отсутствуют некоторые переменные в модуле «Статистика»: нет внутренней температуры контроллера и переменной оставшегося времени работы при работе от батареи;
   - не требуется создание подмодулей в случае использования переменных типа unsigned;
   - все адреса Modbus для устройств master и slave обозначаются как **%Q**.

   Если в пользовательской программе использовались переменные внутренней температуры контроллера и оставшегося времени работы при работе от батареи, подпрограммы, их использующие, необходимо реализовать другим способом или не задействовать при компиляции проекта.

   Подпрограммы, которые используют переменные типа unsigned, и в которых имеется обращение к переменным по адресам, могут нуждаться в модификации в связи с тем, что вид адресов переменных типа unsigned изменится. Также изменится вид адресов входов и выходов в модулях Modbus. Если данные адреса присутствуют в программе в чистом виде (например, в блоке объявления переменных, хотя могут и во всем тексте программы), то необходимо произвести замену адреса там, где это необходимо.

   Если в конфигурации старой версии проекта входам и выходам присвоены имена и в тексте программы обращение к входам и выходам осуществляется только по присвоенным именам, то достаточно будет изменить конфигурацию входов и выходов контроллера, введя соответствующие имена соответствующим ячейкам памяти, связанным с вводом-выводом.
9. Если требуется, то скорректировать код программы, если используются значения из модуля статистики или модуля ШИМ (см. выше). Изменения могут коснуться не только конфигурации ПЛК, но и текста программы. Поэтому, при наличии обращениий к параметрам модуля ШИМ или модуля статистики в тексте конвертируемой программы следует проконтролировать каждый из фрагментов кода, содержащий данные обращения и модифицировать код, с целью сохранения его работоспособности, либо исключить его из проекта с целью сохранения работоспособности остального проекта.
10. Скомпилировать программу, настроить канал связи и загрузить проект в ПЛК.

Россия, 111024, Москва, 2-я ул. Энтузиастов, д. 5, корп. 5
тел.: +7 (495) 641-11-56, факс: (495) 728-41-45
тех. поддержка 24/7: 8-800-775-63-83, support@owen.ru
отдел продаж: sales@owen.ru
www.owen.ru
рег.:1-RU-75044-1.33