

2016

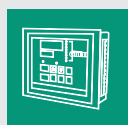


СПК

FAQ

Версия: 1.2

Дата: 23.03.2016



Оглавление

1. Цель документа	6
2. Восемь самых частых вопросов.....	7
3. Начало работы	8
3.1. Что такое СПК?	8
3.2. Как программируется СПК?	8
3.3. Как перезагрузить СПК?	8
3.4. Как выполнить монтаж СПК?	8
4. Вопросы по прошивке	9
4.1. Что такое прошивка?	9
4.2. Как узнать версию прошивки?	9
4.3. В каких случаях следует обновлять прошивку СПК?	10
4.4. Какие могут быть последствия при перепрошивке? Можно ли будет «откатиться» на старую версию прошивки?	10
4.5. Где взять новые прошивки?	10
4.6. Как перепрошить СПК?.....	10
5. Конфигуратор и сервисное меню.....	11
5.1. Для чего нужен конфигуратор?	11
5.2. Как попасть в конфигуратор?	12
5.3. Какой пароль у конфигуратора?.....	12
5.4. Как попасть в сервисное меню?	12
5.5. Что можно сделать из сервисного меню?.....	13
5.6. Как обновить прошивку с USB flash?.....	13
5.7. Зачем нужна калибровка дисплея?	13
5.8. Как загрузить проект с USB flash?.....	14
6. Вопросы по CODESYS	15
6.1. Что такое CODESYS?	15
6.2. Где взять CODESYS?.....	15
6.3. Какую версию CODESYS следует устанавливать?.....	15
6.4. Как установить CODESYS?.....	15
6.5. Что такое target-файл?	16
6.6. Где взять target-файл?.....	16
6.7. Как установить target-файл и подключить его к проекту?	16

6.8. Как связаны версии прошивки, CODESYS и target-файла?	18
6.9. Как перенести проект из одной версии CODESYS в другую?	18
6.10. Как перенести проект с одного компьютера на другой?	18
6.11. Что такое библиотека?	18
6.12. Где взять дополнительные библиотеки?	18
6.13. Как установить библиотеку и подключить ее к проекту?	19
6.14. При открытии проекта возникает ошибка «Библиотека <...> отсутствует в проекте», что делать?	20
6.15. Что такое CODESYS HMI, CODESYS Application Composer, CODESYS SoftMotion?	20
7. Вопросы по подключению СПК к ПК	22
7.1. Как подключиться к СПК из CODESYS?	22
7.2. Как узнать IP-адрес СПК?	24
7.3. Как поменять IP СПК?	24
7.4. CODESYS не видит СПК, что делать?	25
7.5. Как подключить СПК к локальной сети?	26
8. Вопросы по созданию проекта CODESYS	27
8.1. Как сменить язык CODESYS?	27
8.2. У меня не хватает определенных компонентов и настроек, что делать?	28
8.3. Что делать, если интерфейс «рассыпался»?	29
8.4. Как включить сетку в Редакторе визуализации?	29
8.5. Как настроить размер экрана визуализации?	30
8.6. Как отобразить значение параметра в визуализации?	31
8.7. Как изменить значение параметра в визуализации с дисплея контроллера?	32
8.8. Как переключать экраны визуализации?	34
8.9. Почему русский текст на визуализации отображается «закорючками»?	36
8.10. Как добавить в проект экранную клавиатуру с кириллицей?	36
8.11. Как настроить выполнение действия по нажатию на элемент?	37
8.12. Можно ли добавить свой графический примитив в проект?	38
8.13. Как добавить графический файл в проект?	41
8.14. Как реализовать анимацию?	44
8.15. Как использовать в визуализации многострочный текст?	46
8.16. Как создать текст с вертикальной ориентацией?	46
8.17. Почему в Таблице тревог не отображается история?	46
8.18. Как записывать историю тревог на flash- или SD-карту памяти?	47
8.19. Сколько переменных можно отображать на тренде?	47

8.20. Какова частота записи переменных тренда?	47
8.21. Как просмотреть историю тренда?	48
8.21. Как записывать историю тренда на flash- или SD-карту памяти?.....	49
8.22. Как создать мультязычный проект?	50
8.23. Как записать информацию в файл (архивирование данных) ?	52
9. Вопросы по загрузке, запуску, отладке проекта CODESYS	53
9.1. Как загрузить проект в контроллер?	53
9.2. Почему проект исчезает после загрузки контроллера?.....	53
9.3. Что делать, если при подключении к контроллеру возникает окно с запросом пароля?.....	54
9.4. Как удалить проект из контроллера?	54
9.5. Как выгрузить проект с контроллера?	55
9.6. Почему после загрузки проекта на дисплее СПК отображается надпись «Отсутствует загрузочное приложение»?.....	56
9.7. Почему написанный код не выполняется?	57
9.8. Как отладить проект без контроллера?.....	58
9.9. Как отладить программу?	60
9.10. Как эмулировать входные/выходные сигналы?	62
9.11. Что делать при появлении ошибки «Приложение в исключении»?	63
9.12. Что делать при появлении ошибки «Открыто слишком много визуализаций»?	64
9.13. Что делать при возникновении ошибок компиляции?	64
10. Вопросы по памяти СПК	66
10.1. Для чего используется встроенная память СПК? Какой объем встроенной памяти доступен пользователю?.....	66
10.2. Существует ли ограничение на размер проекта?	66
10.3. Какие USB- и SD- накопители можно подключать СПК?	66
10.4. По какому пути монтируются USB- и SD- накопители?	66
10.5. Сколько retain переменных можно использовать в проекте?	66
10.6. Можно ли использовать в проекте persistent переменные?	67
11. Проблемы настройки обмена с другими устройствами	68
11.1. Какие интерфейсы и протоколы используются для подключения к СПК других устройств? 68	
11.2. Как соотносятся номера COM-портов СПК с номерами портов в CODESYS?.....	68
11.3. Какая распиновка у COM-портов СПК 1xx?.....	69
11.4. Основные особенности настройки обмена по Modbus RTU.....	70
11.4.1. Основные особенности настройки обмена по Modbus RTU в режиме Master	70
11.4.2. Основные особенности настройки обмена по Modbus RTU в режиме Slave	77
11.5. Основные особенности настройки обмена по Modbus ASCII	80

11.6. Основные особенности настройки обмена по Modbus TCP	80
11.6.1. Основные особенности настройки обмена по Modbus TCP в режиме Master	80
11.6.2. Основные особенности настройки обмена по Modbus TCP в режиме Slave	87
11.7. Протокол OVEN	90
11.8. Протокол CAN OPEN	90
12. Вопросы по удаленному доступу: WEB-визуализация, модем ПМ01	91
12.1. Почему не работает WEB-визуализация?	91
12.2. Как посмотреть WEB-визуализацию из интернета?	91
12.3. Как настроить обмен данными через модем ПМ01?	91
12.4. Как с помощью модема ПМ01 отправлять sms?	92
13. Вопросы по системному времени.....	93
13.1. Как отобразить системное время в визуализации?	93
13.2. Как изменить системное время контроллера?	93
13.3. Как считать системное время из кода программы?	94
13.4. Как изменить системное время из кода программы?	96
13.5. Что делать, если системное время постоянно сбивается?	97
13.6. Как настроить синхронизацию контроллера с источником точного времени по протоколу SNTP?	97
14. Остальные вопросы.....	98
14.1. Для чего используется функция SysExecute?.....	98
14.2. Как настроить яркость подсветки?.....	99
14.3. Как использовать аппаратные кнопки?.....	100
14.4. Как использовать индикаторы аппаратных кнопок?	101
14.5. Как использовать звукоизлучатель (зуммер, «пищалку»?	102
14.6. Как управлять аудиовыходом?	103
14.7. Как перезагрузить контроллер из кода программы?.....	104
14.8. Как организовать терминальное подключение к контроллеру?	104
14.9. Как открыть в Linux контроллера диспетчер задач?	107
14.10. Как подключиться к файловой системе контроллера?	108
14.11. Как защитить доступ к Linux контроллера паролем?	109

1. Цель документа

Данный документ представляет собой сборник наиболее часто задаваемых вопросов, возникающих при работе с контроллерами СПК, и ответов на них. В некоторых случаях вопросы охватывают слишком большую предметную область – тогда вместо ответа приводится ссылка на документ, посвященный затронутой теме. Все документы находятся на диске с ПО, входящем в комплект поставки, а также доступны для скачивания на [сайте](#) или [форуме](#) компании ОВЕН.

При написании документа использовалась среда программирования **CODESYS V3.5 SP6**, в более ранних или поздних версиях некоторые компоненты могут отсутствовать или отличаться от приведенных на скриншотах.

2. Восемь самых частых вопросов

- [1. Как перепрошить СПК?](#)
- [2. Как связаны версии прошивки, CODESYS и target-файла?](#)
- [3. Что делать, если CODESYS не видит СПК?](#)
- [4. В CODESYS не хватает определенных компонентов и настроек, что делать?](#)
- [5. Как перезагрузить СПК?](#)
- [6. Почему после загрузки проекта на дисплее СПК отображается надпись «Отсутствует загрузочное приложение»?](#)
- [7. Почему проект исчезает после перезагрузки контроллера?](#)
- [8. Как соотносятся номера COM-портов СПК с номерами портов в CODESYS?](#)

3. Начало работы

3.1. Что такое СПК?

Сенсорный программируемый контроллер (СПК) - это устройство класса **HMI** (человеко-машинный интерфейс), которое совмещает в одном корпусе программируемый логический контроллер с панелью оператора. СПК позволяет не только отображать информацию, но и управлять технологическим процессом в соответствии с заданными алгоритмами, а также архивировать и передавать информацию.

Руководство по эксплуатации СПК можно найти на сайте [ОВЕН](#) в разделе **Поддержка/Документация/Руководства по эксплуатации**.

3.2. Как программируется СПК?

Программирование контроллера (в т.ч. создание пользовательских алгоритмов любого уровня сложности) и разработка экранов визуализации осуществляется в единой среде **CODESYS V3.x**, которая распространяется бесплатно и находится на диске с ПО, входящем в комплект поставки.

Вопросы, возникающие при работе с **CODESYS**, рассмотрены в [п. 6](#) (вопросы по установке) и [п. 8](#) (вопросы по использованию) данного документа.

3.3. Как перезагрузить СПК?

Перезагрузить СПК можно одним из следующих способов:

1. выдернув кабель питания (из СПК или из источника питания);
2. нажатием на кнопку **Сброс** (с помощью тонкого заостренного предмета диаметром не более 3 мм) на задней панели СПК (**только для СПК2xx**);
3. с помощью командной строки **Linux**. Можно использовать обращение к командной строке непосредственно из кода программы в **CODESYS** (см. [п. 14.6](#)).

3.4. Как выполнить монтаж СПК?

Процесс монтажа описан в Руководстве по эксплуатации СПК, которое можно найти на сайте [ОВЕН](#) в разделе **Поддержка/Документация/Руководства по эксплуатации**.

4. Вопросы по прошивке

4.1. Что такое прошивка?

Прошивка - это системное программное обеспечение, которое управляет работой контроллера на аппаратном уровне. В связи с добавлением новых функций и исправлением ошибок, регулярно осуществляется выпуск новых версий прошивок. *При необходимости* пользователь может **самостоятельно сменить** версию прошивки

4.2. Как узнать версию прошивки?

Версия прошивки отображается в [сервисном меню](#) и [конфигураторе](#) СПК.

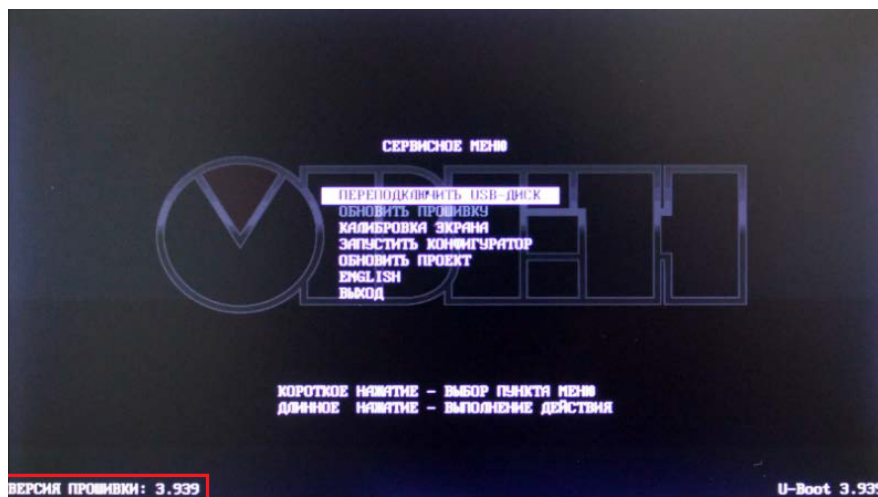


Рис. 4.1. Внешний вид сервисного меню СПК

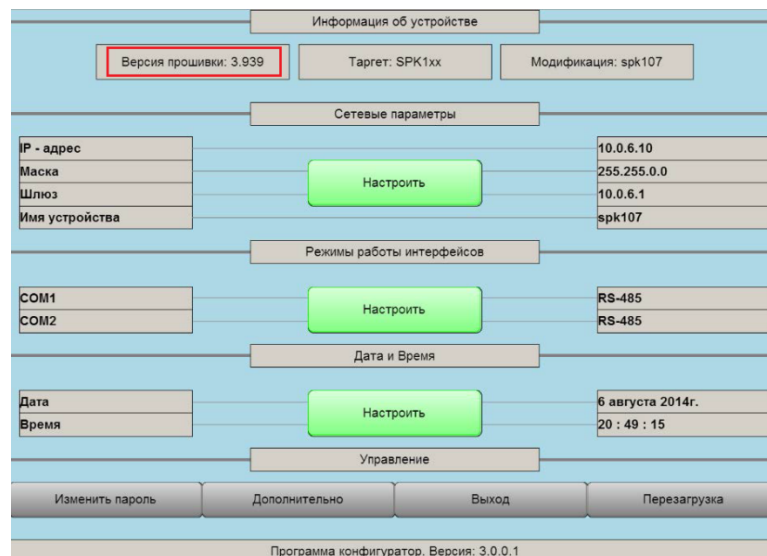


Рис. 4.2. Внешний вид конфигуратора СПК

4.3. В каких случаях следует обновлять прошивку СПК?

Прошивку СПК необходимо обновлять в следующих случаях:

1. по указанию [техподдержки](#) компании ОБЕН;
2. в случае возникновения ошибок, связанных с прошивкой, и рекомендаций на [форуме ОБЕН](#) обновить ее;
3. в случае добавления в новой прошивке поддержки необходимого пользователю функционала и рекомендаций на [форуме ОБЕН](#) обновиться до нее.

4.4. Какие могут быть последствия при перепрошивке? Можно ли будет «откатиться» на старую версию прошивки?

В случае возникновения программных ошибок в процессе перепрошивки, всегда есть возможность перепрошить СПК на прежнюю версию прошивки. При возникновении проблем следует связать с [техподдержкой](#) компании ОБЕН.

4.5. Где взять новые прошивки?

Новые прошивки, необходимое для перепрошивки ПО и инструкции доступны на [сайте ОБЕН](#) в разделе **CODESYS V3/Сервисное ПО**.

4.6. Как перепрошить СПК?

Процесс перепрошивки описан в руководстве, доступном на [сайте ОБЕН](#) в разделе **CODESYS V3/Сервисное ПО**.

5. Конфигуратор и сервисное меню

5.1. Для чего нужен конфигуратор?

С помощью конфигуратора можно:

1. настроить сетевые параметры контроллера;
2. настроить режимы работы последовательных портов (RS-232 или RS-485 соответственно);
3. установить/изменить системное время контроллера;
4. настроить ориентацию дисплея контроллера, режим работы курсора, посмотреть информацию о состоянии памяти контроллера и подключенных запоминающих устройств (вкладка **Дополнительно**).

The screenshot displays the SPK configuration utility interface. At the top, there is a section for 'Информация об устройстве' (Device Information) with fields for 'Версия прошивки: 3.944', 'Таргет: spk207.03.web', and 'Модификация: spk207web'. Below this is the 'Сетевые параметры' (Network Parameters) section, which includes fields for IP address (10.0.6.10), Mask (255.255.0.0), Gateway (10.0.6.1), Device Name (spk207web), DNS1 (10.0.6.1), and DNS2 (8.8.8.8). A green 'Настроить' (Configure) button is present. The 'Режимы работы интерфейсов' (Interface Modes) section shows COM2 and COM3 ports set to RS-485 mode, with another 'Настроить' button. The 'Дата и Время' (Date and Time) section shows the date set to 1 января 1970г. and the time to 0 : 17 : 35, also with a 'Настроить' button. The 'Управление' (Management) section at the bottom contains buttons for 'Изменить пароль', 'Дополнительно', 'Выход', and 'Перезагрузка'. The footer indicates 'Программа конфигуратор. Версия: 3.0.0.3'.

Рис. 5.1. Внешний вид конфигуратора СПК

5.2. Как попасть в конфигуратор?

В зависимости от модели и прошивки СПК, процесс открытия конфигуратора может различаться:

Модель контроллера	Версия прошивки	Способ открытия конфигуратора (наличие/отсутствие сервисного меню)
СПК 1xx	младше 3.935	конфигуратор открывается при кратковременном нажатии на экран контроллера в процессе загрузки (во время отображения надписи TAP to configure, LONG to calibrate);
	3.935 и старше	конфигуратор запускается из сервисного меню , которое открывается после троекратного нажатия на экран в процессе загрузки контроллера
СПК 2xx M01	младше 3.302	конфигуратор запускается при включении контроллера, если тумблер на задней панели СПК находится в положении «1» (верхнее);
	3.302 и старше	конфигуратор открывается при кратковременном нажатии на экран контроллера в процессе загрузки (во время отображения надписи TAP to configure, LONG to calibrate);
СПК 2xx M02, M04	3.943 и старше	конфигуратор запускается из сервисного меню , которое открывается после троекратного нажатия на экран в процессе загрузки контроллера

5.3. Какой пароль у конфигуратора?

Пароль — **owen**, для переключения экранной клавиатуры на нижний регистр необходимо нажать клавишу **shift**.

5.4. Как попасть в сервисное меню?

Сервисное меню присутствует только в определенных прошивках определенных моделей СПК (см. [п. 5.2.](#)). Для того, чтобы попасть в него, необходимо после включения питания **до начала загрузки проекта** коснуться дисплея контроллера три раза:

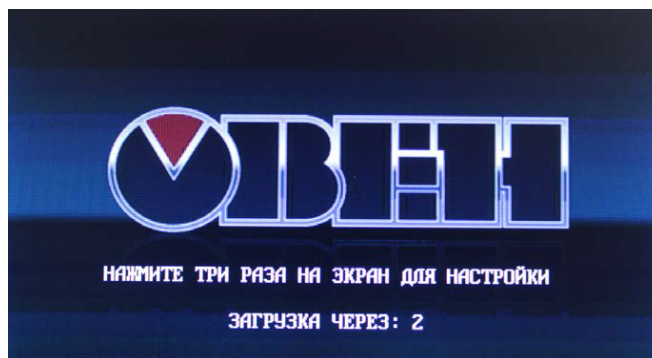


Рис. 5.2. Процесс загрузки контроллера

5.5. Что можно сделать из сервисного меню?

С помощью команд сервисного меню можно:

1. [Обновить прошивку с USB Flash](#);
2. [Произвести калибровку дисплея](#);
3. [Запустить конфигурактор](#);
4. [Загрузить/обновить проект с USB flash](#).



Рис. 5.3. Внешний вид сервисного меню

5.6. Как обновить прошивку с USB flash?

Перепрошивка с flash-накопителя доступна **только** для СПК 1xx и СПК2xx (600 МГц), текущие прошивки которых имеют **сервисное меню** (см. [п. 5.2.](#)).

Процесс перепрошивки с flash-накопителя описан в руководстве, доступном на [сайте ОВЕН](#) в разделе **CODESYS V3/Сервисное ПО**.

5.7. Зачем нужна калибровка дисплея?

По умолчанию дисплей контроллера откалиброван, однако со временем у некоторых приборов сенсорный слой экрана может начать неверно обрабатывать координату точки нажатия. Для корректировки координаты точки нажатия предусмотрена процедура калибровки, представляющая собой последовательное нажатие подсвечивающихся на дисплее ключевых точек.

5.8. Как загрузить проект с USB flash?

Загрузка проекта с flash-накопителя доступно **только** для СПК 1xx и СПК2xx (600 МГц), прошивки которых имеют **сервисное меню** (см. [п. 5.2.](#)).

Для обновления проекта с USB flash необходимо выполнить следующие действия:

[0.] (**только для СПК105**) В [сервисном меню](#) выполнить команду **USB - Порт в режиме сеть**. Если пункт не активирован, СПК автоматически определяет тип подключения HOST или DEVICE (HOST-подключение USB накопителей, DEVICE – подключение к ПК). Если при подключении к ПК, СПК105 автоматически не определился в диспетчере устройств, следует установить режим **Сеть**.

1. отформатировать USB накопитель объемом не более 16 Гб в файловую систему **FAT32**;
2. создать каталог с именем **APP** в корне USB накопителя;
3. произвести компиляцию проекта CODESYS (вкладка **Компиляция**, команда **Компиляция**);
4. создать загрузочное приложение в CODESYS (вкладка **Онлайн**, команда **Создать загрузочное приложение**), в окне **Сохранить как** указать путь к папке. Полный путь к папке не должен содержать символы кириллицы, папка должна находиться на жестком диске ПК;
5. содержимое папки перенести в папку **APP** на USB накопителе;
6. зайти в сервисное меню СПК и подключить USB накопитель с пользовательской программой к USB A порту СПК;
7. в сервисном меню выполнить команду **Переподключить USB**;
8. в сервисном меню выполнить команду **Обновить проект**;
9. проект загрузится в контроллер и автоматически запустится.

6. Вопросы по CODESYS

6.1. Что такое CODESYS?

CODESYS (Controller Development System) — программный комплекс промышленной автоматизации, основанный на стандарте [IEC \(МЭК\) 61131-3](#). Производится и распространяется компанией [3S-Smart Software Solutions GmbH](#) (Германия).

CODESYS используется для создания и отладки прикладного программного обеспечения и разработки интерфейса оператора, которые в сочетании образуют пользовательский проект; этот проект загружается на исполнение в контроллер.

Среда **CODESYS** находится в процессе постоянного развития и улучшения, что приводит к периодическому выпуску **новых версий**. Начиная с **CODESYS 3.0**, версии устанавливаются **независимо** друг от друга (свежая версия не обновляет предыдущую, а устанавливается параллельно), но при этом **необходимо** устанавливать их исключительно в **порядке возрастания**.

6.2. Где взять CODESYS?

CODESYS V3.x присутствует на диске с ПО, который входит в комплект поставки, а так же может быть скачан с [сайта компании OBEH](#) (раздел **CODESYS V3**) или компании 3S ([раздел Download](#)).

6.3. Какую версию CODESYS следует устанавливать?

Рекомендуется устанавливать **CODESYS** с диска, который входит в комплект поставки. При необходимости использовать более старую или более новую версию **CODESYS**, следует помнить о необходимости **соответствия версий** прошивки контроллера, среды **CODESYS** и ее [target-файла](#) (см. руководство «**СПК. Система версий ПО**», расположенное на [сайте OBEH](#) в разделе **CODESYS V3/Документация**).

6.4. Как установить CODESYS?

Процесс установки **CODESYS** описан в документе **Установка CODESYS 3.5 для программирования контроллеров СПК фирмы OBEH**, доступном на [форуме OBEH в разделе СПК](#).

6.5. Что такое target-файл?

Target-файл (файл целевой платформы) является неотъемлемой частью каждого проекта **CODESYS**. Он содержит информацию о ресурсах контроллера и обеспечивает его связь со средой программирования. Каждая модель контроллера **ОВЕН** имеет соответствующий target-файл, который необходимо установить перед началом создания проекта в среду **CODESYS**. Target-файлы находятся на диске с ПО из комплекта поставки, а так же доступны на [сайте ОВЕН](#) в разделе **CODESYS V3/Сервисное ПО**.

Версия target-файла должна соответствовать версии прошивки контроллера (см. руководство «СПК. Система версий ПО», расположенное на сайте **ОВЕН** в разделе **CODESYS V3/Документация**).

6.6. Где взять target-файл?

Target-файл присутствует на диске с ПО, который входит в комплект поставки, а также доступен на [сайте ОВЕН](#) в разделе **CODESYS V3/Сервисное ПО**.

6.7. Как установить target-файл и подключить его к проекту?

На [сайте ОВЕН](#) target-файлы выкладываются в виде **пакетов**. Для установки их среду **CODESYS** необходимо выбрать во вкладке **Инструменты** компонент **Менеджер пакетов**, в нем нажать кнопку **Установить**, после чего указать путь к соответствующему **target-файлу** в формате **.package**:

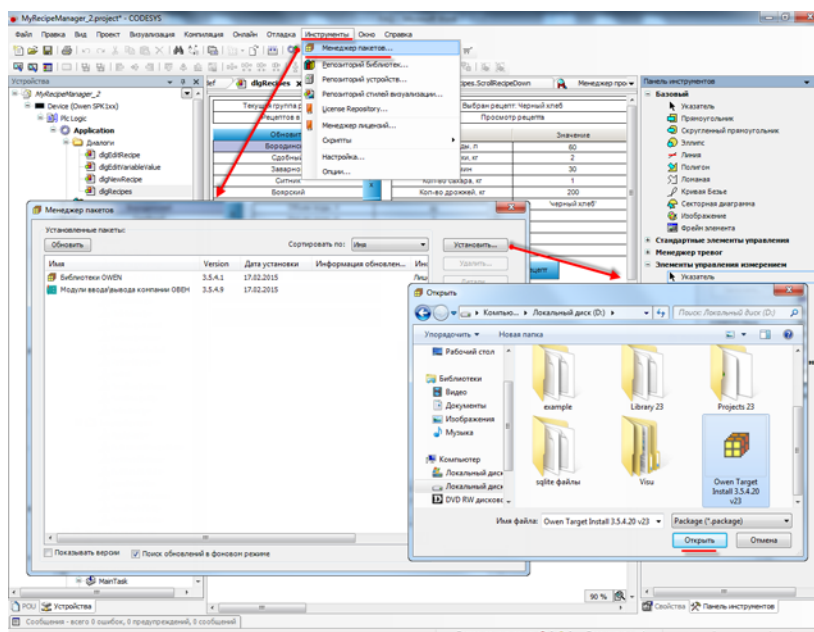


Рис. 6.1. Установка target-файла в среду CODESYS

Для использования target-файла в проекте CODESYS необходимо на **Панели устройств** выбрать компонент **Device** и, нажав на него **ПКМ**, открыть окно **Обновить устройство**, в котором следует выбрать нужный target-файл (*при необходимости* поставить галочки **Отображать все версии** и **Показать устаревшие версии**):

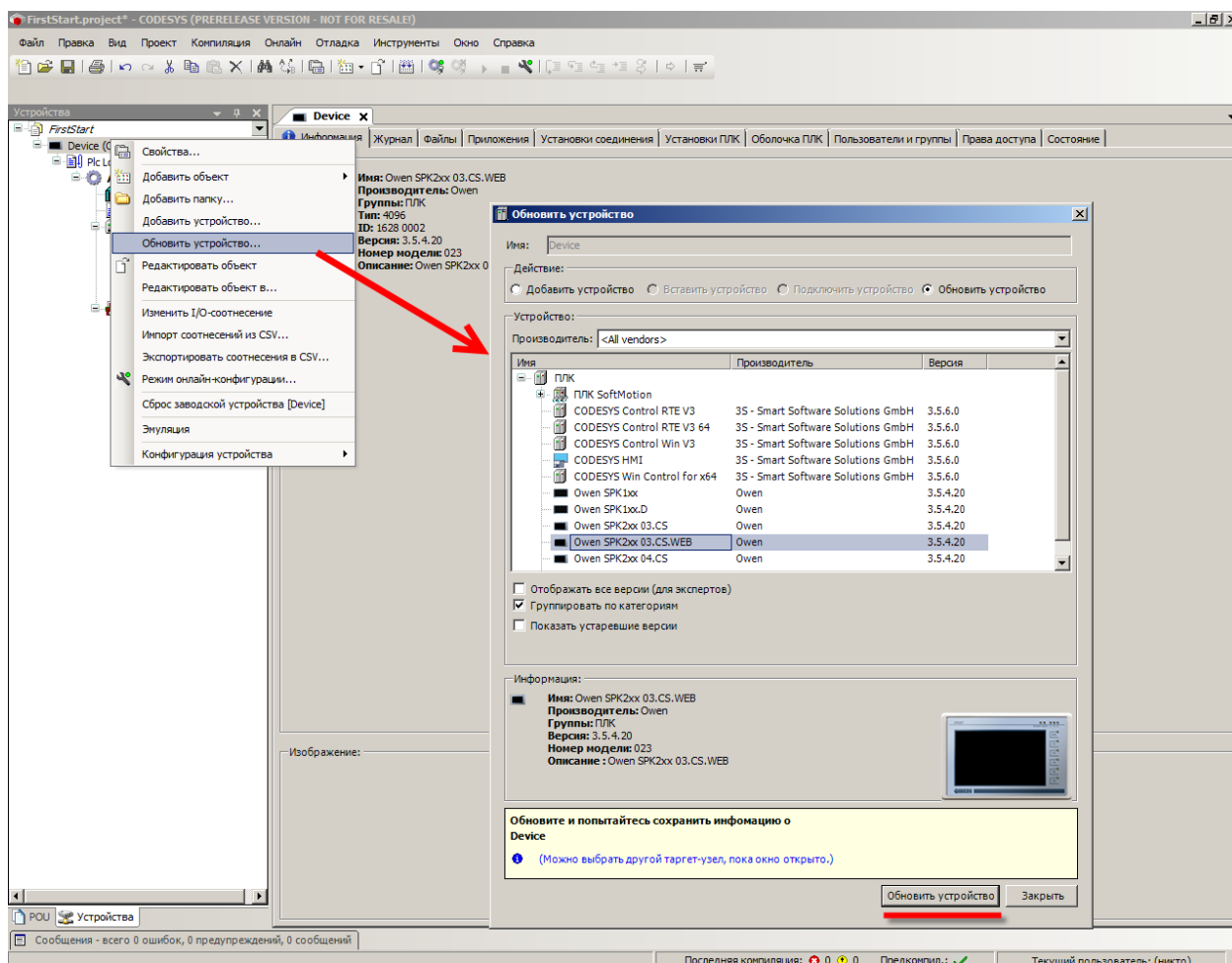


Рис. 6.2. Выбор target-файла в проекте CODESYS

6.8. Как связаны версии прошивки, CODESYS и target-файла?

Нормальная работа контроллера гарантируется при использовании заводской прошивки контроллера в сочетании с версией **CODESYS** и **target-файла** с диска из комплекта поставки; более подробную информацию можно найти в документе «СПК. Система версий ПО», расположенном на [сайте OBEH](#) в разделе **CODESYS V3/Документация**.

6.9. Как перенести проект из одной версии CODESYS в другую?

Этот вопрос рассмотрен в документе «СПК. Система версий ПО», расположенном на [сайте OBEH](#) в разделе **CODESYS V3/Документация**.

6.10. Как перенести проект с одного компьютера на другой?

Для переноса проекта **CODESYS** с одного компьютера на другой необходимо сохранить проект в виде файла формата **.project** (Файл – Сохранить проект как) или архива формата **.archiveproject** (Файл – Архив проекта – Сохранить/Отправить проект). Рекомендуется использовать архивы проектов, т.к. они включают в себя используемые в проекте библиотеки и пользовательские компоненты (например, шаблоны модулей).

6.11. Что такое библиотека?

Библиотека – это редактируемый файл формата **.library** (или нередитруемый формата **.compiled-library**), представляющий собой набор готовых **функциональных блоков**. Библиотеки являются либо **системными** (входят в состав **CODESYS**), либо **пользовательскими** (например, библиотеки, разработанные компанией **OBEH**). С помощью библиотек можно без дополнительных усилий организовать обмен по протоколам с другими устройствами, архивацию файлов и т.д.

6.12. Где взять дополнительные библиотеки?

Библиотеки компании **OBEH** можно найти на [сайте OBEH](#) в разделе **CODESYS V3/Библиотеки CODESYS**, а также на [форуме OBEH в разделе СПК](#). Другие библиотеки можно найти в Интернете, среди них стоит отметить свободно распространяемую библиотеку **OSCAT**, разрабатываемую одноименным сетевым сообществом – на данный момент она включает уже несколько сотен функциональных блоков.

6.13. Как установить библиотеку и подключить ее к проекту?

Компания **ОВЕН** в основном распространяет библиотеки в виде **пакетов**. Их установка описана в [п. 6.7](#). Если библиотека распространяется в формате **.library** (или **.compiled-library**), то для установки библиотеки в **CODESYS** необходимо во вкладке **Инструменты** выбрать компонент **Репозиторий библиотек**, в нем нажать **кнопку Установить**, после чего указать путь к соответствующей библиотеке.

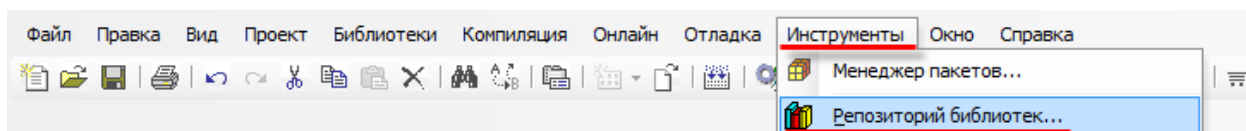


Рис. 6.3. Открытие репозитория библиотек

Для подключения библиотеки к проекту необходимо открыть компонент **Менеджер библиотек** и нажать **кнопку Добавить библиотеку**, после чего выбрать нужную библиотеку и нажать **ОК**.

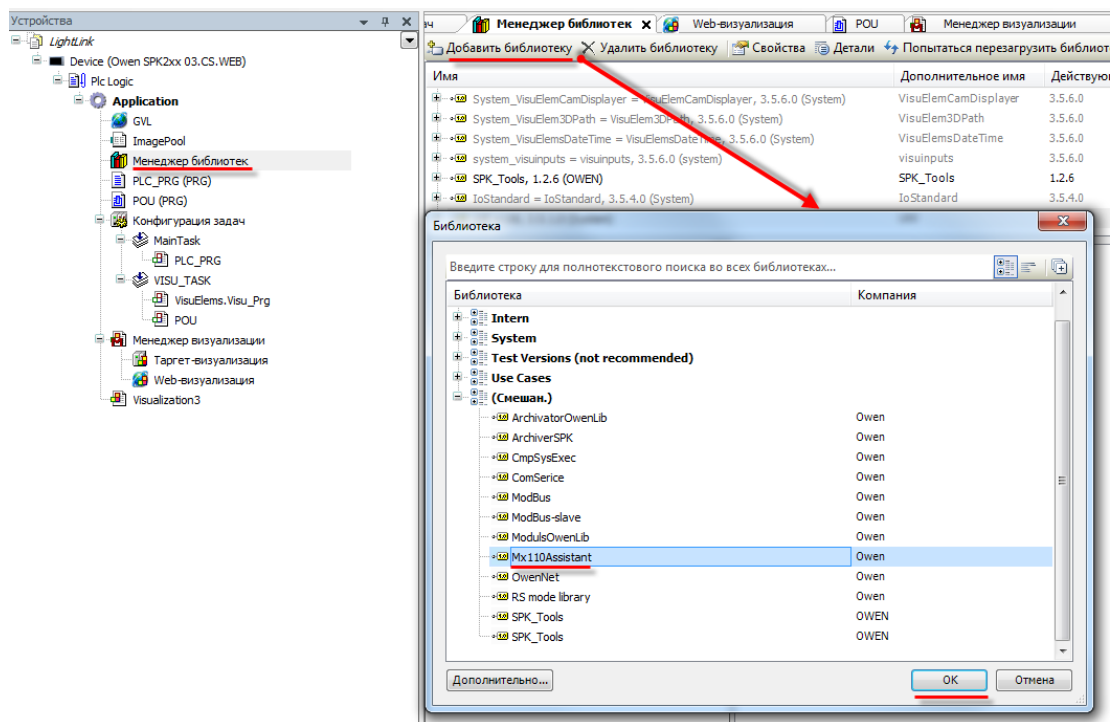


Рис. 6.4. Подключение библиотеки к проекту

6.14. При открытии проекта возникает ошибка «Библиотека <...> отсутствует в проекте», что делать?

Такая ошибка может возникать в двух случаях:

1. На компьютере отсутствует пользовательская библиотека, используемая в проекте. В этом случае необходимо найти библиотеку, которая использовалась при разработке проекта. По этой причине рекомендуется сохранять проекты как [архивы](#).

2. На компьютере отсутствует нужная версия стандартной библиотеки, используемая в проекте. В этом случае рекомендуется установить **архив репозитория**, доступный на [сайте OBEH](#) в разделе **CODESYS V3/CODESYS V3** или [CODESYS Store](#). **Крайне рекомендуется** устанавливать архив репозитория до начала создания проекта; проект, созданный до установки архива репозитория, после его установки может работать некорректно.

6.15. Что такое CODESYS HMI, CODESYS Application Composer, CODESYS SoftMotion?

Это дополнительные **платные** компоненты **CODESYS**, распространяемые **отдельно** от самой среды. Их можно приобрести в онлайн-магазине [CODESYS Store](#).

CODESYS HMI – это приложение, предназначенное для операторского управления с отдельного компьютера локальной сети. Оно обеспечивает управление в реальном времени и не требовательно к ресурсам. Весь интеллект системы сосредоточен в ПЛК/СПК, а **HMI** выполняет роль тонкого клиента отображения. Его типичные применения - это встроенные пульта управления станками, погрузчиками, кранами, трамваями и подобными системами, где нужна быстрая гарантированная реакция и стоимость оборудования критична.

CODESYS Application Composer – представляет собой высокоуровневый инструмент для быстрой разработки, которая выполняется на основе заранее подготовленных наборов прикладных программных модулей. Такой модуль может обслуживать определенную часть машины или системы - например, это может быть пневматический цилиндр, автооператор, терморегулятор и т.п. Каждый модуль включает программный код, конфигурацию входов/выходов, параметры и графическое представление для визуализации. Пользователь строит структуру своей системы управления, используя необходимые модули, определяет их настройки и связи в специальных редакторах. Затем интегрированные генераторы кода автоматически создают законченное, хорошо структурированное программное приложение на языках стандарта **МЭК 61131-3**. Одновременно генерируется соответствующая визуализация. Программы компилируются и загружаются в контроллер. При необходимости пользователь может просматривать и корректировать полученный код. Такой подход позволяет перейти от рутинного программирования к модульному проектированию прикладных проектов. Он предназначен для пользователей, хорошо знающих устройство машин, технологию соответствующего производства, но не владеющих программированием. Кроме того, он в разы ускоряет тиражирование проектов.

Примеры использования **Application Composer** входят в состав **CODESYS** и расположены по адресу ...**3S CODESYS 3.5 SP6\CODESYS\Projects\ApplicationComposer**.

CODESYS SoftMotion - это встроенный в среду программирования и систему исполнения **CODESYS** функциональный набор средств управления движением, от простейших перемещений по одной оси до сложной многомерной интерполяции современных ЧПУ. Обычно задачи управления движением решают специализированные аппаратно-зависимые системы. **CODESYS SoftMotion** предлагает совершенно новое решение - программный инструментальный набор, позволяющий решать задачи любого уровня сложности. **CODESYS SoftMotion** содержит библиотеку числового управления (NC library) PLCopen, интерполятор и блоки подготовки пути, готовые драйверы для наиболее распространенных сетей (CAN, Sercos), элементы визуализации, позволяющие определять произвольные поверхности для каждого приложения, и графические редакторы задания движения. **CODESYS SoftMotion** предлагает решения для таких применений, как обрабатывающие центры с ЧПУ, специализированные машины с ПЛК-функциональностью, ЧПУ с ограниченными функциями перепрограммирования.

Примеры использования **SoftMotion** входят в состав **CODESYS** и расположены по адресу **...\3S CODESYS 3.5 SP6\CODESYS\Projects\SoftMotion**.

7. Вопросы по подключению СПК к ПК

7.1. Как подключиться к СПК из CODESYS?

Подробно этот вопрос рассмотрен в главе 5 руководства «СПК. Первый старт», доступном на [форуме OWEN в разделе СПК](#).

Для подключения к СПК из **CODESYS** следует помнить о:

1. необходимости соответствия версии **target-файла** прошивке контроллера;
2. необходимости соответствия **сетевых настроек** контроллера и компьютера.

При необходимости изменить **target-файл** проекта можно, выбрав на **Панели устройств** компонент **Device** и, нажав на него **ПКМ**, открыть окно **Обновить устройство**:

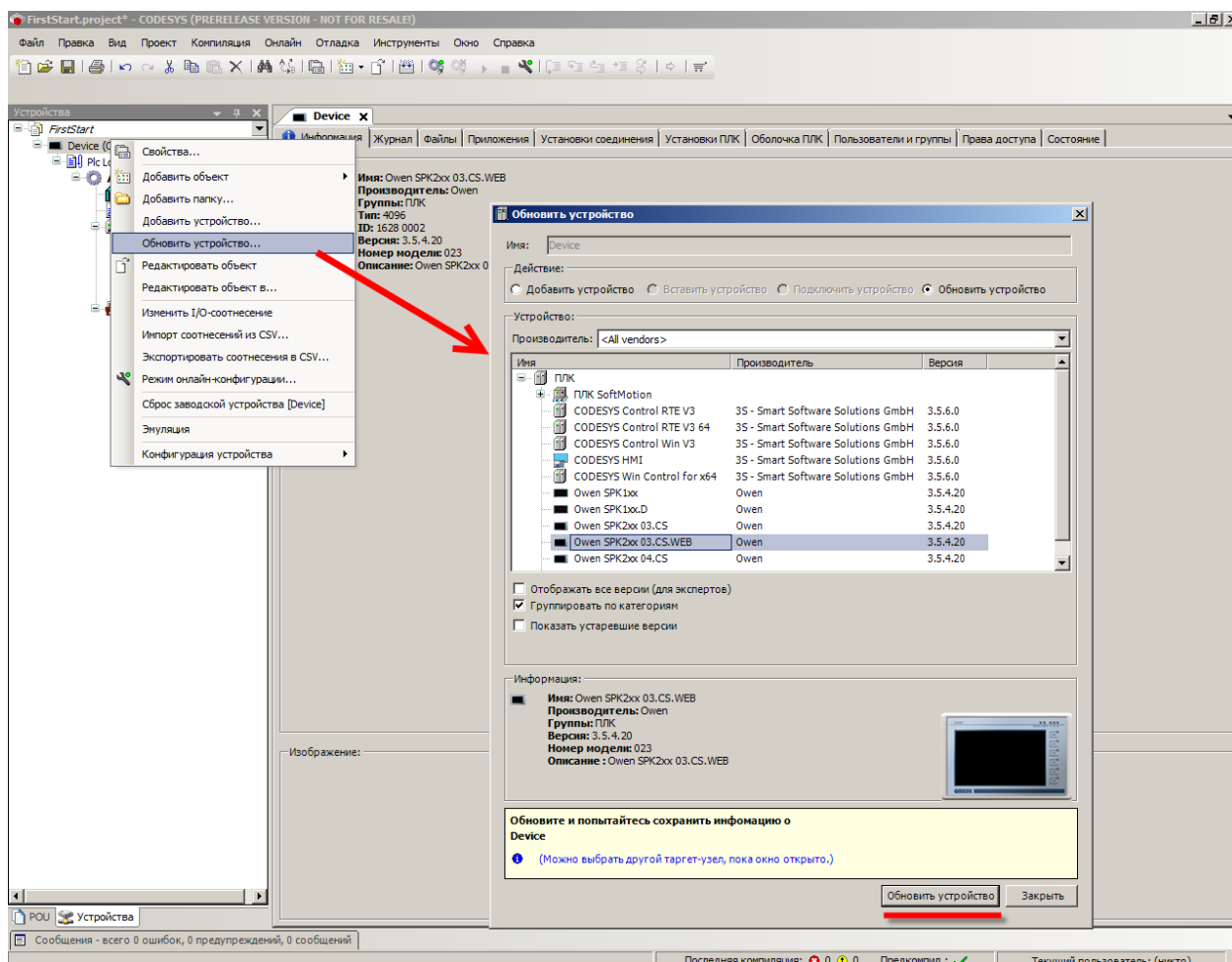


Рис. 7.1. Окно **Обновить устройство**

Теперь следует выбрать устройство, соответствующее имеющейся модели контроллера. **Название модели** указано на **задней панели СПК** и в **конфигураторе**.

После выбора устройства нужно нажать кнопку **Обновить устройство** и закрыть окно. На **Панели устройств** у компонента **Device** отобразится название выбранного **устройства**.

После этого необходимо произвести настройку **gateway** (шлюза).

Двойным нажатием **ЛКМ** по компоненту **Device** (или одиночным нажатием по вкладке вверху рабочей области) перейдем к его настройкам. Откроем вкладку **Установки соединения** и нажав на кнопку **gateway** выберем пункт **Add new gateway**:

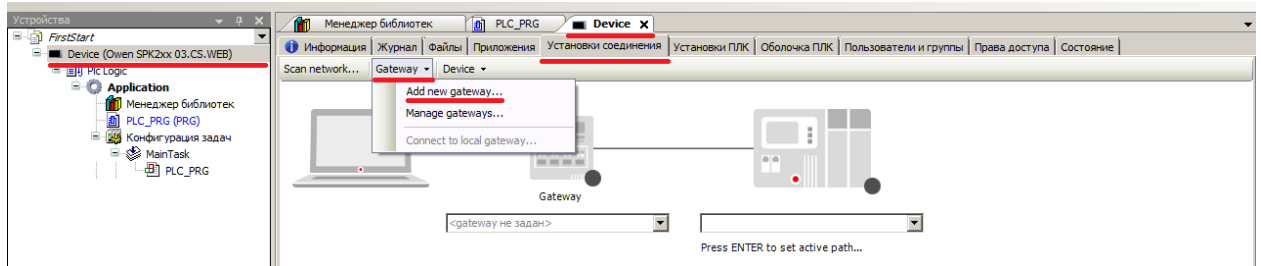


Рис. 7.2. Создание нового gateway (шлюза)

Настройки оставим по умолчанию (имя – **Gateway-1**, IP-адрес – **localhost**). Закроем окно настройки шлюза и нажмем кнопку **Scan network**. В появившемся списке выберем наш контроллер и установим связь, нажав **OK**.

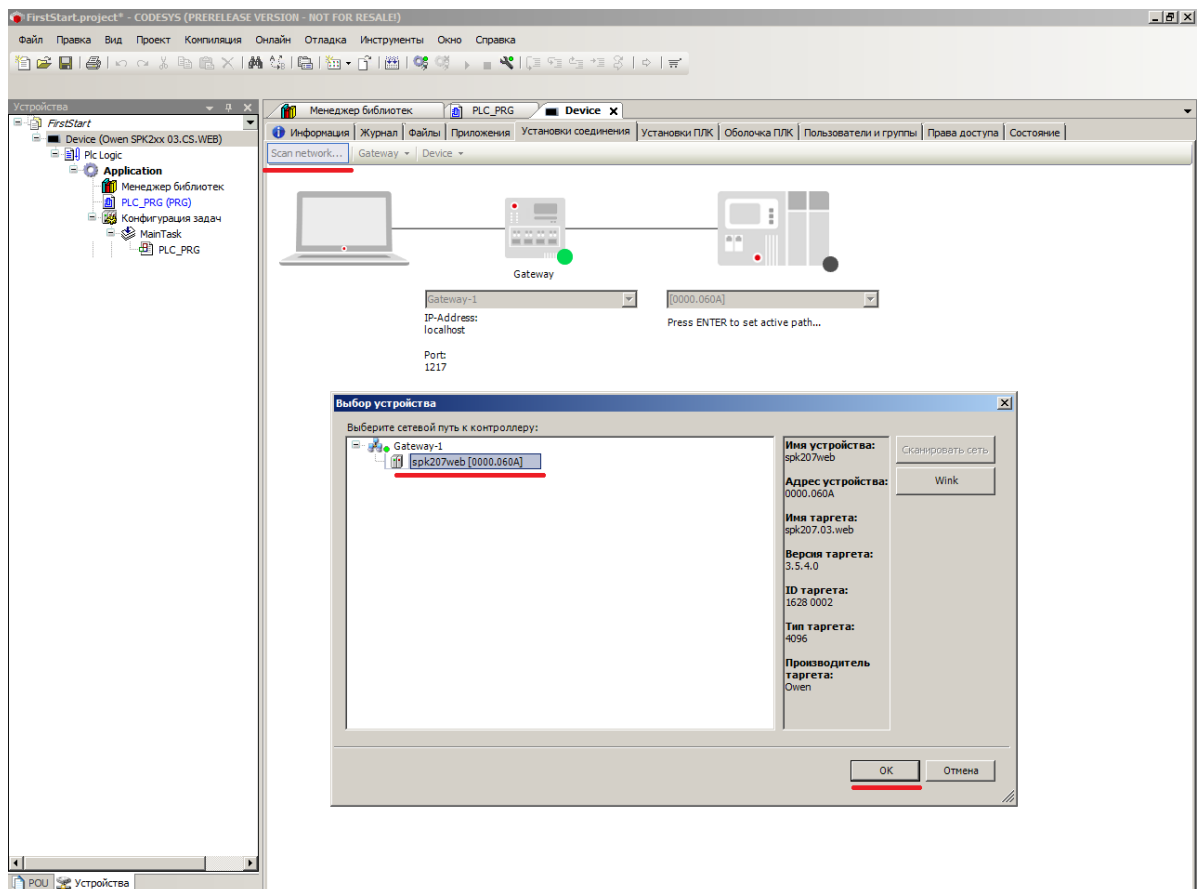


Рис. 7.3. Окно сканирования сети

В случае успешной установки связи индикаторы шлюза и контроллера загорятся зеленым:

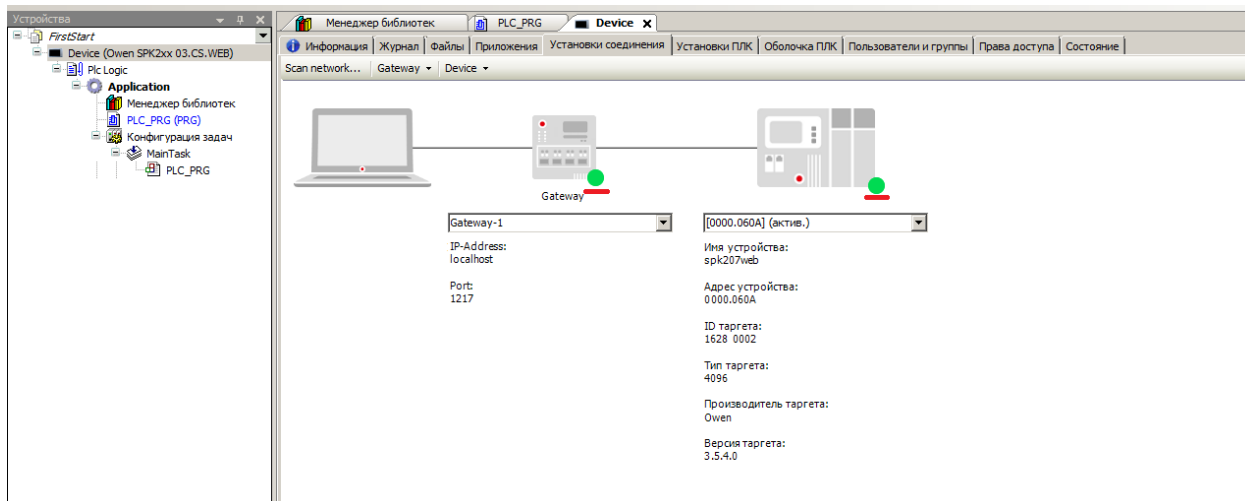


Рис. 7.4. Результат успешной установки связи

Обратите внимание, что подключение к СПК из **CODESYS** при запущенном [конфигураторе](#) невозможно.

7.2. Как узнать IP-адрес СПК?

По умолчанию IP-адрес СПК – **10.0.6.10**.

Узнать IP-адрес контроллера можно следующими способами:

1. в [конфигураторе](#), меню **Сетевые параметры**;
2. с помощью сканера IP-адресов, например, [Advanced IP Scanner](#) – найдя в сформированном им списке устройство с **MAC-адресом**, совпадающим с адресом, указанным на задней панели СПК.

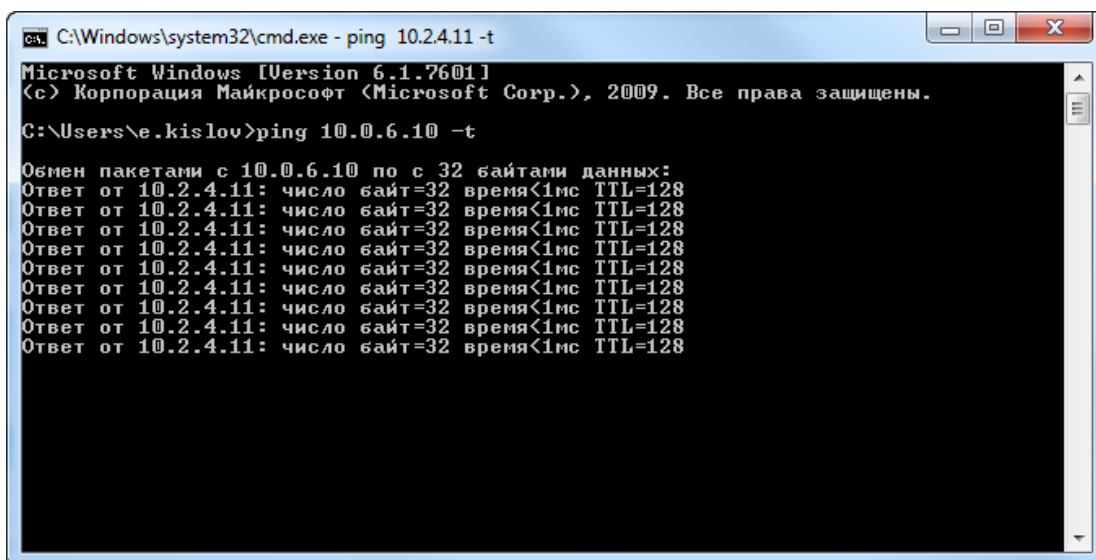
7.3. Как поменять IP СПК?

С помощью [конфигуратора](#), меню **Сетевые параметры**.

7.4. CODESYS не видит СПК, что делать?

1. Проверить наличие **пинга** между компьютером и контроллером:

Открыть **командную строку** (Пуск — Все программы — Стандартные — **Командная строка**) и ввести команду **ping <IP-адрес контроллера> -t**. При наличии связи можно будет увидеть следующий ответ:



```
C:\Windows\system32\cmd.exe - ping 10.24.11 -t
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Users\е.kislov>ping 10.0.6.10 -t

Обмен пакетами с 10.0.6.10 по 32 байтами данных:
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
Ответ от 10.2.4.11: число байт=32 время<1мс TTL=128
```

Рис. 7.5. Результат выполнения команды **ping**

2. Если пинг есть, тогда необходимо проверить соответствие версий CODESYS, [target-файла](#) и [прошивки](#) контроллера (см. документ «СПК. Система версий ПО», расположенный на [сайте OBEH](#) в разделе **CODESYS V3/Документация**).

3. Если пинга нет, тогда необходимо проверить соответствие сетевых настроек контроллера (в [конфигураторе](#)) и компьютера.

7.5. Как подключить СПК к локальной сети?

При наличии в локальной сети **DHCP-сервера**, достаточно в [конфигураторе](#) СПК включить **режим DHCP** (вкладка **Сетевые параметры**). В этом случае при подключении СПК к любому настроенному сетевому устройству (коммутатору или роутеру), он получит сетевые настройки автоматически.

Сетевые параметры			
<input checked="" type="checkbox"/> DHCP			
IP - адрес			
10	0	6	10
Маска			
255	255	0	0
Шлюз			
10	0	6	1
Имя устройства			
spk207web			
DNS1			
10	0	6	1
DNS2			
8	8	8	8
Закрыть			

Рис. 7.6. Включение режима DHCP

При отсутствии DHCP-сервера необходимо выставить сетевые параметры СПК в соответствии с требованиями локальной сети.

8. Вопросы по созданию проекта CODESYS

8.1. Как сменить язык CODESYS?

По умолчанию среда программирования запускается с **русскоязычным** интерфейсом. При необходимости можно поменять язык в меню **Инструменты**, вкладка **Опции**, пункт **Международные установки**:

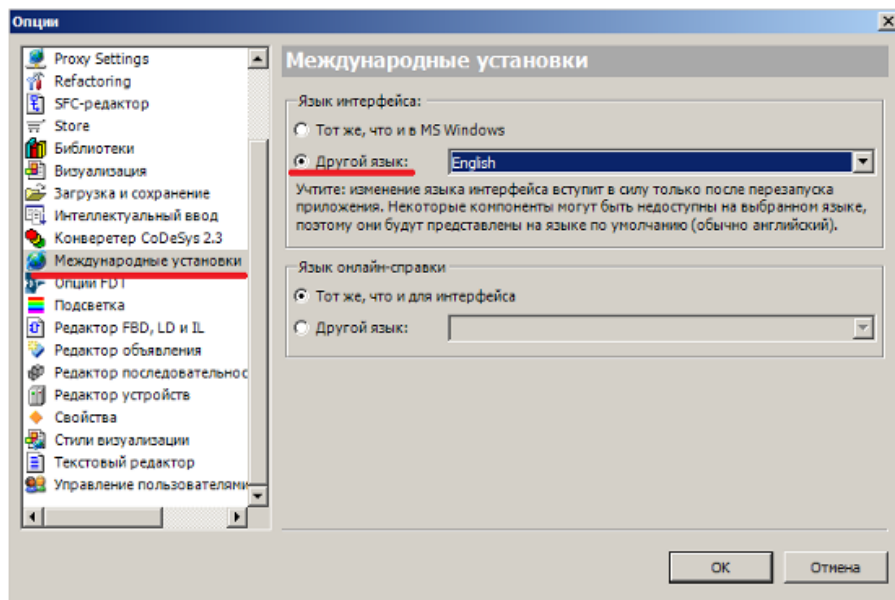


Рис. 8.1. Меню языковых настроек CODESYS

Изменения вступят в силу **после перезапуска** CODESYS.

8.2. У меня не хватает определенных компонентов и настроек, что делать?

1. Включить **профессиональный** режим настроек CODESYS, поскольку он предоставляет больше настроек и возможностей по сравнению со **стандартным**. При необходимости режим можно сменить в меню **Инструменты**, вкладка **Опции**, пункт **Свойства**, кнопка **Заданные наборы свойств**:

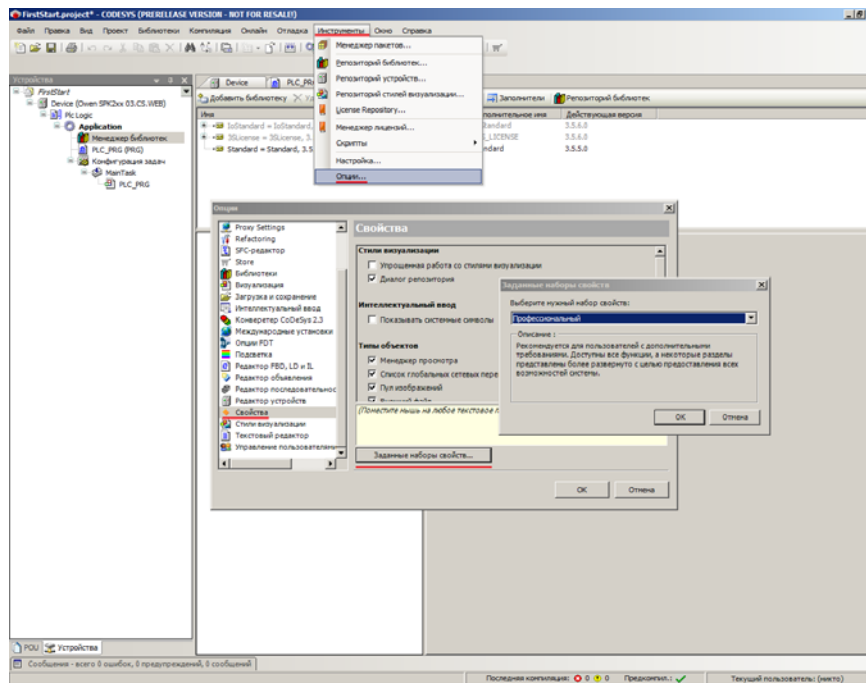


Рис. 8.2. Смена режима установок среды CODESYS

2. При просмотре свойств элементов в **Редакторе визуализации** использовать режим **Эксперт**:

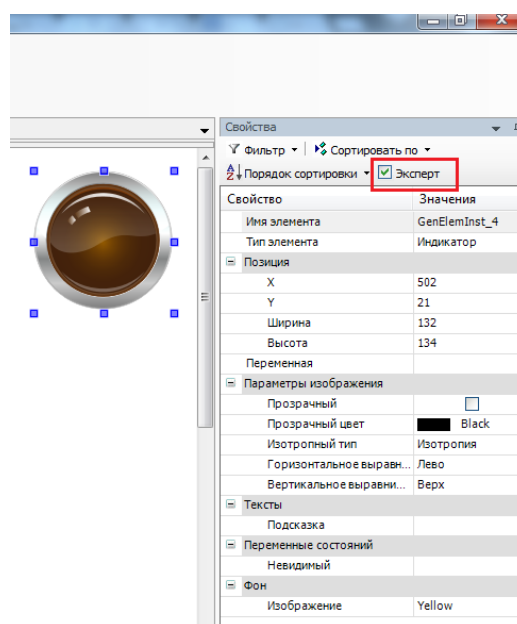


Рис. 8.3. Установка режима «Эксперт»

8.3. Что делать, если интерфейс «рассыпался»?

При возникновении проблем с интерфейсом **CODESYS** (пропадания окон, изменения их местоположения и т.д.) можно сбросить внешний вид интерфейса к настройкам по умолчанию с помощью команды **Сбросить параметры окна** из меню **Окно**:

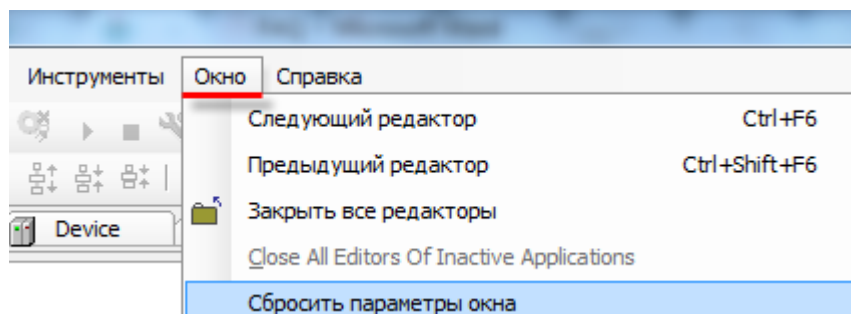


Рис. 8.4. Команда сброса параметров окна

8.4. Как включить сетку в Редакторе визуализации?

В меню **Опции** (**Инструменты – Опции – Визуализация – Сетка**):

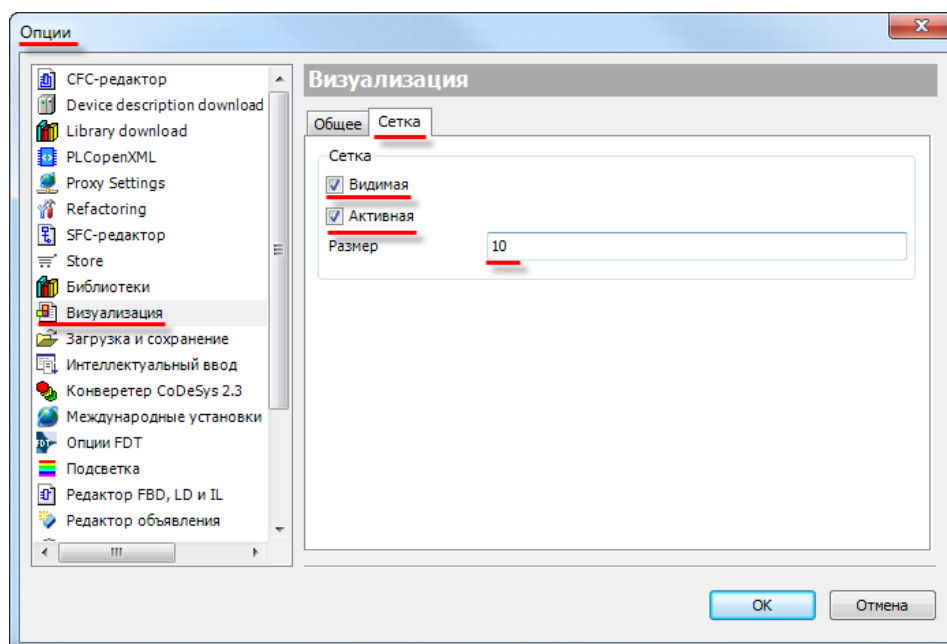


Рис. 8.5. Включение сетки в Редакторе визуализации

8.5. Как настроить размер экрана визуализации?

Если разрешение визуализации в CODESYS не соответствует разрешению дисплея СПК, то она может отображаться некорректно. Разрешение дисплеев всех моделей СПК – **800x480**, за исключением СПК105 – его разрешение **480x272**.

Разрешение визуализации в **CODESYS** настраивается для:

1. каждого типа визуализации отдельно (target-визуализация, web-визуализация);

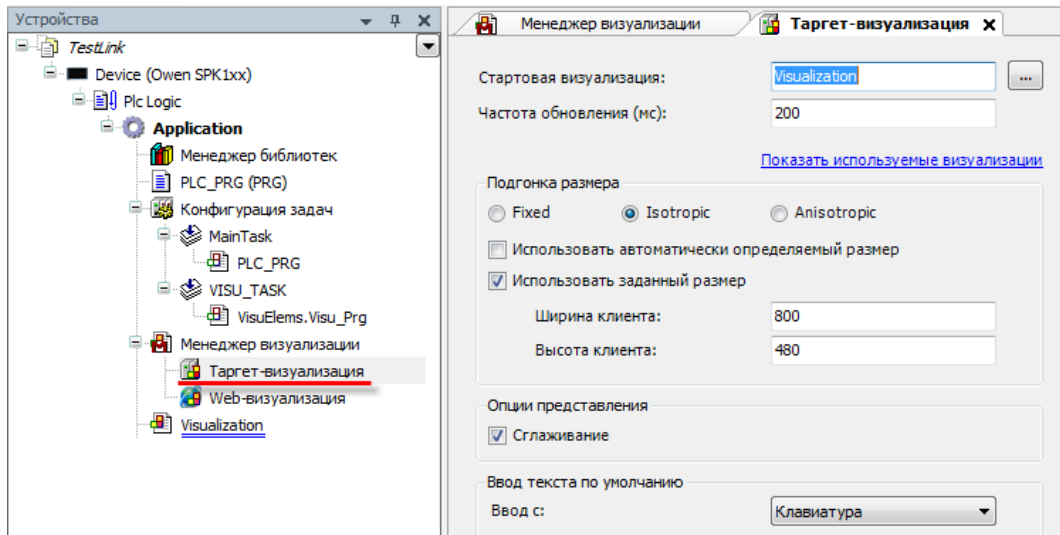


Рис. 8.6. Настройка разрешения target-визуализации

2. каждого экрана визуализации отдельно.

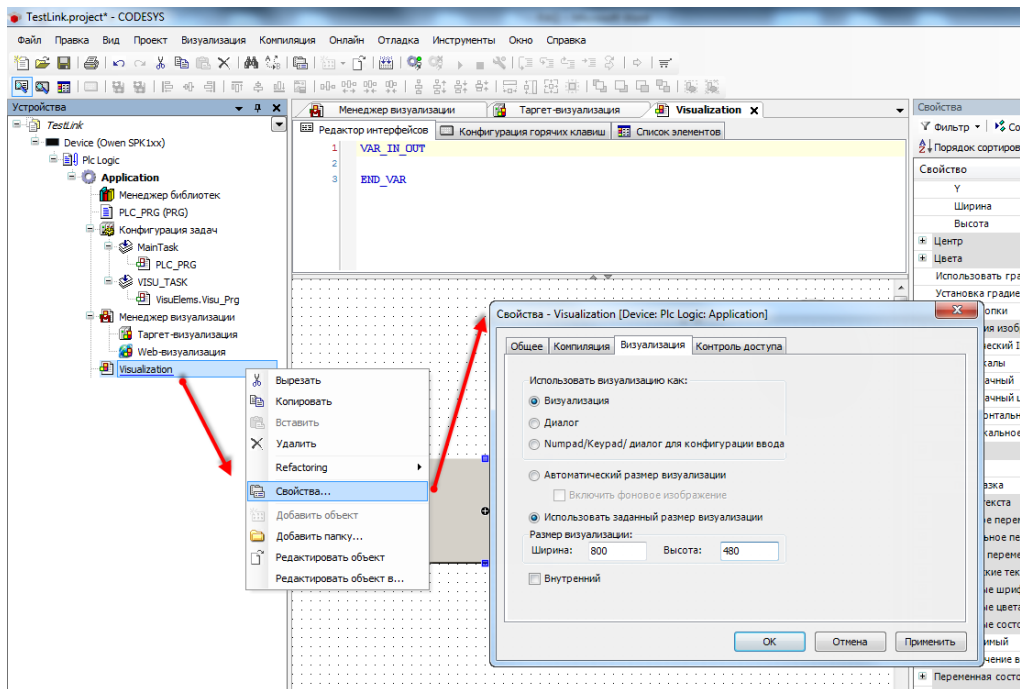


Рис. 8.7. Настройка разрешения экрана визуализации

8.6. Как отобразить значение параметра в визуализации?

Чтобы отобразить значение параметра в визуализации, необходимо:

1. объявить соответствующую переменную (в программе или **Глобальном списке переменных**):

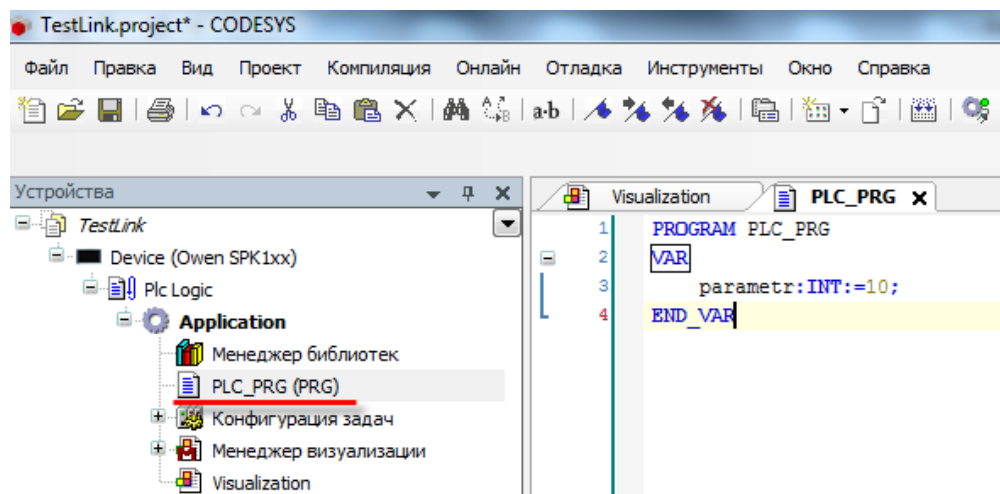


Рис. 8.8. Объявление переменной в программе PLC_PRG

2. привязать эту переменную к элементу визуализации и указать **формат вывода** (подробнее о форматах вывода см. п. 7.5.1 документа «СПК. Первый старт», доступного на [форуме ОВЕН в разделе СПК](#)):

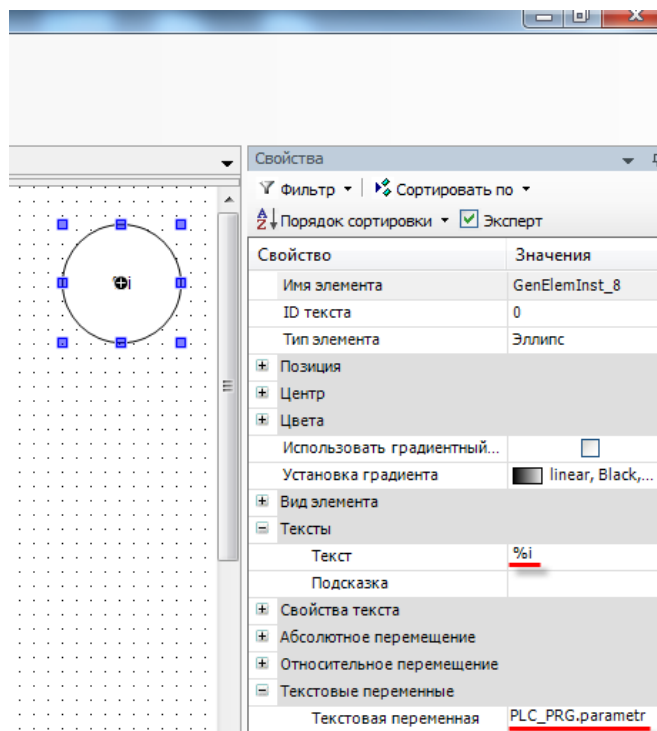


Рис. 8.9. Привязка переменной к элементу

В результате визуализация в процессе работы контроллера будет выглядеть следующим образом:

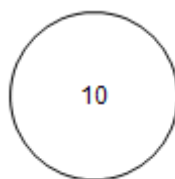


Рис. 8.10. Отображение значения параметра в визуализации

8.7. Как изменить значение параметра в визуализации с дисплея контроллера?

Изменить значение параметра в визуализации с дисплея контроллера можно различными способами; рассмотрим самый простой из них:

1. привяжем к элементу визуализации переменную и настроим формат ее вывода (см. [рис. 8.9.](#));
2. Выберем в настройках элемента вкладку **Inputconfiguration** и нажмем **ЛКМ** на поле **OnClick**. Откроется диалоговое окно **Конфигурации ввода**. Выберем на левой панели действие **Записать переменную**, с помощью кнопки «>» присвоим его выделенному полю ввода, и в настройках действия выберем **тип клавиатуры** (в нашем случае – цифровая) и записываемую переменную (по умолчанию используется переменная, привязанная к элементу). Нажмем **ОК**.

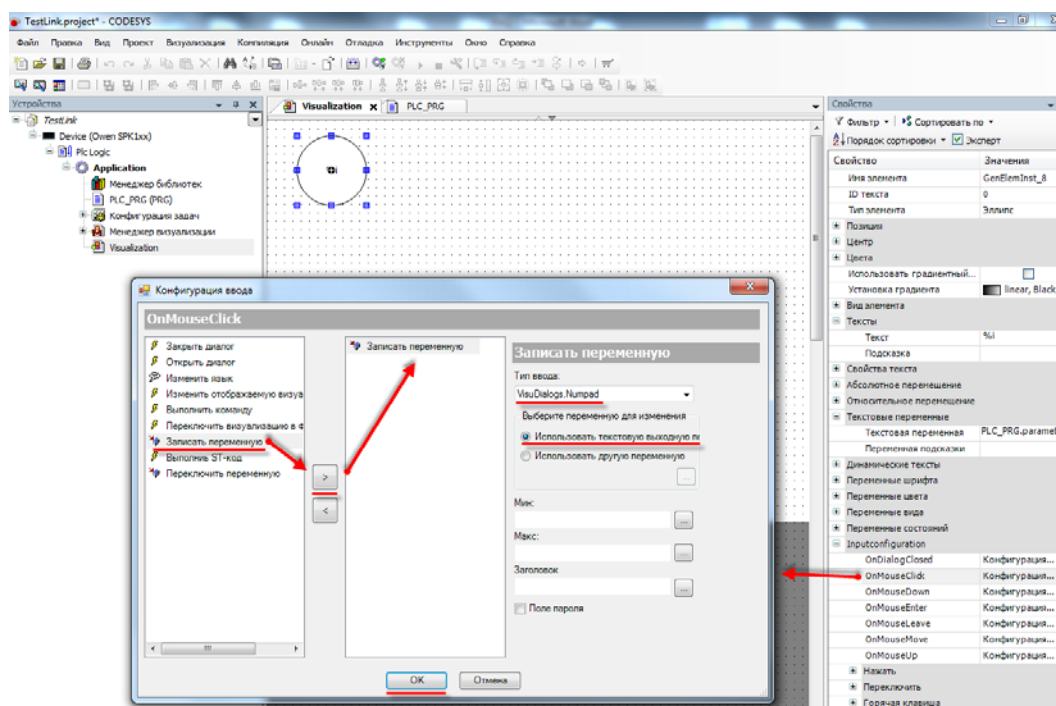


Рис. 8.11. Настройка действия для элемента

В результате на экране визуализации в процессе работы контроллера можно будет менять значение переменной с помощью **экранной клавиатуры**, вызываемой нажатием на элемент:

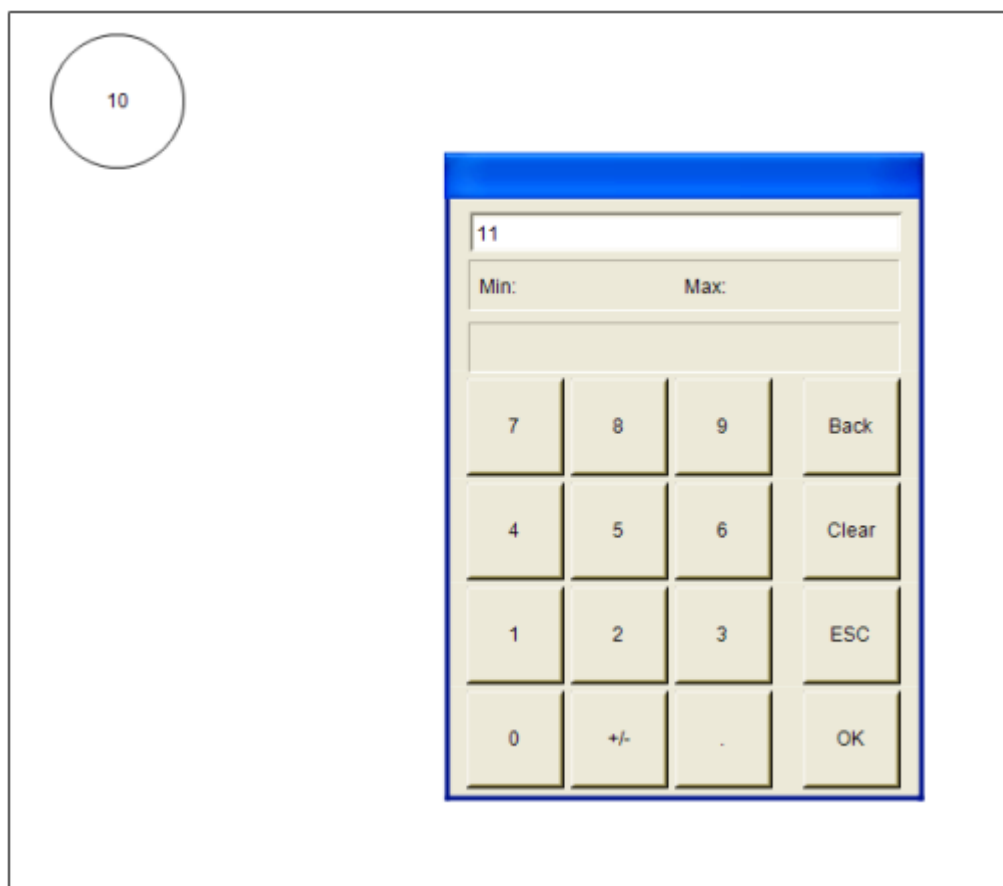


Рис. 8.12. Изменение значения параметра в процессе работы контроллера

8.8. Как переключать экраны визуализации?

1. Переключение экранов визуализации по нажатию кнопки

Предположим, проект содержит два экрана визуализации (**Visualization** и **Visualization2**), и необходимо реализовать переход с первого экрана на второй по нажатию кнопки. В свойствах кнопки необходимо выбрать вкладку **Inputconfiguration** и нажать **ЛКМ** на поле **OnMouseClicked**. Откроется диалоговое окно **Конфигурации ввода**. Выберем на левой панели действие **Изменить отображаемую визуализацию**, с помощью кнопки «>» присвоим его выделенному полю ввода, и в настройках действия выберем **экран визуализации** (в нашем случае – **Visualization2**). Нажмем **ОК**.

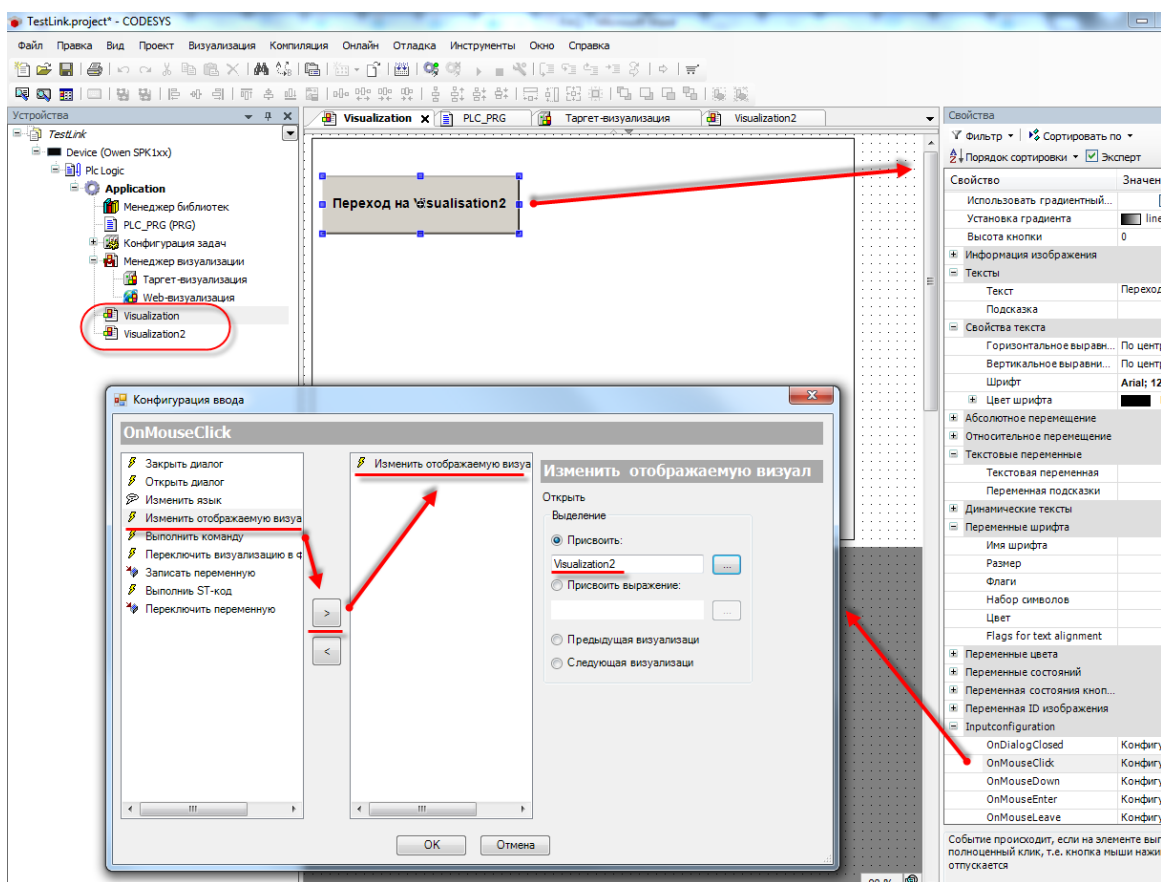


Рис. 8.13. Настройка действия для кнопки

Теперь в процессе работы контроллера можно будет перейти с одного экрана визуализации на другой по нажатию кнопки **Переход на Visualization2**.

2. Переключение визуализации из кода программы

Переключать переменные из кода программы можно с помощью строковой системной переменной **CurrentVisu**, которая определяет, какой из экранов отображается в данный момент. Соответственно, записывая в нее названия экранов визуализации, можно осуществлять переключения между ними.

Обратите внимание, что при использовании в проекте системной переменной **CurrentVisu**, переключение экранов будет затрагивать **всех клиентов визуализации**. Соответственно, при необходимости многопользовательского доступа, использование этой переменной крайне нежелательно.

Для того, чтобы включить переменную **CurrentVisu** в проект, необходимо поставить соответствующую галочку в установках **Менеджера визуализации**:

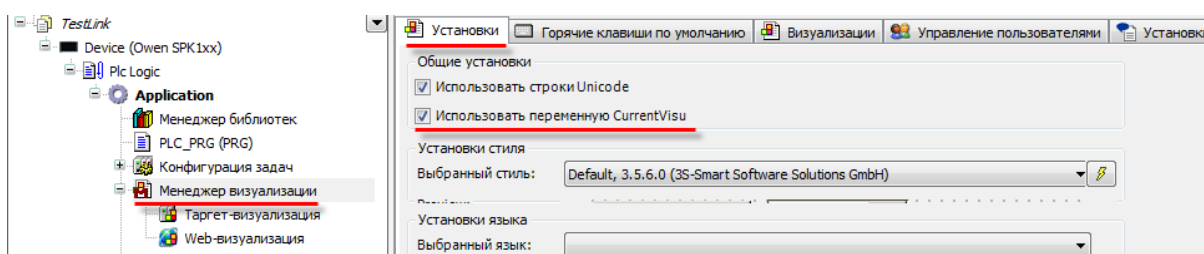


Рис. 8.14. Добавление в проект переменной **CurrentVisu**

Теперь можно переключать экраны визуализации из кода программы, например, подобным образом:

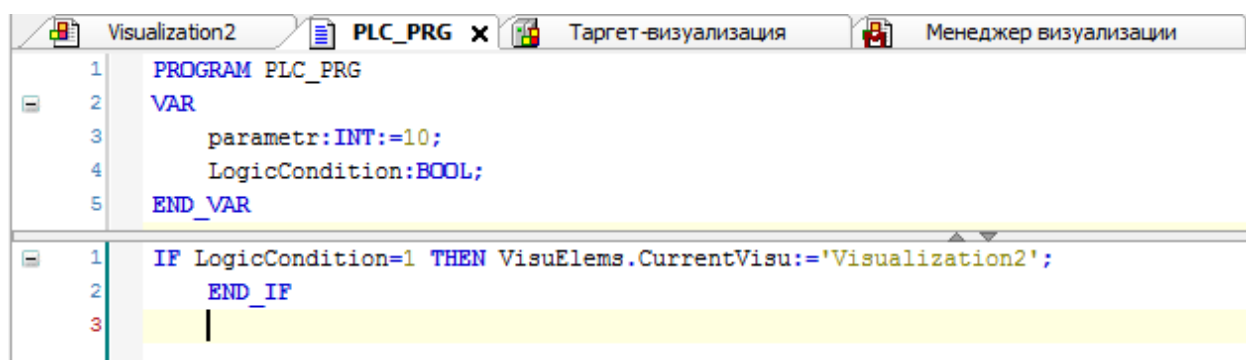


Рис. 8.15. Код переключения экранов визуализации

Если логическая переменная **LogicCondition** принимает значение **1 (TRUE)**, то осуществляется переключение текущего экрана визуализации на экран визуализации **Visualization2**.

Обратите внимание, что в коде программы обращение к системной переменной производится с указанием библиотеки: **VisuElems.CurrentVisu**. Обращение к переменной **CurrentVisu** (без указания библиотеки) не приведет к желаемому результату.

8.9. Почему русский текст на визуализации отображается «закорючками»?

Если вместо кириллических символов вы видите на дисплее контроллера что-то подобное:

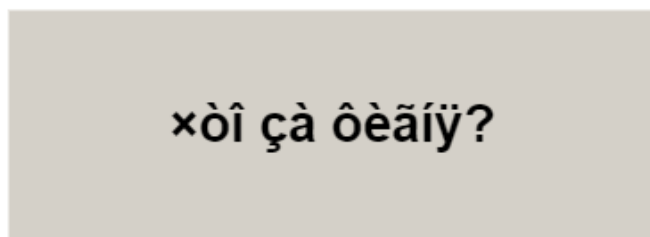


Рис. 8.16. Проблема с отображением кириллицы

то в CODESYS не включена поддержка **Юникода**. Включить ее можно в **Менеджере визуализации**:

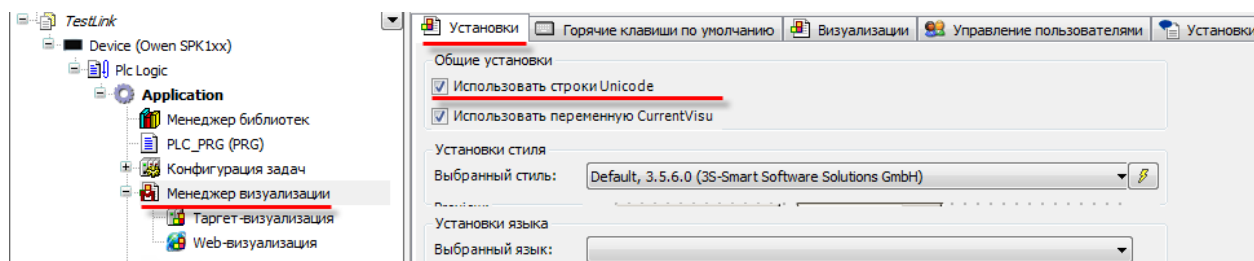


Рис. 8.17. Включение поддержки Юникода

После этого появится возможность отображения *статического* кириллического текста.

Если вам нужно отображать в визуализации кириллические **строковые переменные**, тогда следует использовать переменные типа **WSTRING**:

```
CyrilString:WSTRING:="Привет Кириллу и Мефодию";
```

Рис. 8.18. Объявление переменной типа **WSTRING**

Обратите внимание, что содержимое строки типа **WSTRING** указывается в **двойных** кавычках (содержимое строки типа **STRING** – в **одиночных**).

8.10. Как добавить в проект экранную клавиатуру с кириллицей?

Для изменения значений переменных типа **WSTRING** с дисплея контроллера необходима русскоязычная экранная клавиатура. В текущей версии **CODESYS** (3.5 SP6) такая клавиатура еще не реализована; ее появление ожидается в следующих версиях среды программирования.

8.11. Как настроить выполнение действия по нажатию на элемент?

Для выполнения действия по нажатию (или другим условиям – например, наведению курсора, закрытию диалогового окна и т.д.) на элемент, необходимо настроить в свойствах элемента вкладки **Inputconfiguration**.

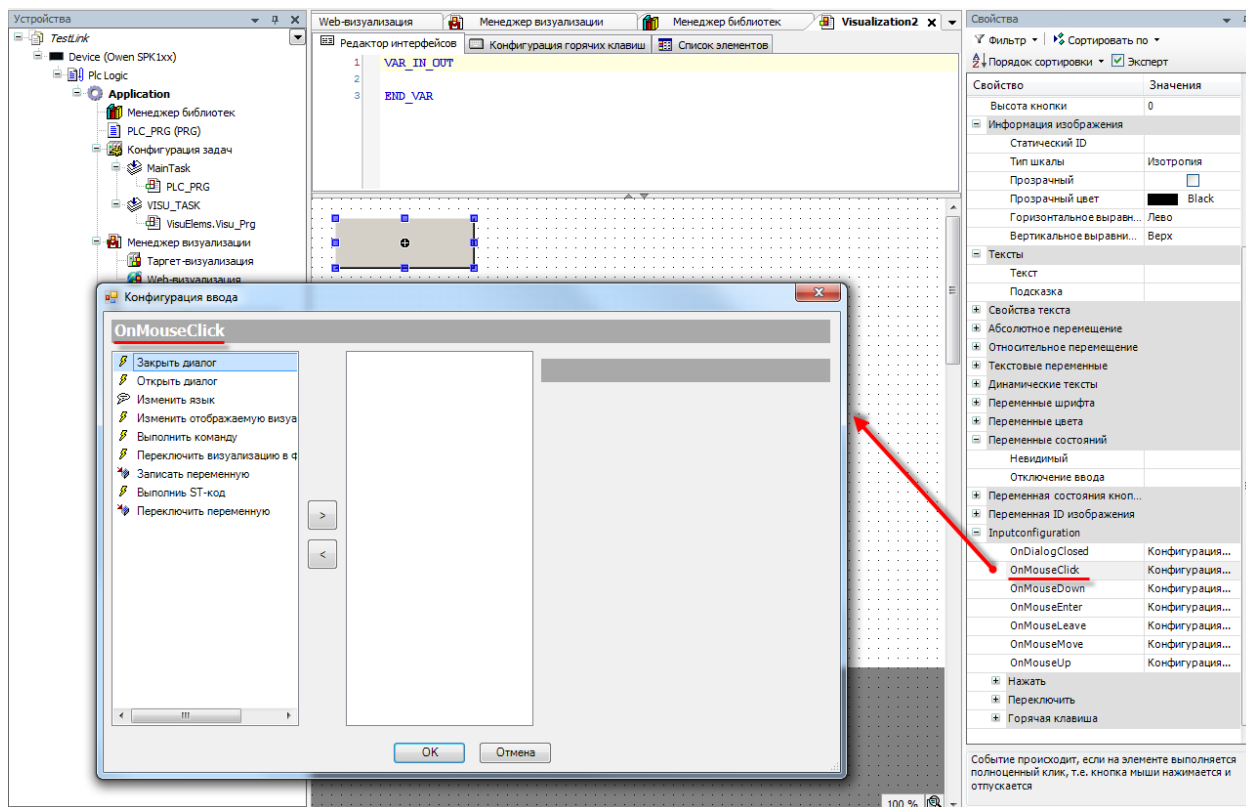


Рис. 8.19. Настройка действий элемента

Список доступных действий:

1. открытие диалогового окна;
2. закрытие диалогового окна;
3. изменение языка визуализации;
4. перехода на другой экран визуализации;
5. выполнение команды (например, загрузка рецепта из файла);
6. переключение фрейма в визуализации;
7. запись переменной;
8. выполнение кода на языке ST;
9. переключение логической переменной.

8.12. Можно ли добавить свой графический примитив в проект?

Добавить новый элемент на **Панель инструментов Редактора визуализации** нельзя; но существует возможность в некоторых пределах редактировать доступные шаблоны некоторых элементов. Проясним этот вопрос на примере элемента **Индикатор** (лампа):

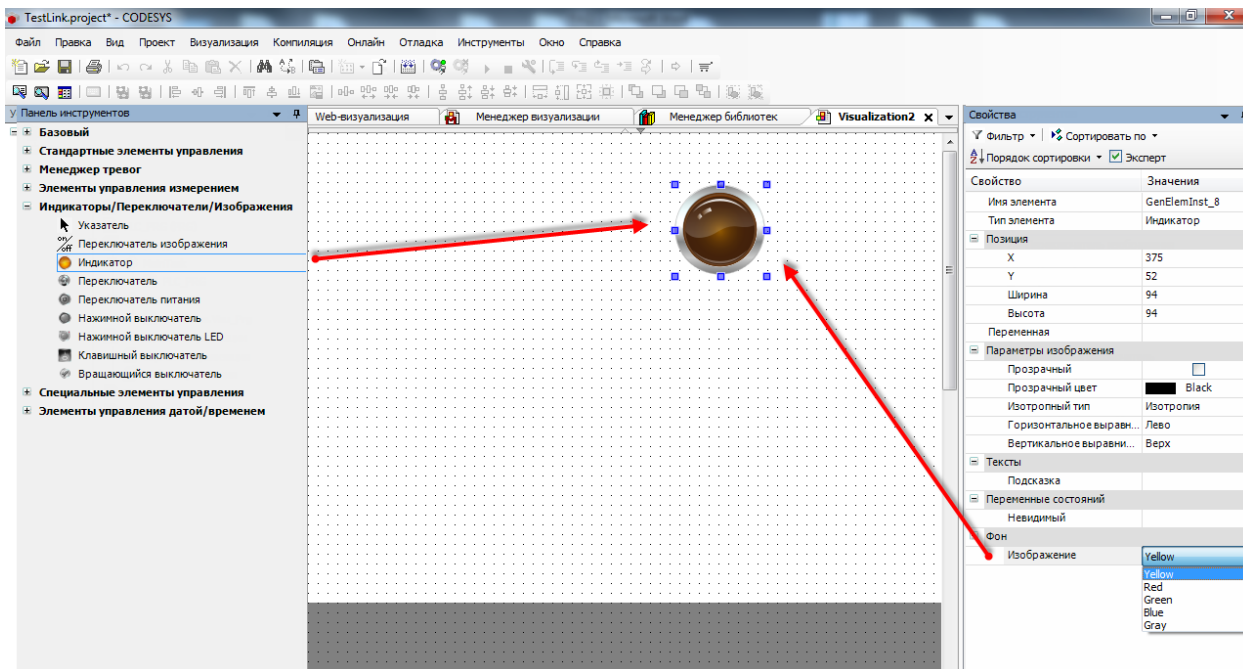


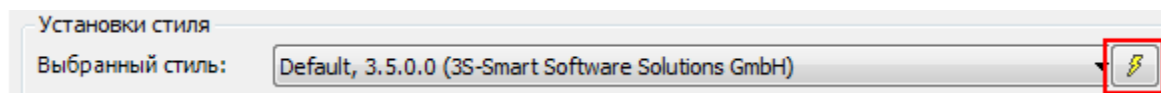
Рис. 8.20. Выбор цвета индикатора

Для этого элемента должно быть выбрано одно из пяти доступных фоновых изображений; если **логическая переменная**, привязанная к элементу, принимает значение **TRUE**, лампа загорается соответствующим цветом; если логическая переменная имеет значение **FALSE**, лампа выглядит потухшей, сохраняя соответствующий оттенок.

Нельзя создать свой индикатор, работающий по другому принципу (например, многопозиционный индикатор, к которому привязывается переменная типа **INT**, определяющая в зависимости от своего значения цвет элемента).

Можно создать свое (или отредактировать существующее) фоновое изображение – т.е., например, задать фиолетовый цвет для лампы, причем пользователь сам определяет, какой внешний вид лампа будет иметь при том или ином состоянии логической переменной (**TRUE** или **FALSE**).

Сделать это можно с помощью **Редактора стилей визуализации**. Чтобы открыть его, необходимо запустить программу **VisualStylesEditor.exe**, расположенную в папке **...\\3S CODESYS\\CODESYS\\Common**. Ее ярлык также расположен в **Менеджере визуализации**:



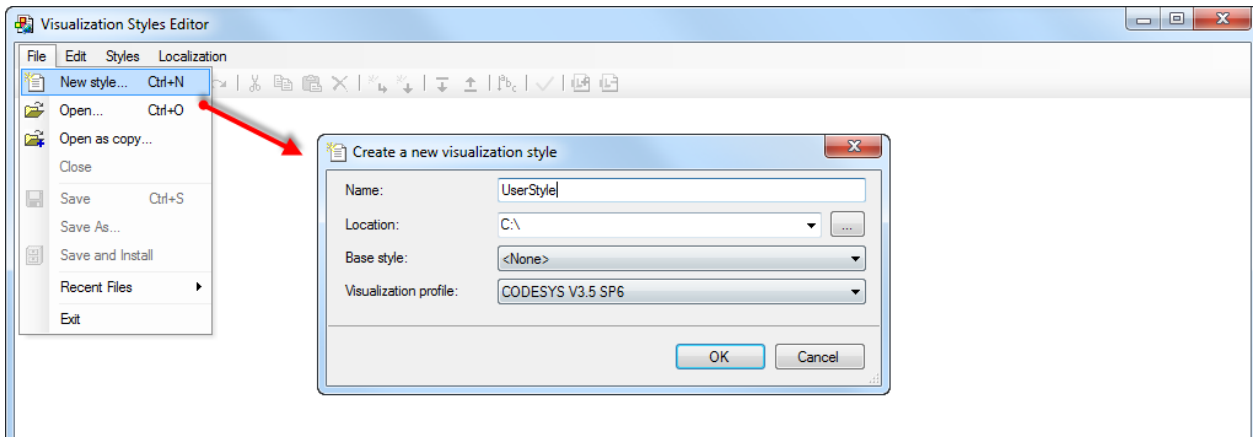


Рис. 8.21. Создание нового стиля визуализации

Для того, чтобы добавить созданный стиль визуализации в CODESYS, необходимо в меню **Инструменты** выбрать вкладку **Репозиторий стилей визуализации** и нажать кнопку **Установить**, после чего указать путь к файлу стиля:

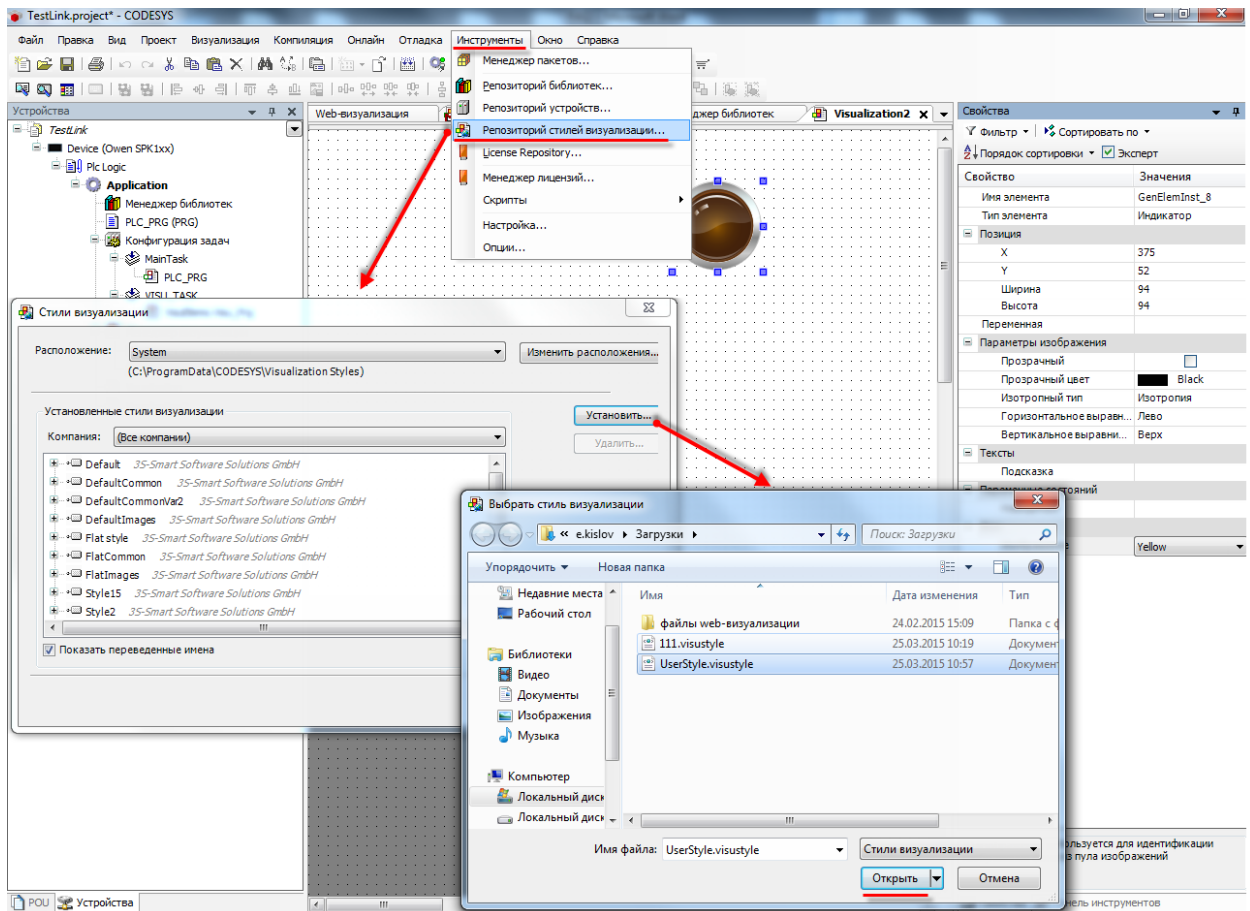


Рис. 8.22. Установка стиля визуализации в CODESYS

Чтобы использовать созданный стиль визуализации в проекте **CODESYS**, необходимо выбрать его в установках **Менеджера визуализации**:

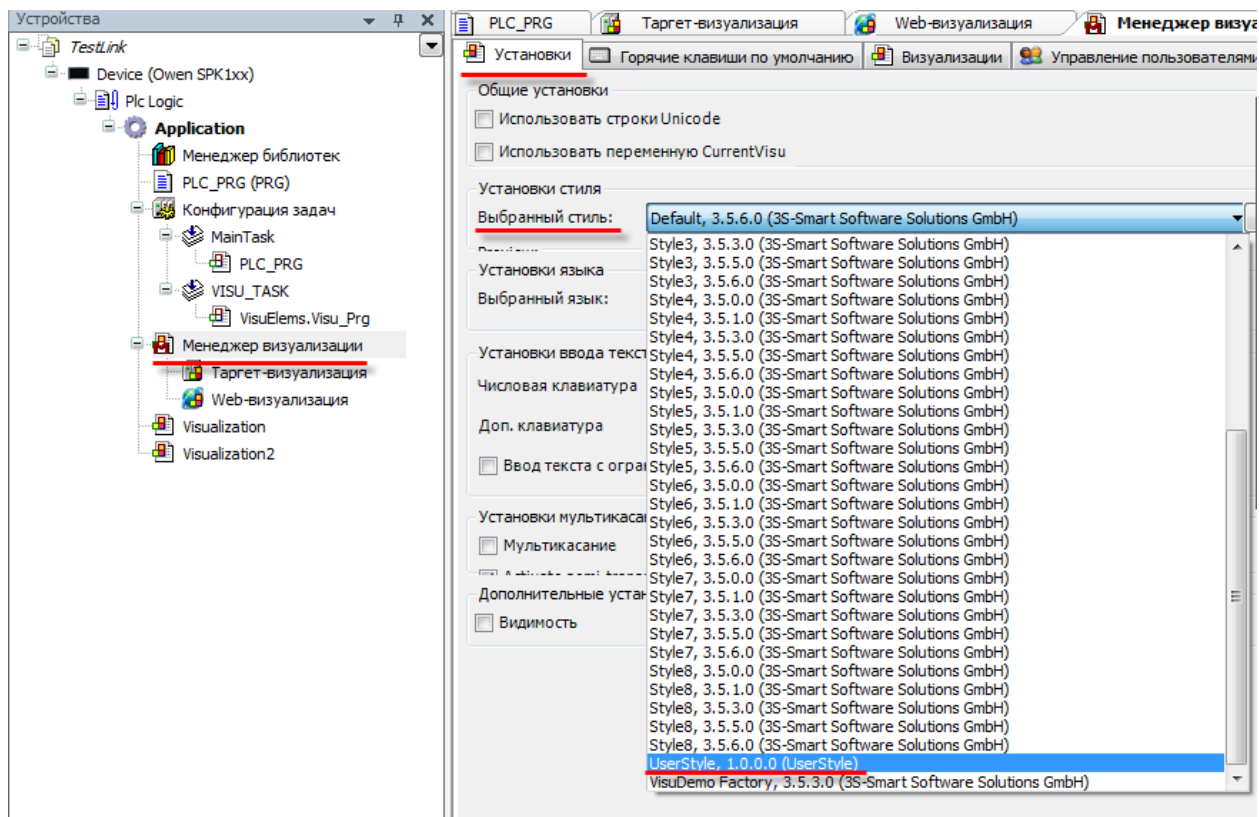


Рис. 8.23. Выбор стиля визуализации в проекте CODESYS

8.13. Как добавить графический файл в проект?

CODESYS позволяет загружать в проект пользовательские изображения, которые в дальнейшем могут использоваться в процессе разработки экранов визуализации (например, для создания фона экрана). Поддерживается большинство популярных форматов графических файлов, таких, как .jpg, .png, .bmp, .svg и т.д. **Обратите внимание**, что название файла **не должно** содержать **кириллических символов**.

Загрузка изображений осуществляется через компонент **Пул изображений**. Предварительно его нужно добавить в проект:

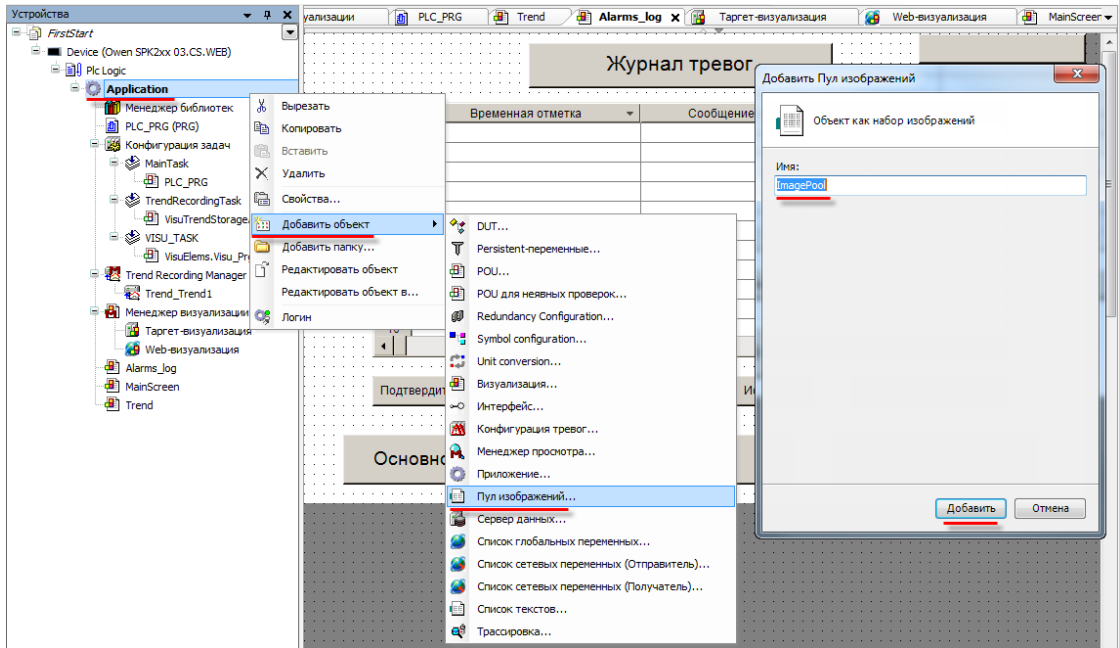


Рис. 8.24. Добавление Пула изображений

Пул изображений представляет собой таблицу следующего вида:

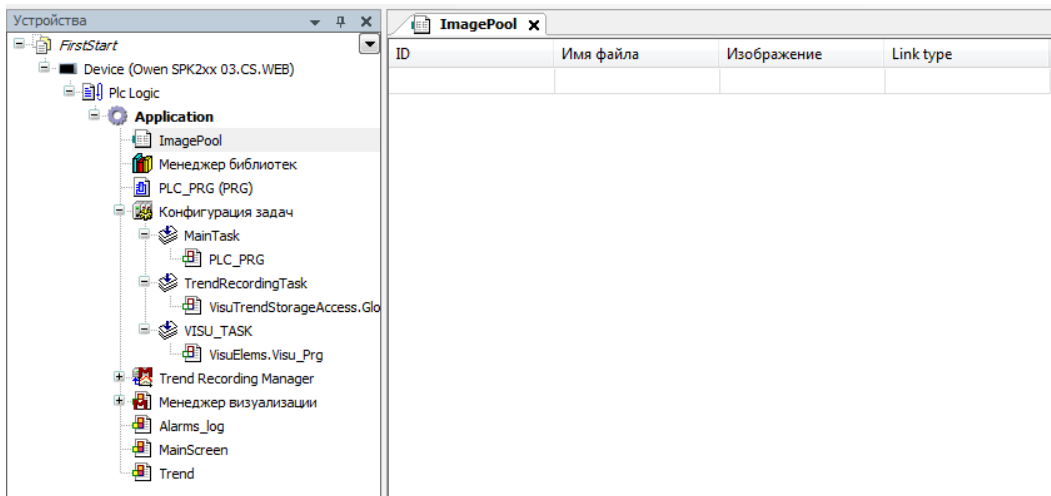


Рис. 8.25. Внешний вид Пула изображений

Для добавления изображения следует нажать **ЛКМ** на ячейку **Имя файла** и с помощью появившейся кнопки перейти к выбору файла:

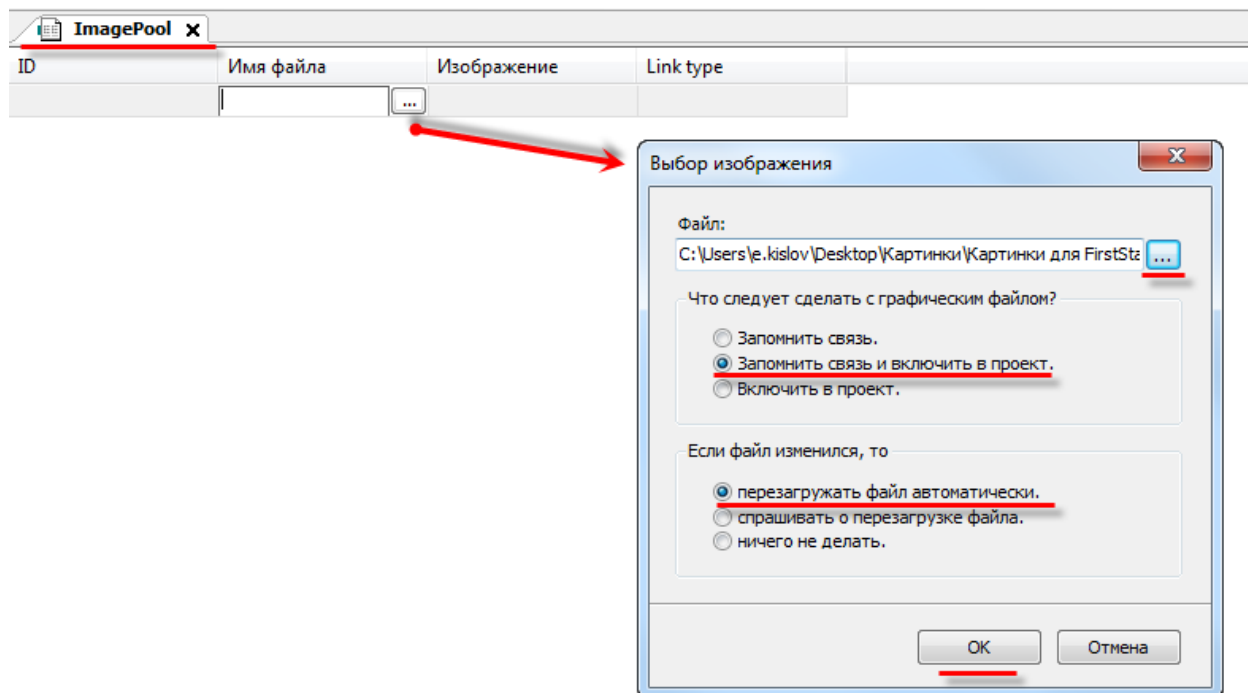


Рис. 8.26. Выбор изображения для загрузки

Теперь необходимо указать путь к графическому файлу. В расположенных ниже меню рекомендуется выбрать пункты **Запомнить связь и включить в проект** и **Перезагрузить файл автоматически**. Это позволяет не совершать дополнительных операций при изменении файла изображения – оно будет автоматически меняться в проекте.

После добавления изображения, его пиктограмма отобразится в **Пуле**; также рядом с ней будет указан **идентификатор (ID)** и **тип связи**.

Добавленные графические файлы можно использовать для:

1. создания фоновое изображения экрана визуализации;
2. создания статических изображений с помощью элемента **Изображение**;
3. создания динамических (переключаемых по значению логической переменной) изображений с помощью элемента **Переключатель изображений**;
4. создания пиктограмм для **Журнала тревог**.
5. фоновых изображений для некоторых элементов (Отображение Линейки, Кнопка).

Для использования графического файла в качестве **фонового изображения** необходимо нажать на любое место экрана визуализации **ПКМ** и в контекстном меню выбрать пункт **Фон**:

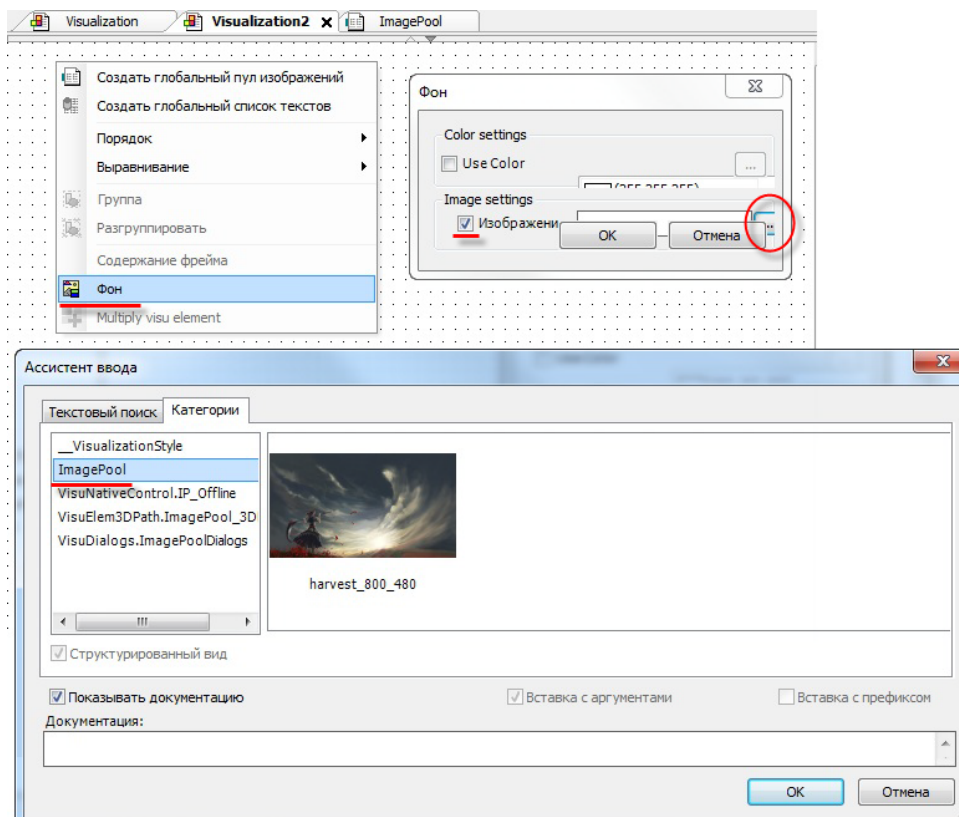


Рис. 8.27. Выбор фонового изображения

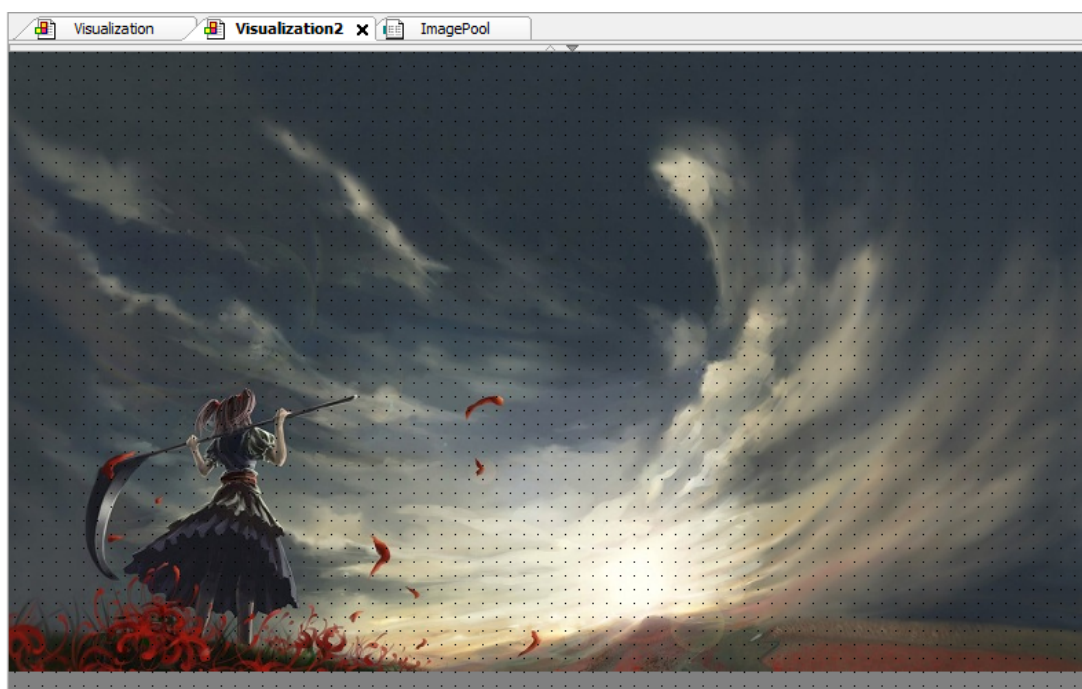
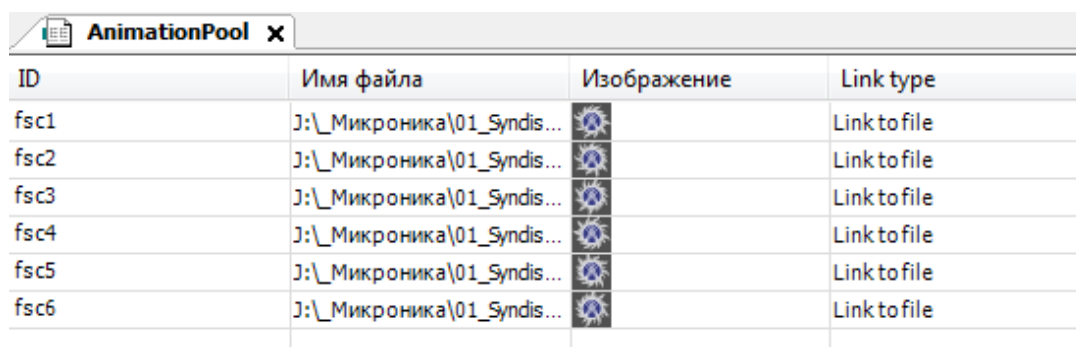


Рис. 8.28. Экран визуализации с фоновым рисунком

8.14. Как реализовать анимацию?

Специального элемента для реализации анимации в **CODESYS** на данный момент (**V3.5 SP6**) нет; анимацию можно создать искусственно, с помощью последовательного переключения заранее подготовленных изображений (**кадров** анимации). Вот одна из возможных реализаций (с помощью элемента **Изображение**):

1. Подготовить графические файлы с названиями, которые содержат порядковые номера кадров визуализации (без использования незначущих разрядов – т.е. animation1, animation10, animation100, но **НЕ** animation001, animation010, animation100). В качестве примера используем файлы fsc1 ... fsc6:









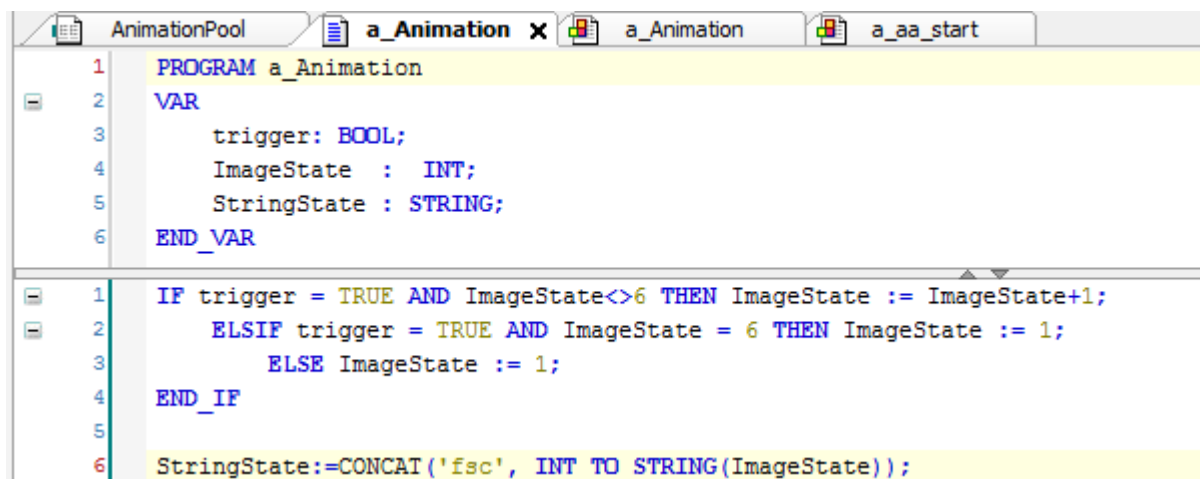
ID	Имя файла	Изображение	Link type
fsc1	J:_Микроника\01_Syndis...		Link to file
fsc2	J:_Микроника\01_Syndis...		Link to file
fsc3	J:_Микроника\01_Syndis...		Link to file
fsc4	J:_Микроника\01_Syndis...		Link to file
fsc5	J:_Микроника\01_Syndis...		Link to file
fsc6	J:_Микроника\01_Syndis...		Link to file

Рис. 8.29. Пул изображений с кадрами анимации

2. В коде программы использовать следующую конструкцию:



```
1 PROGRAM a_Animation
2 VAR
3     trigger: BOOL;
4     ImageState : INT;
5     StringState : STRING;
6 END_VAR
7
8 IF trigger = TRUE AND ImageState <> 6 THEN ImageState := ImageState + 1;
9   ELSIF trigger = TRUE AND ImageState = 6 THEN ImageState := 1;
10  ELSE ImageState := 1;
11 END_IF
12
13 StringState := CONCAT('fsc', INT_TO_STRING(ImageState));
```

Рис. 8.30. Код смены кадров

где **Trigger** – логическая переменная для запуска анимации (по значению **TRUE**);

ImageState – порядковый номер кадра анимации;

StringState – переменная, определяющая текущий кадр анимации, которая будет привязана к элементу **Изображение**.

Рекомендуется создать для анимации отдельную программу и задачу; настраивая время цикла задачи, мы будем определять **FPS** (количество кадров в секунду) анимации.

3. На экране визуализации добавим элемент **Изображение** и в его **Свойствах** во вкладке **Переменная ID изображения** привяжем переменную **StringState** из нашей программы **a_Animation**:

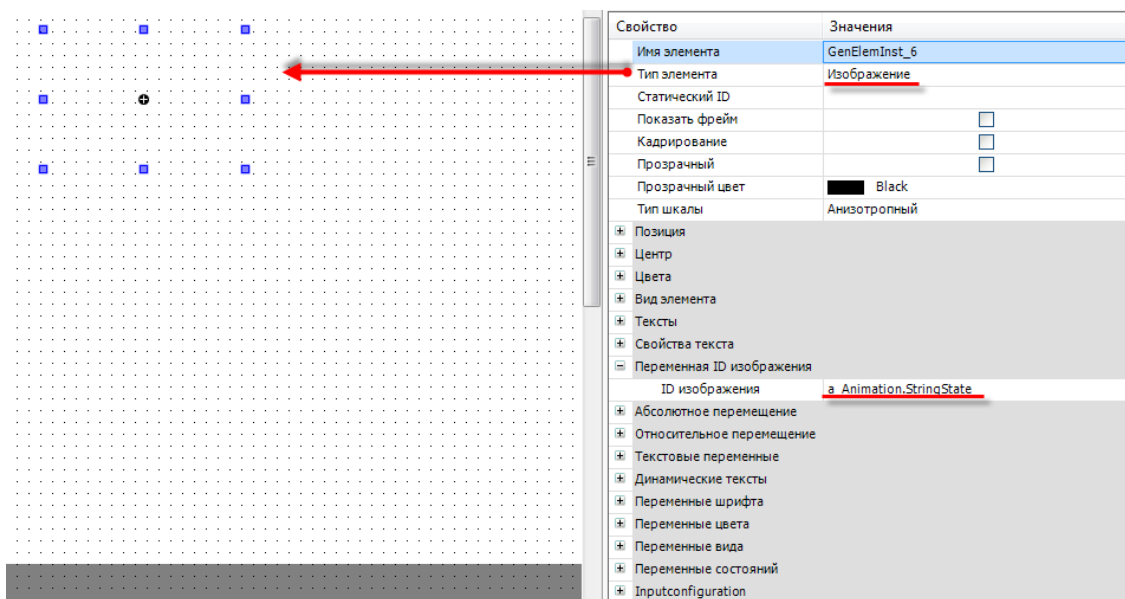


Рис. 8.31. Привязка переменной к элементу **Изображение**

На этом реализация анимации завершена. Она будет работать следующим образом: если переменная **Trigger** принимает **TRUE**, в элементе **Изображение** начинают последовательно отображаться файлы **fsc1**, **fsc2** и т.д. (частота смены файлов определяется временем цикла соответствующей задачи). После отображения последнего файла (в нашем случае – шестого) опять отображается первый, затем второй и т.д. Если переменная **Trigger** принимает значение **FALSE**, элемент **Изображение** начинает статично отображать первый кадр. При необходимости в этом случае останавливать изображение на последнем отображенном кадре, достаточно удалить из программы строку **ELSE ImageState := 1;**

8.15. Как использовать в визуализации многострочный текст?

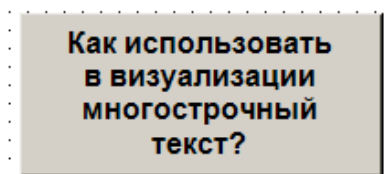


Рис. 8.32. Многострочный текст

Переход на следующую строку при наборе текста осуществляется комбинацией клавиш **Ctrl+Enter**.

8.16. Как создать текст с вертикальной ориентацией?

На данный момент такая возможность отсутствует, но можно сменить ориентацию дисплея СПК в [конфигураторе](#).

8.17. Почему в Таблице тревог не отображается история?

В первую очередь, необходимо отметить, что история тревог **не работает** в режиме эмуляции – для ее проверки нужно использовать либо СПК, либо [виртуальный контроллер](#).

Для записи истории тревог необходимо:

1. Наличие галочки **Архивация** у классов тревог, которые должны записываться в историю;

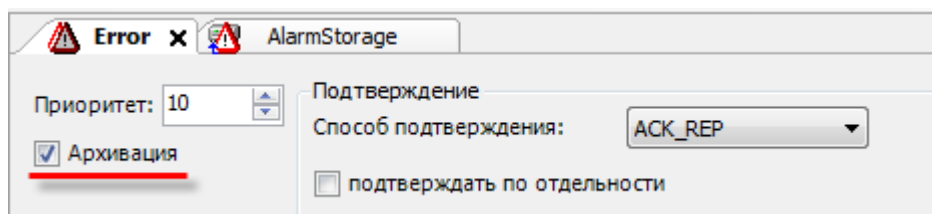


Рис. 8.33. Настройка архивации для класса тревог

2. Указание **Хранилища тревог** для группы тревог, которые должны записываться в историю.

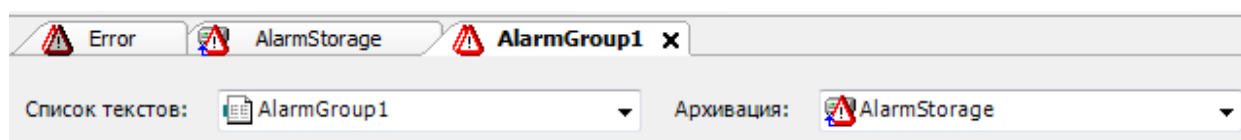


Рис. 8.34. Настройка архивации для группы тревог

8.18. Как записывать историю тревог на flash- или SD-карту памяти?

Данный функционал в настоящее время не реализован.

8.19. Сколько переменных можно отображать на тренде?

Число отображаемых (и, соответственно, архивируемых) на тренде переменных можно настроить во вкладке **Trend Storage**; по умолчанию оно равно 10, максимально возможное значение – 199. Но для корректной работы контроллера **крайне рекомендуется** не увеличивать значение этого параметра.

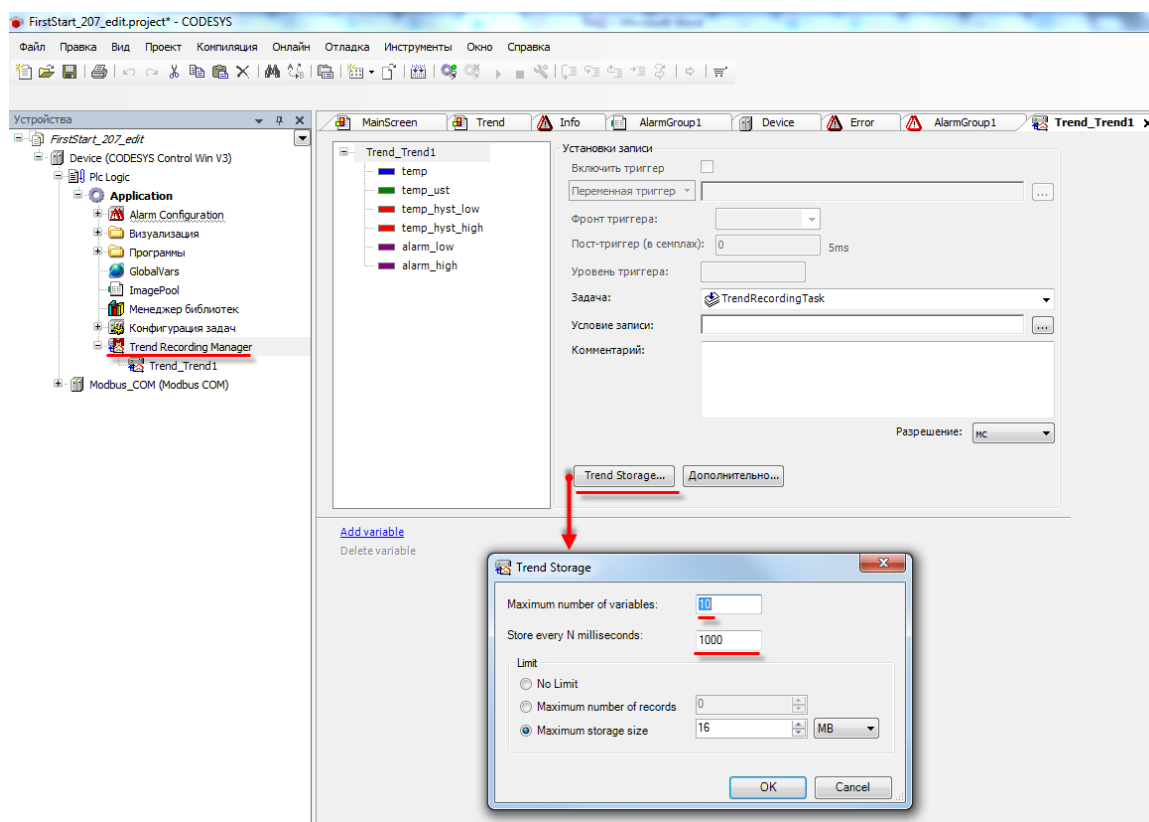


Рис. 8.35. Настройки архивации тренда

8.20. Какова частота записи переменных тренда?

Частоту записи переменных тренда можно настроить во вкладке **Trend Storage** (см. рис. 8.35); по умолчанию она равна 1000 мс. Для корректной работы контроллера **крайне рекомендуется** не уменьшать значение этого параметра.

8.21. Как просмотреть историю тренда?

Для просмотра истории тренда необходимо добавить **Селектор диапазона дат**. Для этого нужно нажать на тренд **ПКМ** и выбрать пункт **Вставить компоненты для управления трендом**:

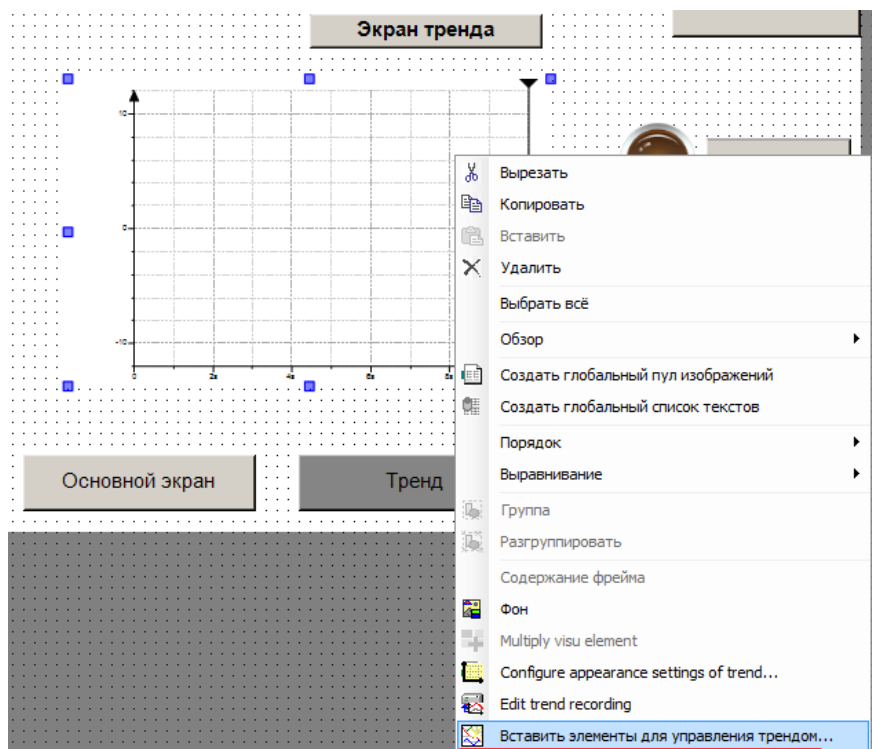


Рис. 8.36. Добавление элементов управления трендом

Настройки **Мастера трендов** можно оставить по умолчанию:

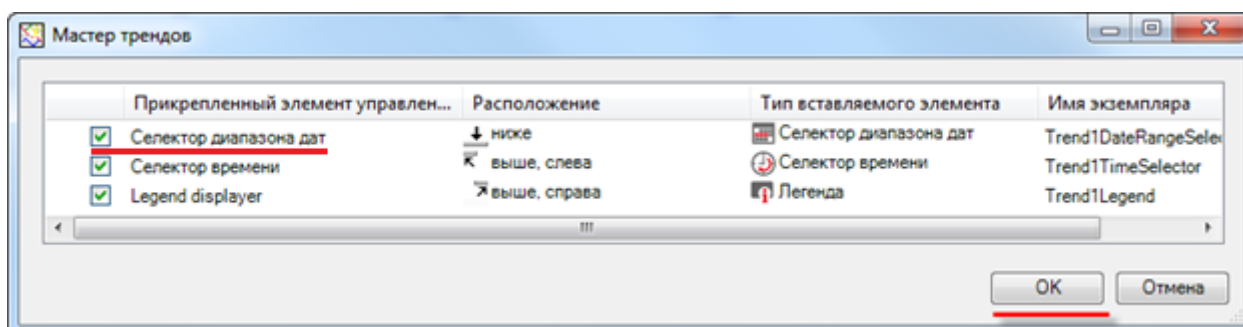


Рис. 8.37. Меню Мастера трендов

В процессе работы проекта для просмотра истории следует перемещать ползунок селектора диапазона дат:

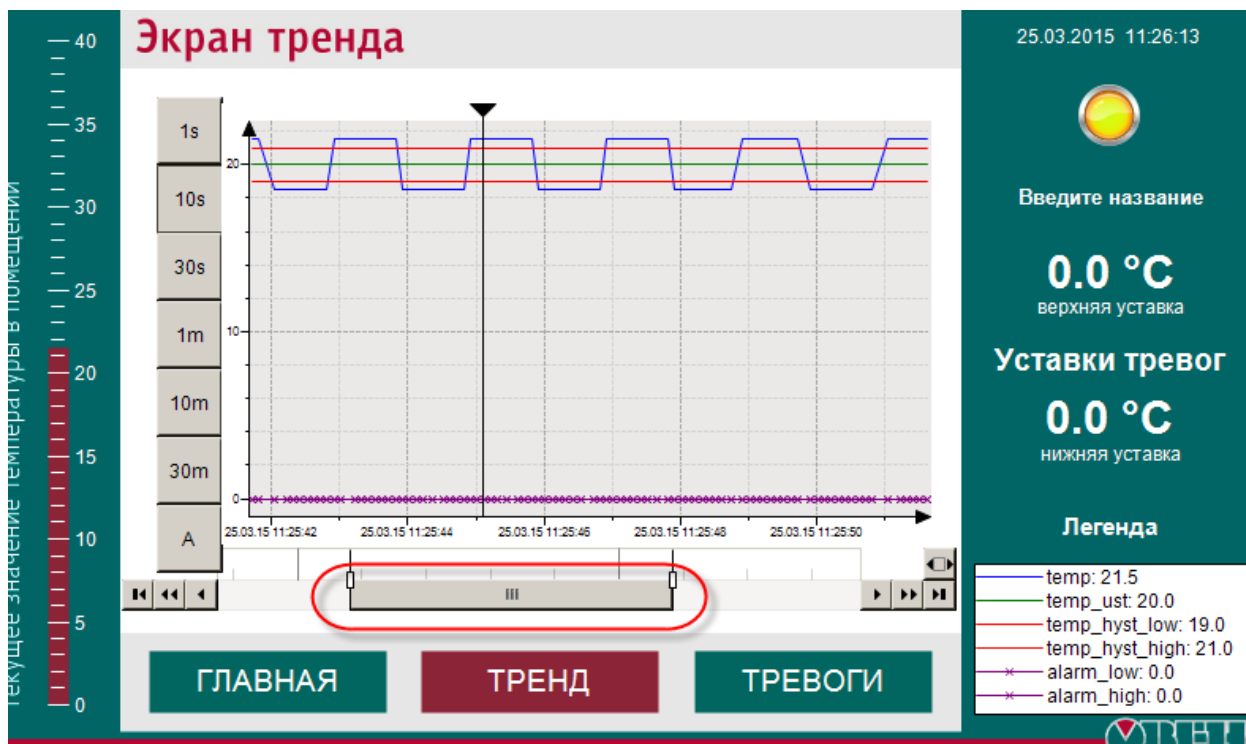



Рис. 8.38. Просмотр истории тренда

Для возвращения к режиму обновления тренда в реальном времени, необходимо нажать соответствующую кнопку: .

8.21. Как записывать историю тренда на flash- или SD-карту памяти?

Данный функционал в настоящее время не реализован.

8.22. Как создать мультиязычный проект?

Для создания **мультиязычного** проекта (т.е. проекта, язык текстовых надписей которого можно переключать в процессе работы) следует воспользоваться компонентом **Список текстов**:

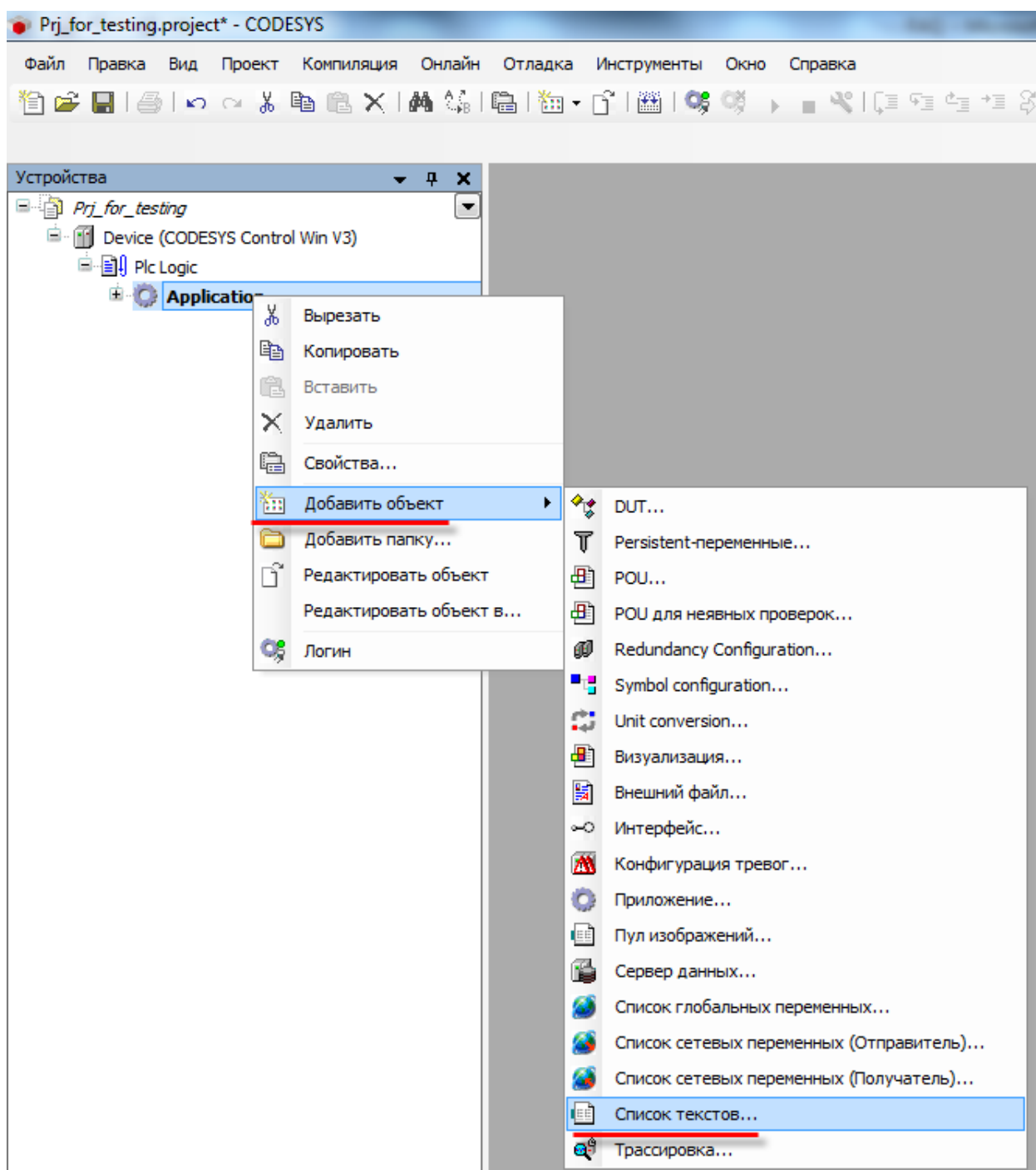


Рис. 8.39. Добавление списка текстов

Для того, чтобы добавить в **Список текстов** новый язык, следует нажать **ПКМ** на любую ячейку и в контекстном меню выбрать пункт **Добавить язык**:

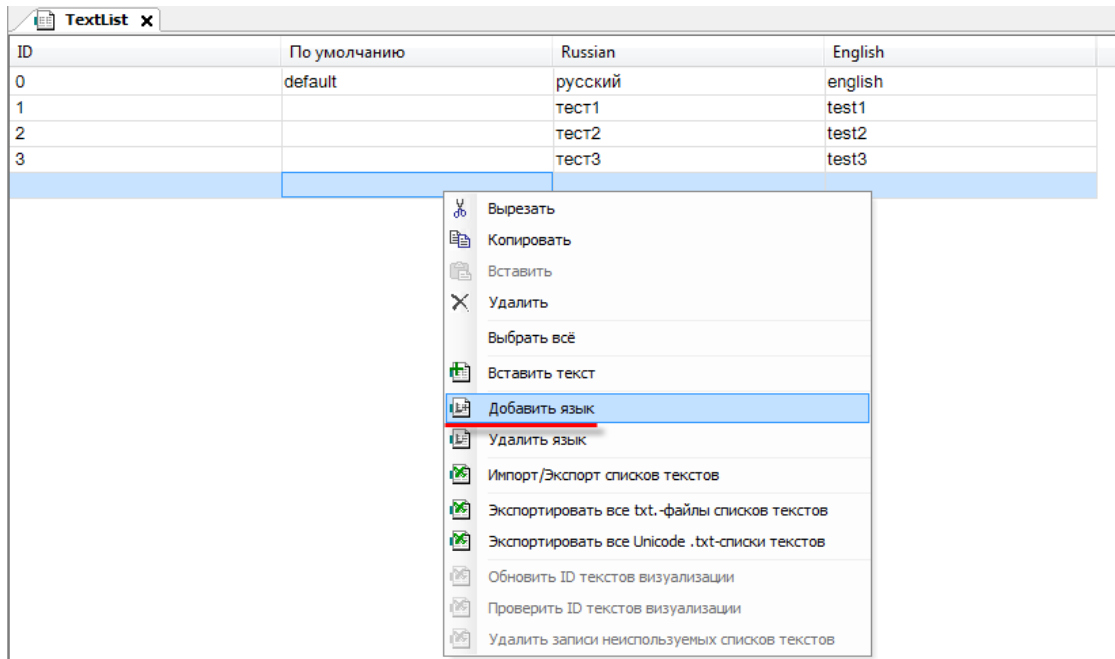


Рис. 8.40. Добавление языка в список текстов

Настроим кнопку смены языка следующим образом:

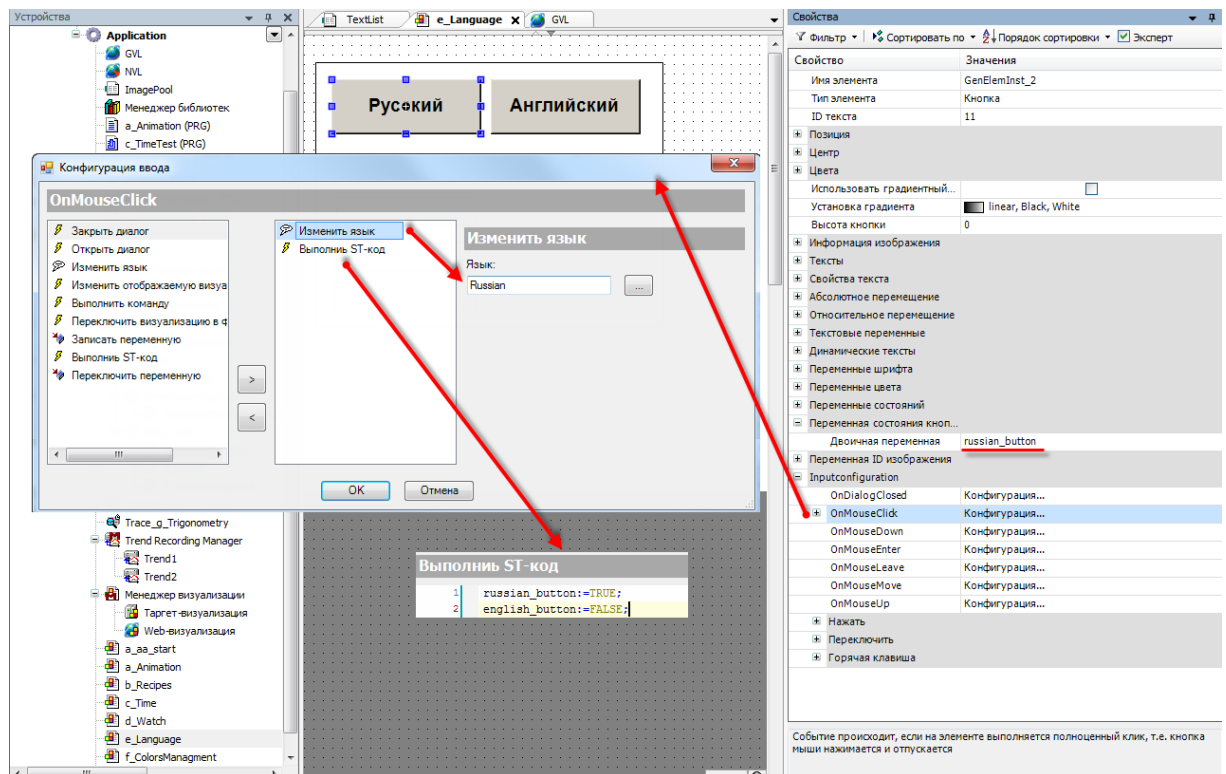


Рис. 8.41. Настройка кнопки переключения языка

Очевидно, что для кнопки выбора английского языка настройки будут следующие:

Двоичная переменная: **english_button**

Язык: **english**

ST-code: **english_button := TRUE;**

russian_button := FALSE;

В результате в процессе работы проекта можно будет переключить язык надписей с русского на английский и обратно:

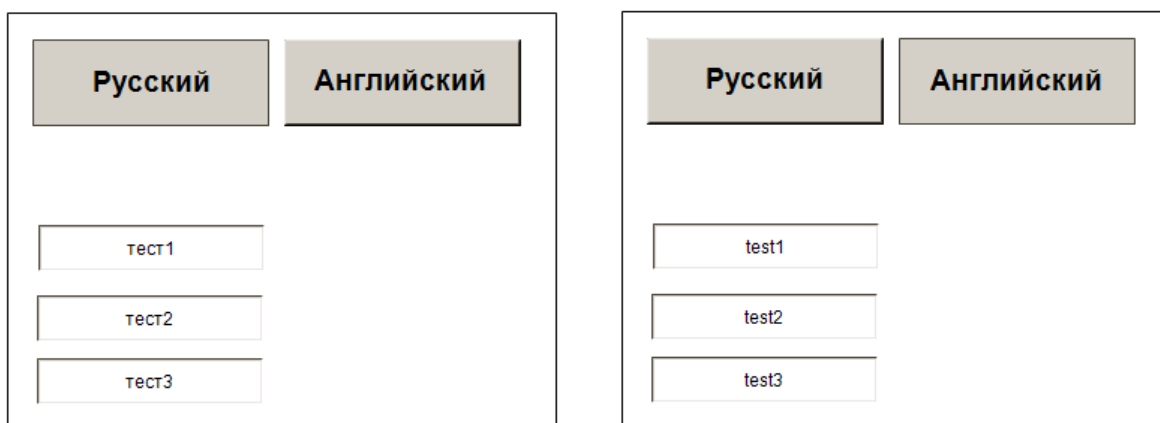


Рис. 8.42. Смена языка в процессе работы контроллера
(слева – выбран русский язык, справа – английский)

8.23. Как записать информацию в файл (архивирование данных) ?

Для ведения архива рекомендуется использовать библиотеку **ArchiverSPK**, которая находится на диске с ПО из комплекта поставки. Последнюю версию библиотеки и инструкцию по ее использованию можно найти на [форуме ОВЕН в разделе СПК](#).

Основные возможности библиотеки:

- запись архива в формате CSV (формат, совместимый с Excel);
- число архивируемых переменных: от 1 до 64;
- поддерживаемые архиватором типы переменных: DWORD, DINT, REAL, STRING;
- возможность выбора устройства, на которое ведется запись данных (внутренняя память контроллера или USB/SD запоминающее устройство);
- копирование архива на другой носитель;
- возможность одновременно архивировать переменные разных типов с различным периодом.

9. Вопросы по загрузке, запуску, отладке проекта CODESYS

9.1. Как загрузить проект в контроллер?

Для загрузки проекта необходимо наладить связь между СПК и **CODESYS** (см. [п. 7.1.](#)). Если связь налажена, то достаточно выполнить команды из меню **Онлайн**: **Логин** - для загрузки проекта в оперативную память и **Создать загрузочное приложение** - для загрузки проекта во flash-память контроллера (энергонезависимую).

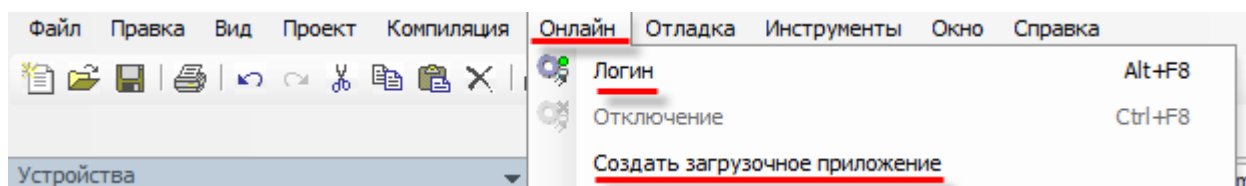


Рис. 9.1. Основные команды меню Онлайн

9.2. Почему проект исчезает после загрузки контроллера?

Потому что он был записан в оперативную память, которая очищается после перезагрузки контроллера. Необходимо залить проект во flash-память с помощью команды **Создать загрузочное приложение** (см. рис. 9.1.).

9.3. Что делать, если при подключении к контроллеру возникает окно с запросом пароля?

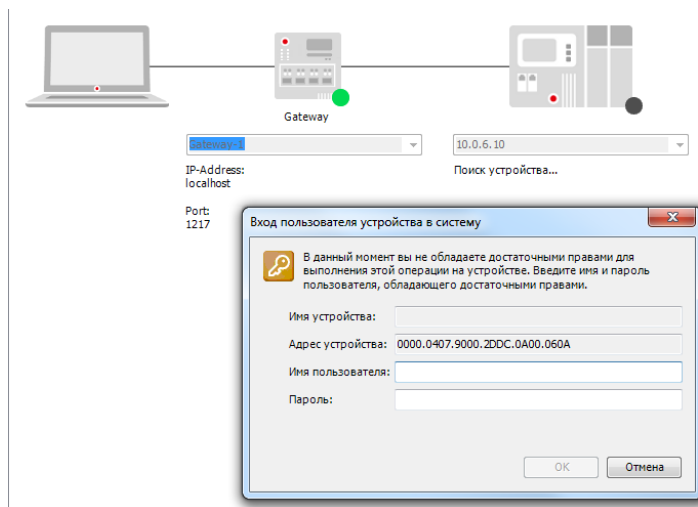


Рис. 9.2. Окно ввода пароля при подключении к контроллеру

Диалоговое окно ввода пароля появляется во время подключения к контроллеру с запущенным [конфигуратором](#); в этом случае необходимо перезагрузить контроллер и дождаться загрузки проекта (в случае, если проект отсутствует – появления окна **Отсутствует загрузочное приложение**).

Так же это окно может появляться при подключении к контроллеру, проект CODESYS которого защищен паролем; в этом случае следует связаться с разработчиками проекта.

9.4. Как удалить проект из контроллера?

Каждый заливаемый в контроллер проект полностью перезаписывает предыдущий; если необходимо удалить проект из контроллера, это можно сделать с помощью команды **Сброс заводской** из меню **Онлайн**:

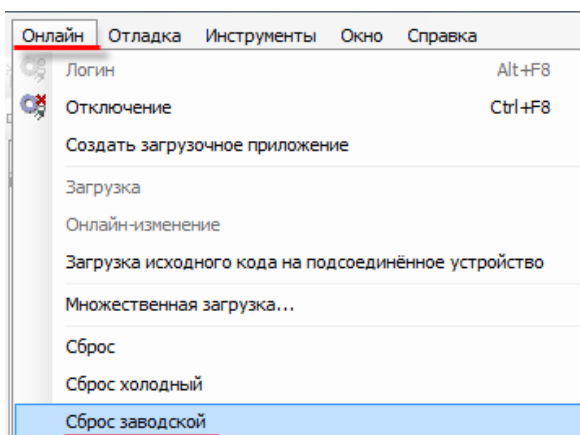


Рис. 9.3. Удаление проекта из контроллера

9.5. Как выгрузить проект с контроллера?

Для того, чтобы иметь возможность выгрузить проект с контроллера, необходимо на этапе загрузки проекта в контроллер произвести **Загрузку исходного кода**:

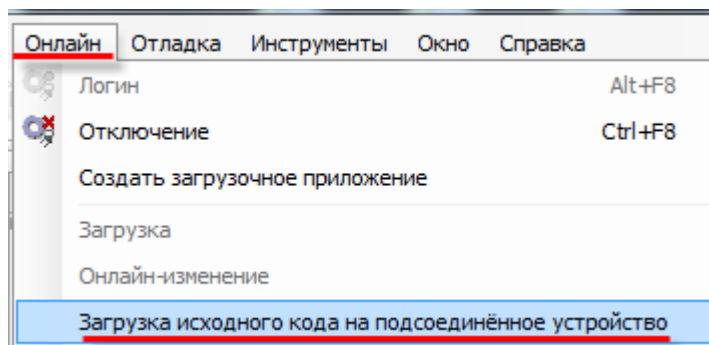


Рис. 9.4. Команда загрузки исходного кода проекта в контроллер для возможности последующей выгрузки

В случае, если исходный код был загружен в контроллер, можно выгрузить его в **CODESYS** с помощью одноименной команды из меню **Файл**:

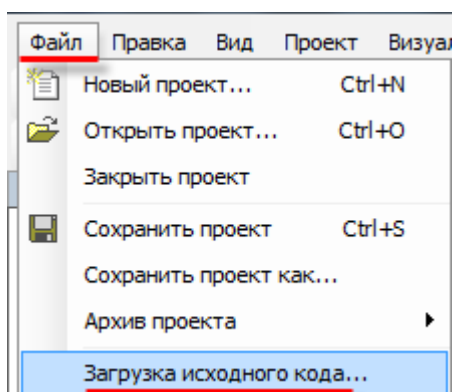


Рис. 9.5. Команда выгрузки проекта с контроллера

9.6. Почему после загрузки проекта на дисплее СПК отображается надпись «Отсутствует загрузочное приложение»?

После выполнения команд **Логин/Создание загрузочного приложения** программа загружается в контроллер, но не запускается. Для ее запуска необходимо выполнить команду **Старт** из меню **Отладка**:

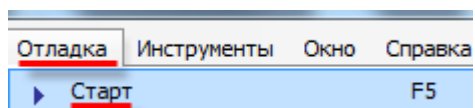


Рис. 9.6. Команда запуска проекта

Если проект загружен во flash-память контроллера (с помощью команды **Создать загрузочное приложение**), то загруженная программа автоматически запустится после перезагрузки.

Текущее состояние программы отображается в **статусной строке** CODESYS, расположенной внизу экрана:

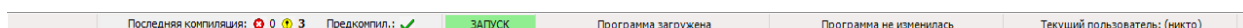


Рис. 9.7. Строка состояния запущенного проекта

9.7. Почему написанный код не выполняется?

В значительном количестве случаев такое может происходить, если программа не привязана к задаче. Иными словами, программа находится в контроллере, но не вызывается на исполнение.

Необходимо добавить компонент **Конфигурация задач** (обычно он присутствует в проекте по умолчанию), создать с его помощью новую задачу и привязать к ней программу:

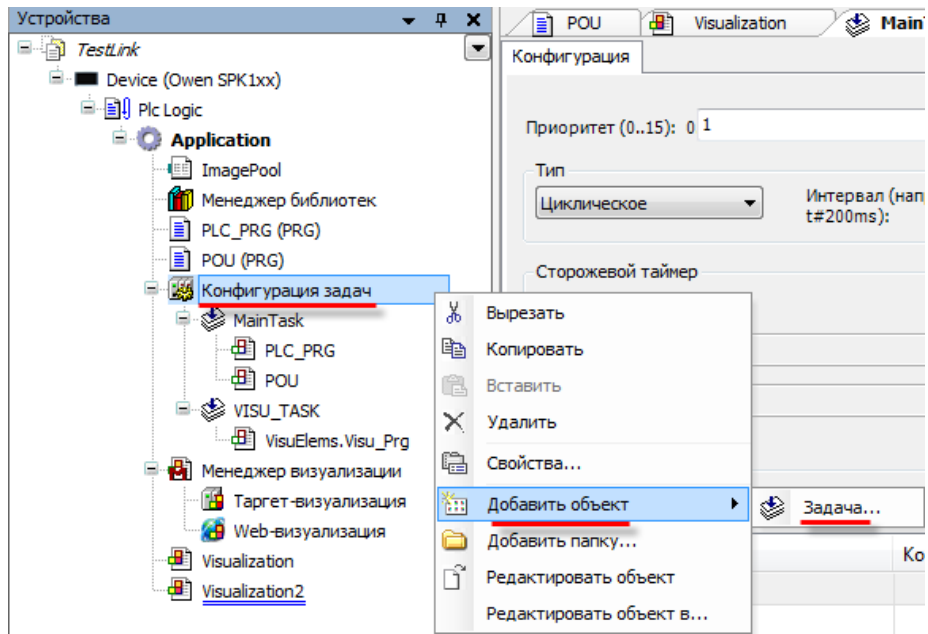


Рис. 9.8. Добавление задачи

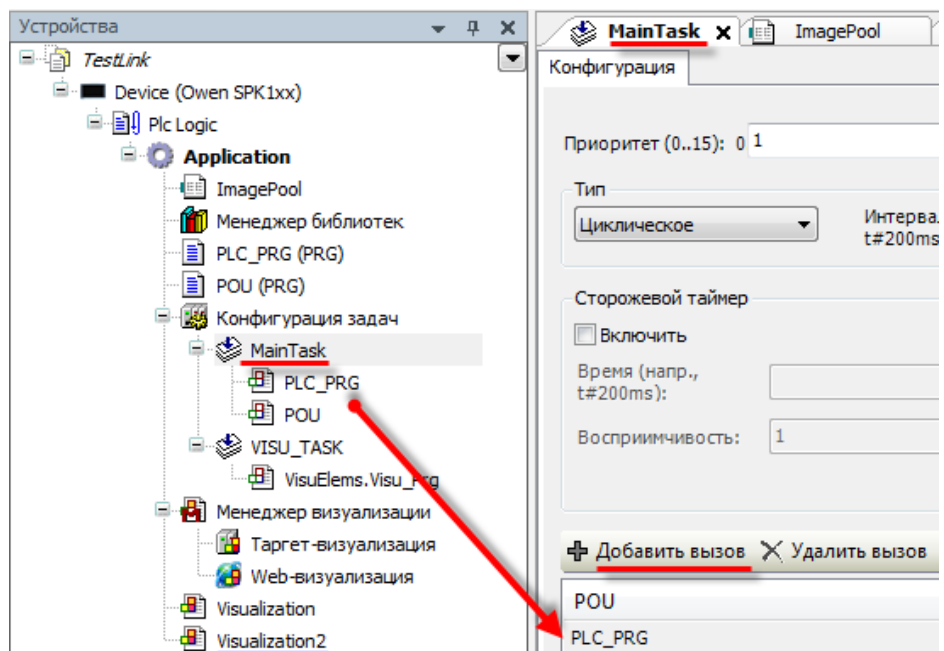


Рис. 9.9. Привязка программы к задаче

9.8. Как отладить проект без контроллера?

В случае отсутствия реального контроллера можно произвести отладку проекта средствами **CODESYS**. Существует два средства отладки – **режим эмуляции** в CODESYS и использование **виртуального контроллера**.

Режим эмуляции позволяет проверить только базовые действия (например, вывод значений) и не рекомендуется к использованию. Его можно включить с помощью команды **Эмуляция** из меню **Отладка**;

Виртуальный контроллер представляет собой программную эмуляцию реального контроллера, запускаемую на ПК с ОС семейства Windows. Для его запуска необходимо:

1. Запустить виртуальный контроллер с помощью иконки на **панели задач** Windows:

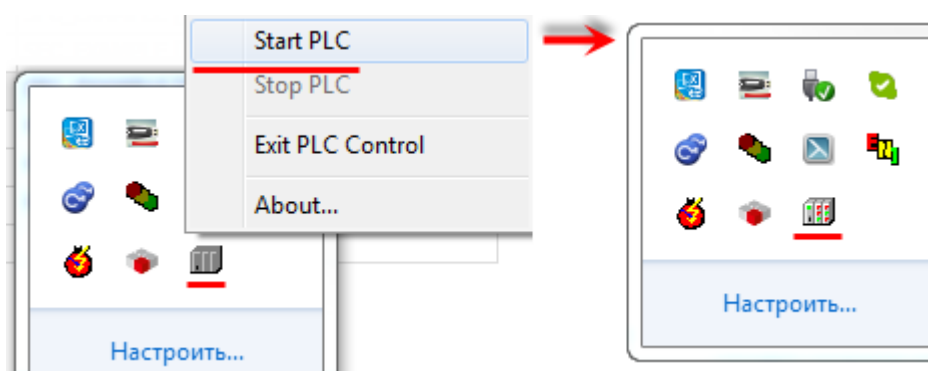


Рис. 9.10. Запуск виртуального контроллера

2. Выбрать в проекте target-файл **CODESYS Control Win V3**:

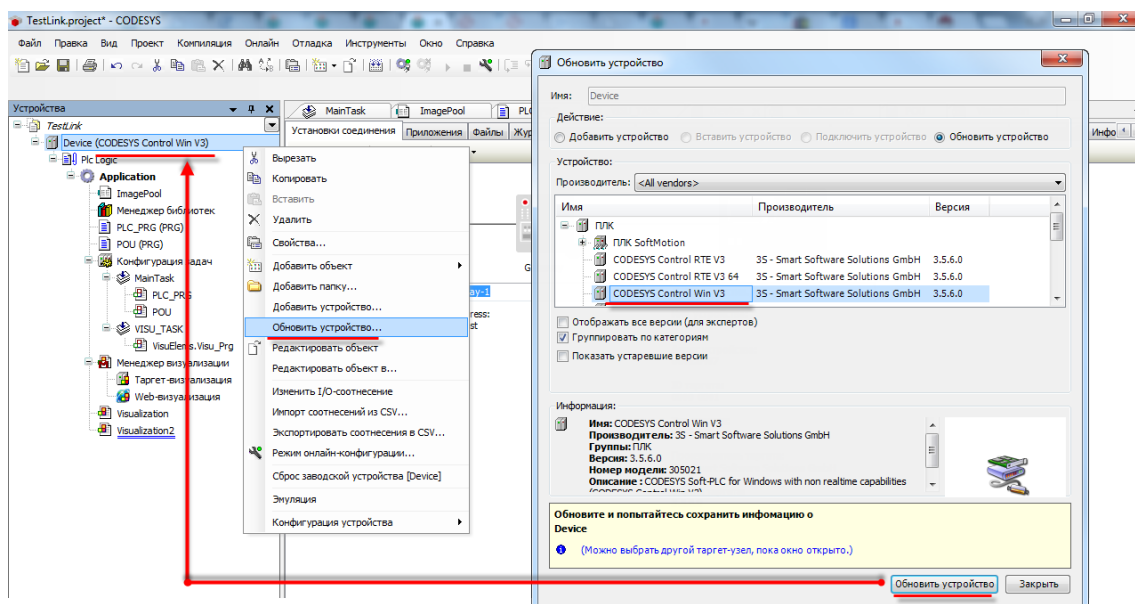


Рис. 9.11. Выбора target-файла виртуального контроллера

3. В установках соединения произвести сканирование сети и выбрать устройство, чье имя совпадает с именем ПК:

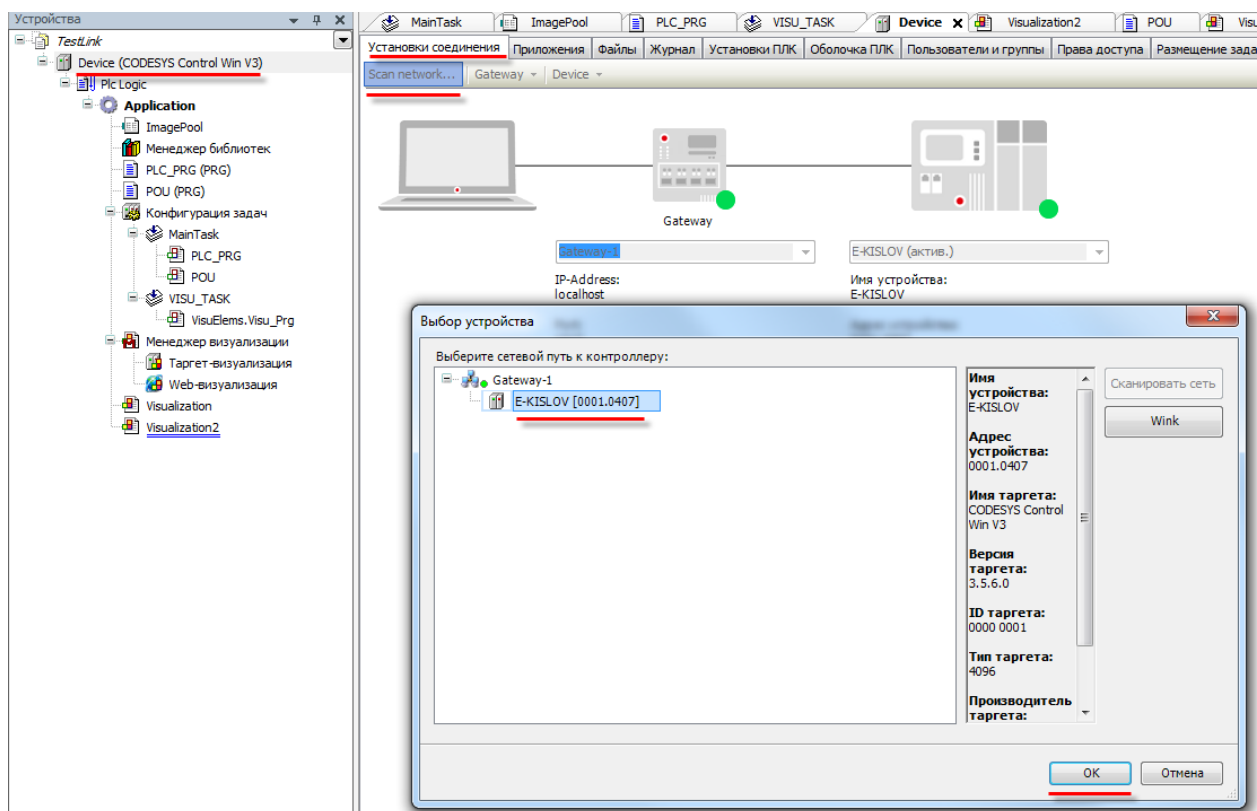


Рис. 9.12. Окно сканирования сети. Подключение к виртуальному контроллеру

Необходимо обратить внимание, что вместе с CODESYS распространяется версия виртуального контроллера с **ограничением времени** непрерывной работы (около часа). После этого контроллер можно перезапустить. При необходимости лицензию на виртуальный контроллер можно приобрести у компании 3S.

9.9. Как отладить программу?

Удобным средством отладки программ является использование **точек останова** – меток в программе, по достижению которых программа останавливается. Обратите внимание, что для их использования необходим запуск приложения на СПК или виртуальном контроллере. В **режиме эмуляции** точки останова не работают.

Для добавления точки останова необходимо выделить нужную строку или функциональный блок программы и нажать **F9** (можно также использовать команды **Переключить точку останова** или **Новая точка останова** из меню **Отладка** или контекстного меню редактора программирования). Обратите внимание, что точки останова можно поставить только на определенных строках; рядом с номерами таких строк расположены серые кружки. После установки точки останова, кружок становится красным:

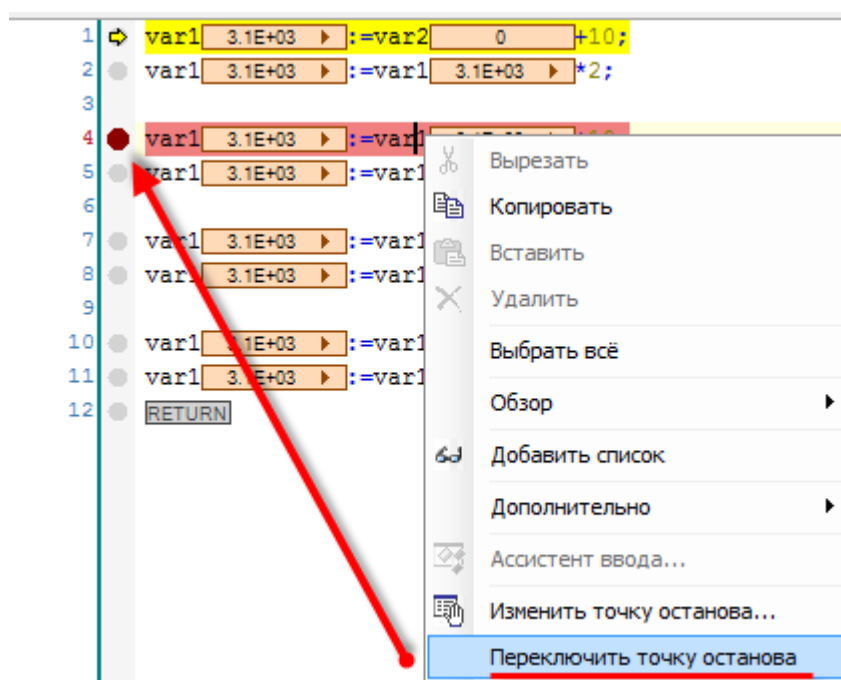


Рис. 9.13. Добавление точки останова

При наличии в проекте точек останова становятся активными команды выполнения команды по шагам из меню **Отладка**:

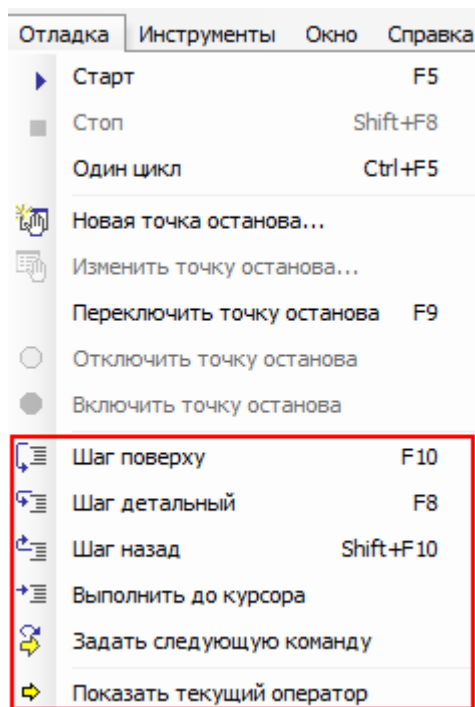


Рис. 9.14. Команды отладки

Шаг поверху – выполняется один шаг (одна инструкция программы). Если конструкция вызывает РОУ, то РОУ выполняется целиком;

Шаг детальный – выполняется один шаг. Вызванный РОУ выполняется по шагам;

Шаг назад – выполняется переход в начало приложения. Если вызван РОУ, то выполняется переход к вызывающей его конструкции;

Выполнить до курсора – выполняет весь программный код до места установки курсора, после чего программа останавливается;

Задать следующую команду – выполняет команду после позиции курсора;

Показать текущий оператор – показывает текущую позицию курсора.

9.10. Как эмулировать входные/выходные сигналы?

Для изменения значений переменных при отладке программы удобно использовать **Списки просмотра**. Чтобы добавить такой список, необходимо в меню **Вид** выбрать вкладку **Просмотр**, а в ней – любой из четырех списков.

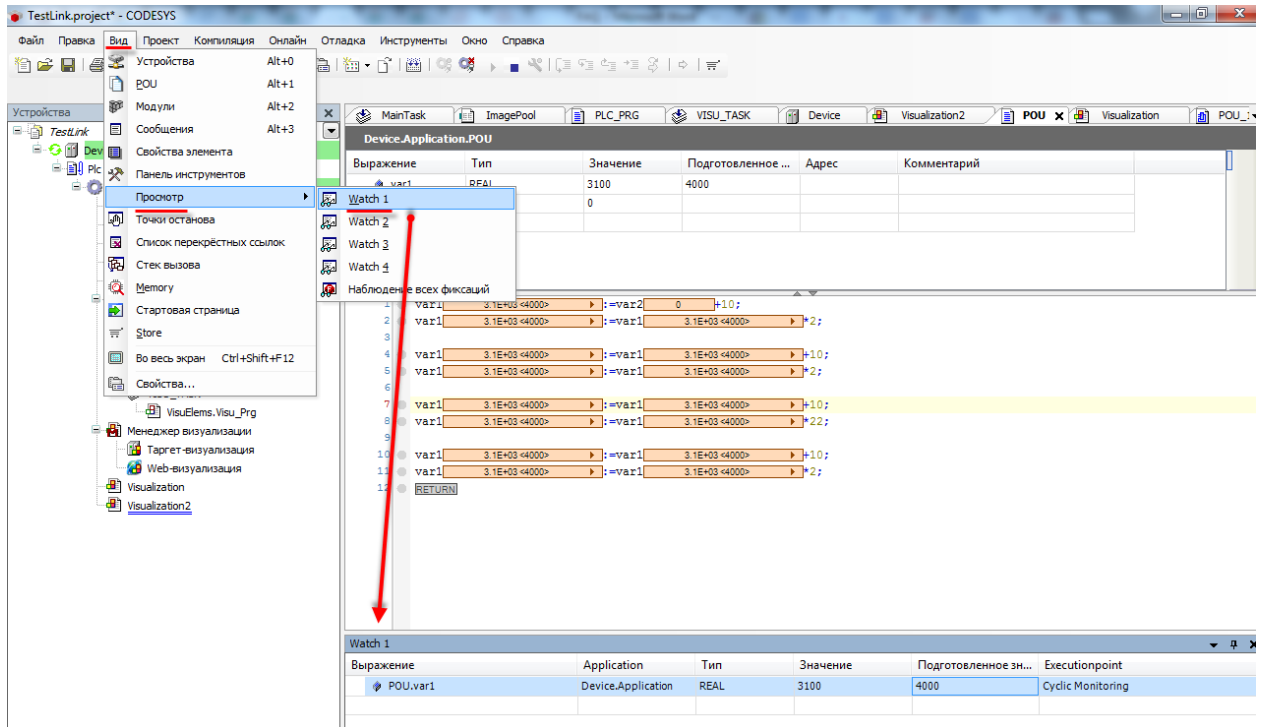


Рис. 9.15. Создание списка просмотра

В столбце **Выражение** выбирается контролируемая **переменная**; значение из столбца **Подготовленное значение** записывается в переменную при выборе соответствующей команды из контекстного меню (открывается по нажатию **ПКМ** на строку):

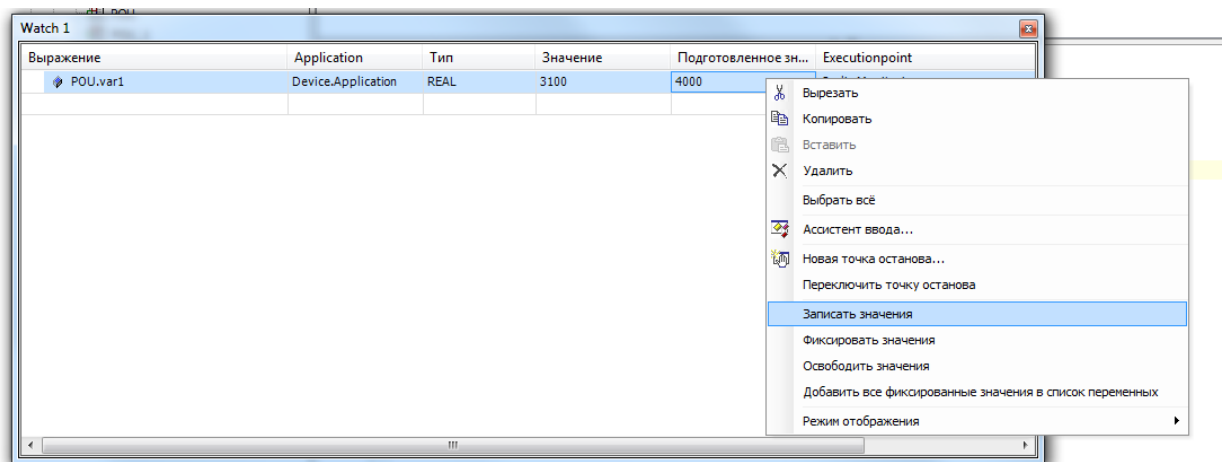


Рис. 9.16. Изменения значения переменной в процессе отладки

9.11. Что делать при появлении ошибки «Приложение в исключении»?



Рис. 9.17. Строка состояния проекта, остановленного с ошибкой «Исключение»

Такой статус в **Строке состояния** может появиться при возникновении в процессе работы программы критической ошибки. Подробная информация об ошибке содержится на вкладке **Журнал** компонента **Device**. Как видно по рис. 9.18, в данном случае ошибка вызвана возникновением в программе деления на ноль.

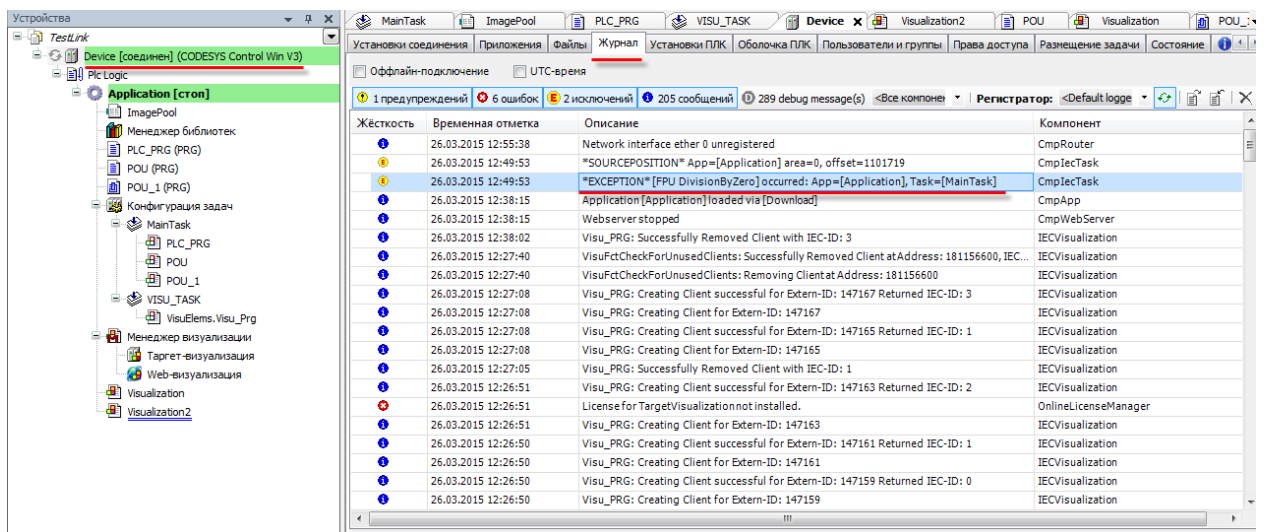


Рис. 9.18. Журнал проекта

9.12. Что делать при появлении ошибки «Открыто слишком много визуализаций»?

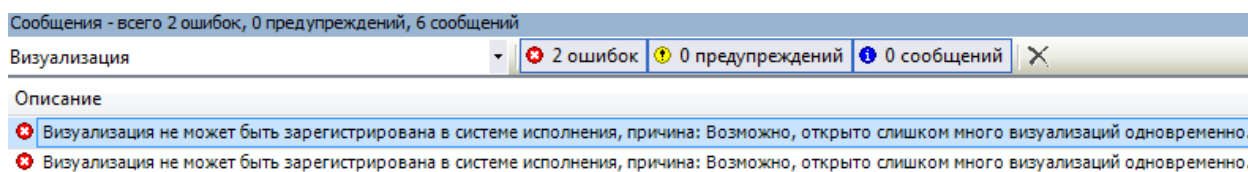


Рис. 9.19. Ошибка «Открыто слишком много визуализаций»

Такая ошибка возникает, если в момент подключения в среде программирования **CODESYS** открыто несколько экранов визуализации:



Рис. 9.20. Вкладки экранов визуализации

Для решения этой проблемы достаточно закрыть все экраны визуализации (один экран, например, стартовый, можно оставить).

9.13. Что делать при возникновении ошибок компиляции?

Если в процессе компиляции возникают ошибки, то проект не будет загружен в контроллер. Ошибки компиляции в целом можно разделить на три группы:

1. Ошибки версий компонентов. Такие ошибки, например, возникают при попытке скомпилировать проект, созданный в более новой версии **CODESYS**. Следует обновить **CODESYS** или добавить в проект необходимые компоненты (например, с помощью установки архива репозитория).

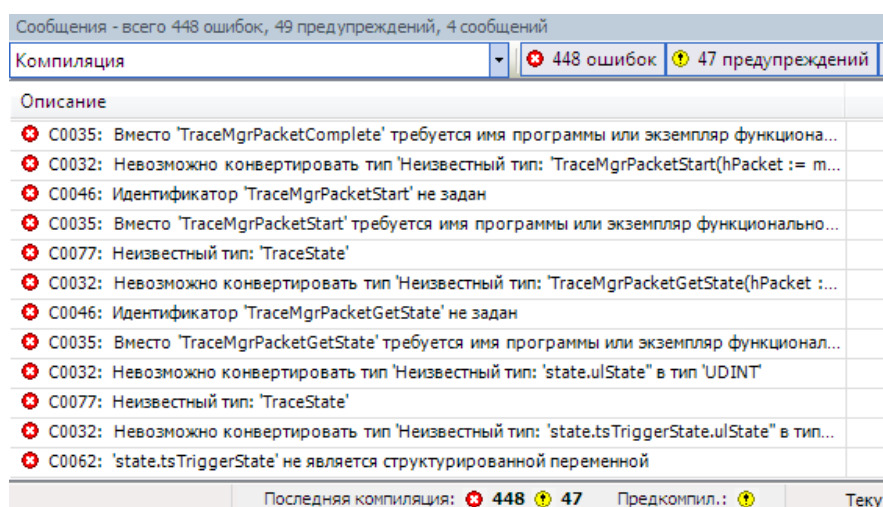


Рис. 9.21. Ошибки, вызванные отсутствием компонентов CODESYS

2. Ошибки программирования. При их возникновении следует исправить соответствующий программный код.

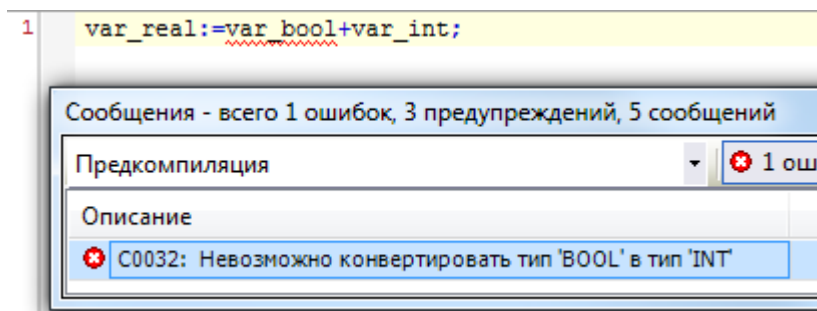


Рис. 9.22. Ошибка программирования

3. Ошибки в результате многочисленных последовательных компиляций. Каждая компиляция приводит к созданию (или перезаписи) файлов компиляции в папке проекта. В определенных случаях при частой перезаписи этих файлов возникают ошибки (см. рис. 9.23). В связи с этим после внесения в проект значительных изменений **крайне рекомендуется** выполнять команды **Очистить все** и **Перекомпиляция** из меню **Компиляция**.

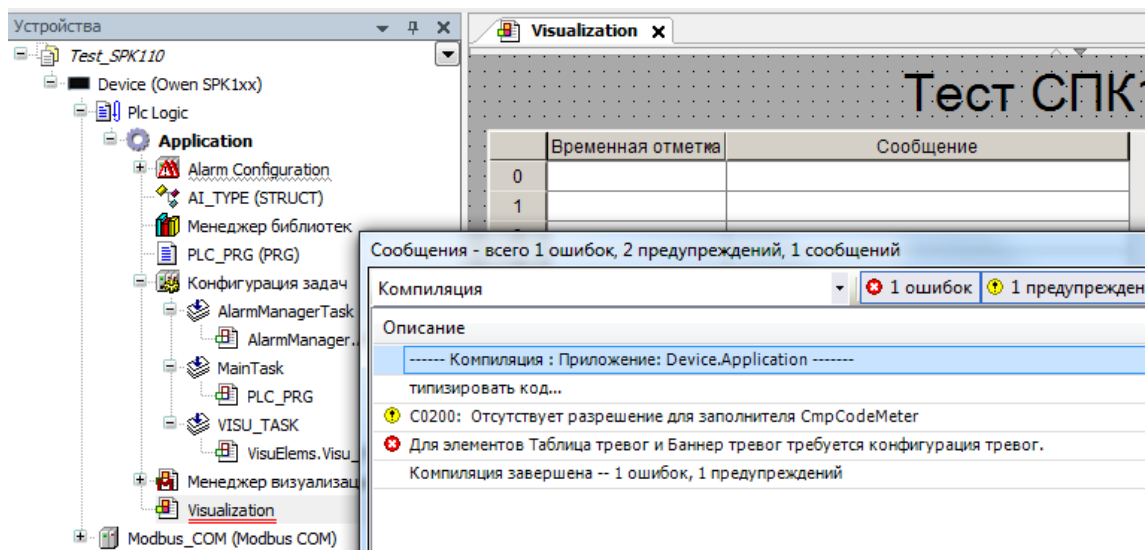


Рис. 9.23. Ошибка файлов компиляции

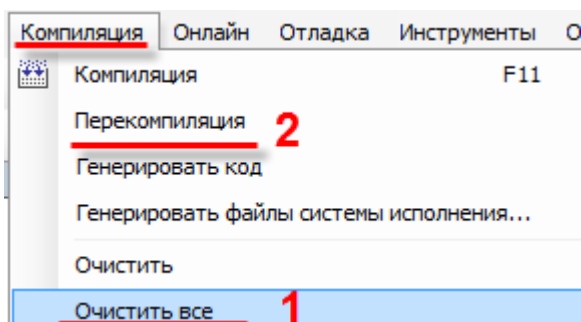


Рис. 9.24. Решение проблемы - очистка файлов компиляции и последующая перекомпиляция

10. Вопросы по памяти СПК

10.1. Для чего используется встроенная память СПК? Какой объем встроенной памяти доступен пользователю?

Встроенная память СПК подразделяется на оперативную память и flash-память. Объем оперативной памяти – 64 Мб для СПК1хх и 128 Мб для СПК2хх. Объем flash-памяти – 128 Мб для любой из моделей СПК, из которых около определенного процента занимает **Linux** и системные библиотеки, а остальное доступно пользователю. Кол-во свободной памяти можно посмотреть в [конфигураторе СПК](#).

10.2. Существует ли ограничение на размер проекта?

Размер записываемого в СПК проекта не должен превышать 7 Мб. При необходимости загрузить проект большего размера, следует обратиться в [техподдержку](#) компании ОВЕН.

10.3. Какие USB- и SD- накопители можно подключать СПК?

Любые, удовлетворяющие следующим требованиям: файловая система – FAT (16 или 32), объем – не более 32 Гб. Рекомендуется использовать накопители промышленного исполнения. При возникновении проблем с подключением накопителя следует обратиться в [техподдержку](#) компании ОВЕН.

10.4. По какому пути монтируются USB- и SD- накопители?

USB A: `/mnt/ufs/media/sda1`

USB B: `/mnt/ufs/media/sdb1`

SD: `/mnt/ufs/media/mmcblk0p1`

10.5. Сколько retain переменных можно использовать в проекте?

Объем энергонезависимой памяти для записи **retain** переменных (сохраняемых при пропадании питания) составляет 4 Кб – это соответствует 1000 переменных типа DWORD.

10.6. Можно ли использовать в проекте persistent переменные?

В данный момент использование persistent переменных не поддерживается.

11. Проблемы настройки обмена с другими устройствами

Один из наиболее часто возникающих у пользователей вопросов – как настроить обмен данными с другими устройствами (датчиками, модулями, контроллерами и т.д.), подключаемыми к СПК. Поскольку настройка обмена – крайне сложный процесс, то в данном документе будут рассмотрены лишь его ключевые моменты. Полную информацию вы можете найти в руководстве **Настройка обмена по протоколу Modbus в среде программирования CODESYS 3.5 для контроллеров фирмы ОВЕН**, доступном на [форуме компании ОВЕН в разделе СПК](#).

11.1. Какие интерфейсы и протоколы используются для подключения к СПК других устройств?

В зависимости от модели и модификации СПК, набор присутствующих интерфейсов для подключения других устройств и поддерживаемых этими интерфейсами протоколов может различаться:

Модель и модификация контроллера	Интерфейсы	Протоколы
СПК1xx	RS-232/RS-485	Modbus (RTU, ASCII), ОВЕН
СПК2xx-х.03.xx-xx-х	RS-232/RS-485	Modbus (RTU, ASCII), ОВЕН
	Ethernet	Modbus TCP
СПК2xx-х.04.xx-xx-х	RS-232/RS-485	Modbus (RTU, ASCII), ОВЕН
	Ethernet	Modbus TCP
	CAN	CAN OPEN (Master)

11.2. Как соотносятся номера COM-портов СПК с номерами портов в CODESYS?

При настройке интерфейсов RS-232/485 в **CODESYS** необходимо указывать номера портов. Номер порта в CODESYS **не соответствует** номеру, указанному на корпусе прибора. Соответствие между номерами портов на корпусе СПК и в CODESYS приведено ниже:

Нумерация портов на корпусе прибора	Нумерация портов в CODESYS			
	СПК105*	СПК107	СПК110	СПК2xx.03
COM1	2 (RS-485) 3 (RS-232)		2	
COM2	-		3	
COM3	-	-		4

* В СПК105 интерфейсы RS-485 и RS-232 выведены на один порт COM1 и, в отличие от остальных СПК, поддерживается их одновременная работа

11.3. Какая распиновка у COM-портов СПК 1xx?

COM-порты СПК1xx отличаются **нестандартной** распиновкой, которая приведена в таблице ниже. COM-порты СПК2xx имеют стандартную распиновку.

Номер контакта 	Наименование сигнала
1	RS-485 A
2	RXD
3	TXD
4	-
5	GND
6	RS-485 B
7	-
8	-
9	-

11.4. Основные особенности настройки обмена по Modbus RTU

11.4.1. Основные особенности настройки обмена по Modbus RTU в режиме Master

Обратите внимание, что приведенная инструкция справедлива для CODESYS V3.5 SP6. В других версиях CODESYS количество возможных настроек для сетевых компонентов может отличаться.

1. Убедитесь, что в проект добавлены все необходимые компоненты:

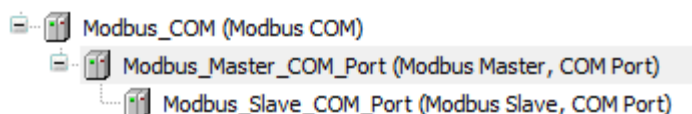


Рис. 11.1. Компоненты master-устройства **Modbus RTU** с опросом одного slave-устройства

2. Версии сетевых компонентов, используемые в проекте, не должны превышать версию **target-файла** контроллера (*рекомендуется* по возможности использовать идентичные версии).
3. Проверьте тип передаваемых переменных – это обязательно должен быть **BOOL** или **WORD**.
4. Проверьте настройки компонента **Modbus COM** (см. рис. 11.1)

4.1. Вкладка **Конфигурация последовательного порта Modbus**

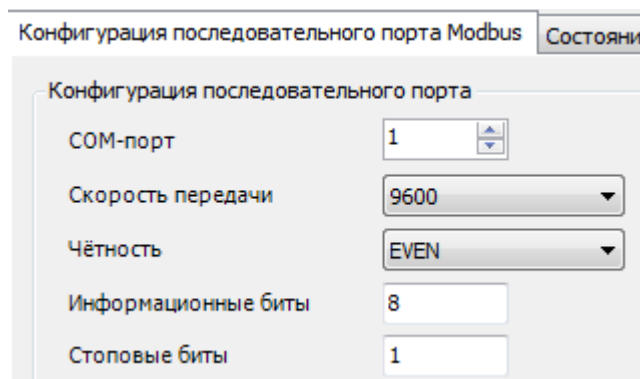


Рис. 11.2. Конфигурация последовательного порта Modbus

COM-порт – номер COM-порта, используемого для передачи данных. Номер порта в CODESYS *не соответствует* номеру порта на задней панели СПК ([см. п. 11.2.](#)).

Скорость передачи – измеряется в бит/с, максимальное значение – 115200.

Четность – режим контроля паритета: EVEN – четный, ODD – нечетный, NONE – отсутствует;

Информационные биты – количество бит передаваемых данных, рекомендуемое значение – 8;

Стоповые биты – *рекомендуется* выставлять значение 1 при наличии контроля паритета и 2 - при его отсутствии.

5. Проверьте настройки компонента **Modbus Master, COM** (см. [рис. 11.1](#))

5.1. Вкладка Конфигурация **Modbus Master**

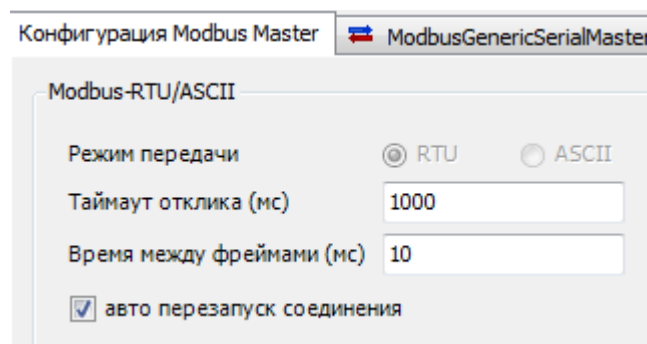


Рис. 11.3. Конфигурация **Modbus Master**

Таймаут отклика – время, которое master дает slave-устройству на ответ. По истечению этого времени, master делает паузу на **время между фреймами** и переходит к опросу следующего slave-устройства. Значение, введенное здесь, будет по умолчанию использоваться для всех slave-устройств. На вкладке **Конфигурация Slave** (см. [рис. 11.5](#)) для каждого устройства можно задать индивидуальный таймаут отклика;

Время между фреймами – время между окончанием ответа slave-устройства и началом опроса следующего. Чем выше скорость, тем меньшим может быть это значение (на скорости 115200 бит/с – 3-5 мс). В то же время определенные устройства (например, СМІ2), в течение 50 мс удерживают линию после ответа, поэтому в данном случае не имеет смысла выставлять время между фреймами меньше, чем это значение.

Автоперезапуск соединения – при **отсутствии** галочки, не ответившее slave-устройство исключается из дальнейшего опроса. **Настоятельно рекомендуется** всегда включать эту опцию.

5.2. Вкладка ModbusGenericSerialMaster Соотнесение входов/выходов

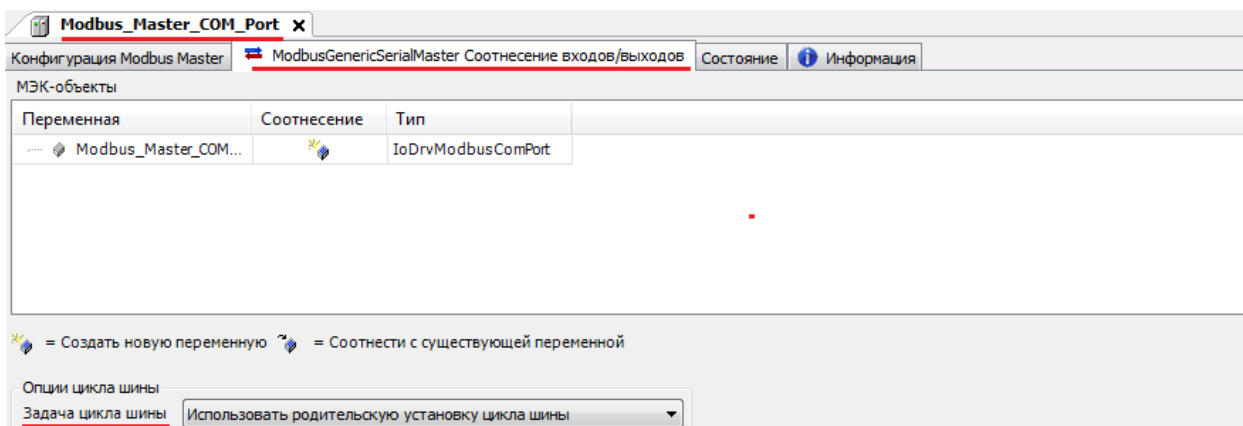


Рис. 11.4. Конфигурация **ModbusGenericSerialMaster** Соотнесение входов/выходов

Задача цикла шины – позволяет выбрать задачу, которая будет определять время цикла вызова **Modbus RTU Master**. При значении **Использовать родительскую установку цикла шины** используется задача, указанная во вкладке **Device – Установки ПЛК – Bus Cycle Task**.

6. Проверьте настройки компонента **Modbus Slave, COM PORT** (см. [рис. 11.1](#))

6.1. Вкладка **Конфигурация Modbus Slave**

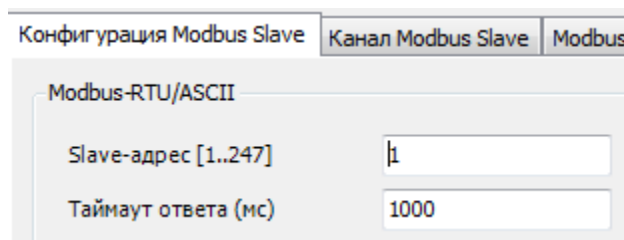


Рис. 11.5. Конфигурация **Modbus Slave**

Slave-адрес – номер slave-устройства в сети Modbus. Может принимать значение от 1 до 247. Должен соответствовать **ID** в конфигурации slave-устройства (см. [рис. 11.12](#));

Таймаут ответа – время, которое master дает slave-устройству на ответ. Спустя это время, master переходит к опросу следующего slave-устройства. Имеет приоритет по сравнению с аналогичным параметром компонента **Modbus Master** (см. [рис. 11.3](#)).

6.2. Вкладка Канал Modbus Slave

Регистр – ячейка памяти, в которой хранится переменная, передаваемая по протоколу Modbus. Существует четыре типа регистров, отличающихся типом хранимых переменных и проводимыми над ними операциями:

1. регистры флагов (coils). Тип переменных – BOOL, возможные операции – чтение и запись.
2. дискретные входы (discrete inputs). Тип переменных – BOOL, возможные операции – только чтение.
3. регистры хранения (holding registers). Тип переменных – WORD, возможные операции – чтение и запись.
4. регистры входа (input registers). Тип переменных – WORD, возможные операции – только чтение.

Канал – неразрывная последовательность регистров (в простейшем случае – один регистр), опрашиваемая определенной функцией с заданным интервалом опроса.

Функция определяет тип операции, производимой с каналом.

Master-устройство может как запрашивать данные у slave-устройства, так и записывать их в него.

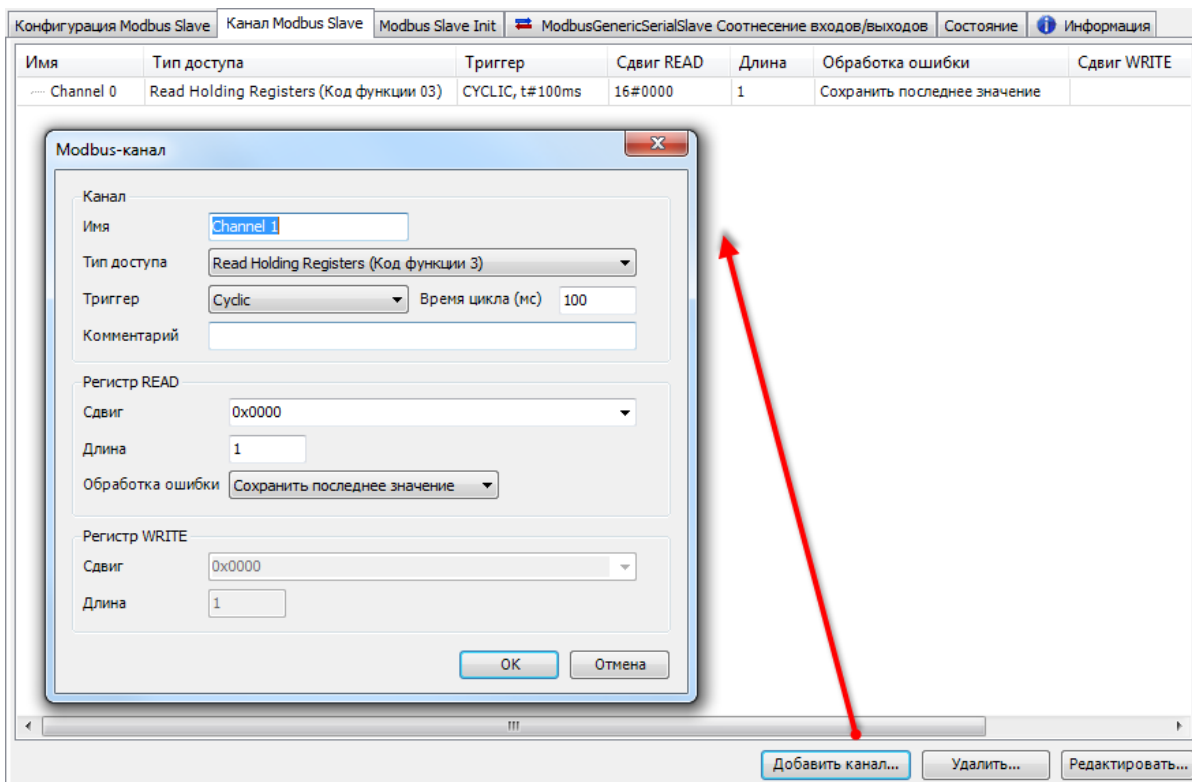


Рис. 11.6. Настройки канала Modbus Slave

Тип доступа – функция Modbus, применяемая к данному каналу. Список доступных функций приведен на рис. 11.7;

Триггер – позволяет настроить тип опроса slave-устройства: циклический (**Cyclic**) или по событию (**Rising edge**);

Время цикла – частота опроса slave-устройства. Должно быть равным или кратным времени цикла приложения (см. [рис. 11.4](#)). Также время цикла должно выбираться в зависимости от опрашиваемого устройства – например, для модулей MB110.8A время обновления данных одного канала для термопары типа ТХК составляет 0.4 секунды, соответственно, разумное время цикла в секундах равно произведению 0.4 на число используемых каналов.

Сдвиг – адрес первого из опрашиваемых регистров в шестнадцатеричном формате;

Длина – количество опрашиваемых регистров;

Обработка ошибки – обработка данных в случае возникновения ошибок соединения: сохранить последние значения или сбросить значения в ноль.

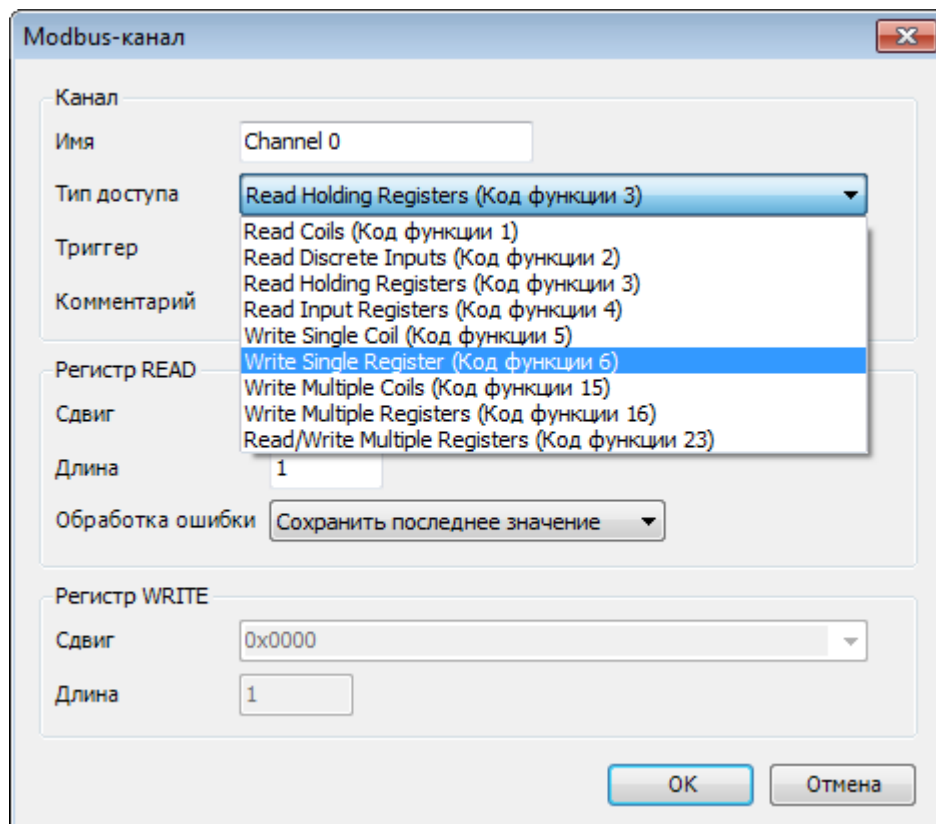


Рис. 11.7. Список функций Modbus, доступных в CODESYS

№ функции (10-ричный формат)	№ функции (16-ричный формат)	Название функции	Тип данных	Описание
1	(0x01)	Read Coils	BOOL	чтение значений из нескольких регистров флагов
2	0x02	Read Discrete Inputs	BOOL	чтение значений из нескольких дискретных входов
3	0x03	Read Holding Registers	WORD	чтение значений из нескольких регистров хранения
4	0x04	Read Input Registers	WORD	чтение значений из нескольких регистров ввода
5	0x05	Write Single Coil	BOOL	запись значения в один регистр флага
6	0x06	Write Single Registers	WORD	запись значения в один регистр хранения
15	0x0F	Write Multiple Coils	BOOL	запись значений в несколько регистров флагов
16	0x10	Write Multiple Registers	WORD	запись значений в несколько регистров хранения
23	0x17	Read/Write Multiple Registers	WORD	чтение/запись значений в несколько регистров хранения

6.3. Вкладка **ModbusGenericSerialSlave** Соотношение входов/выходов

На этой вкладке происходит привязка переменных программы (типа BOOL или WORD) к регистрам Modbus. В столбце **Канал** указывается номер канала, в квадратных скобках – номер регистра. Адресация регистров ведется с нуля. После привязывания переменной к регистру, его адрес зачеркивается.

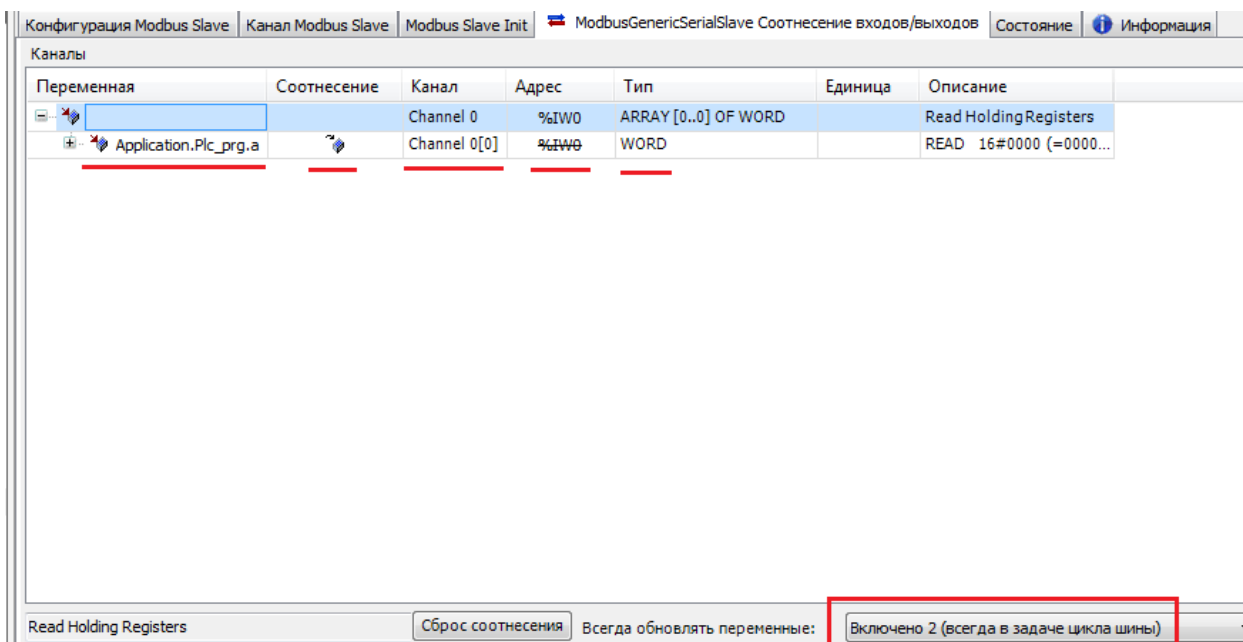


Рис. 11.8. Привязка переменных к регистрам

Крайне важно обратить внимание на то, что во вкладке **Всегда обновлять переменные** для корректного обмена должно быть выставлено значение **Включено 2 (всегда в задаче цикла шины)**.

7. В случае наличия связи между СПК и другим устройством во время запуска программы в CODESYS у сетевых компонентов будут отображаться зеленые пиктограммы:

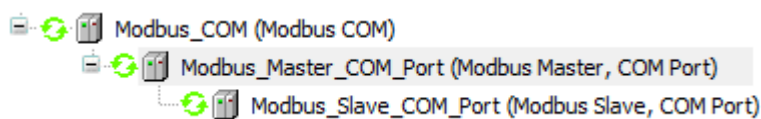


Рис. 11.9. Внешний вид сетевых компонентов master-устройства Modbus COM при наличии связи со slave-устройством

11.4.2. Основные особенности настройки обмена по Modbus RTU в режиме Slave

Обратите внимание, что приведенная инструкция справедлива для CODESYS V3.5 SP6. В других версиях CODESYS количество возможных настроек для сетевых компонентов может отличаться.

1. Убедитесь, что в проект добавлены все необходимые компоненты:

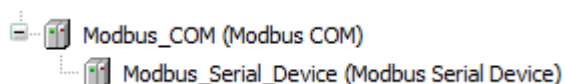


Рис. 11.10. Компоненты slave-устройства **Modbus RTU**

2. Версии сетевых компонентов, используемые в проекте, не должны превышать версию **target-файла** контроллера (*рекомендуется* по возможности использовать идентичные версии).
3. Проверьте тип передаваемых переменных – это обязательно должен быть **word**.
4. Проверьте настройки компонента **Modbus COM** (см. рис. 11.10)

4.1. Вкладка **Конфигурация последовательного порта Modbus**

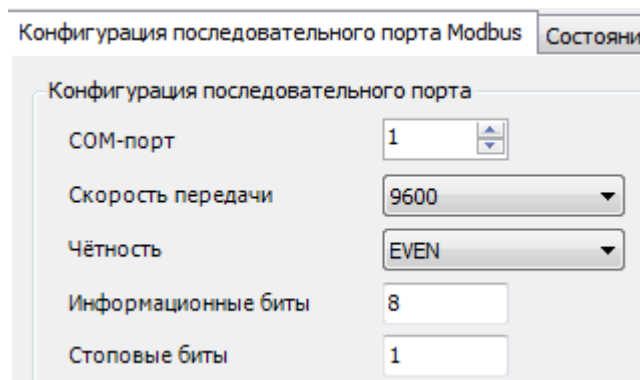


Рис. 11.11. Конфигурация последовательного порта Modbus

COM-порт – номер COM-порта, используемого для передачи данных. Номер порта в CODESYS *не соответствует* номеру порта на задней панели СПК (см. [п. 11.2.](#));

Скорость передачи – измеряется в бит/с, максимальное значение – 115200.

Четность – режим контроля паритета: EVEN – четный, ODD – нечетный, NONE – отсутствует;

Информационные биты – количество бит передаваемых данных, рекомендуемое значение – 8;

Стоповые биты – *рекомендуется* выставлять значение 1 при наличии контроля паритета и 2 - при его отсутствии.

5. Проверьте настройки компонента **Modbus Serial Device** (см. [рис. 11.10](#))

5.1. Вкладка **Modbus Serial Device**

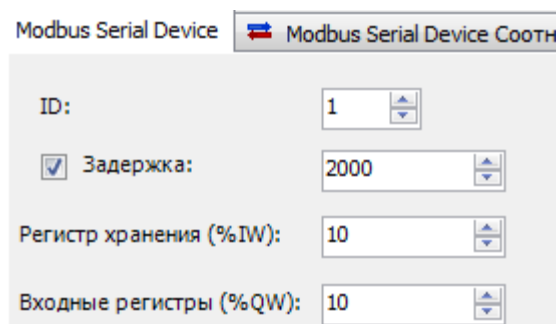


Рис. 11.12. Конфигурация **Modbus Serial Device**

Unit ID – номер slave-устройства в протоколе Modbus. Может принимать значение от 1 до 247. Должен соответствовать slave-адресу в конфигурации master-устройства (см. [рис. 11.3](#));

Задержка – время ожидания запроса от master-устройства. Если за это время запроса не происходит, то данные обнуляются. *При отсутствии галочки* обнуления данных *не происходит*;

Регистр хранения – определяет количество **holding** регистров канала (см. [рис. 11.13](#)). Параметр может принимать значения от 2 до 40;

Входные регистры – определяют количество **input** регистров канала (см. [рис. 11.13](#)). Параметр может принимать значения от 2 до 40.

5.2. Вкладка Modbus Serial Device Соотнесение входов/выходов

На этой вкладке происходит привязка переменных (типа BOOL или WORD) slave-устройства к переменным Modbus. В столбце **Канал** указывается номер канала, в квадратных скобках – номер регистра. Адресация ведется с нулевого регистра. Адрес привязанного регистра зачеркивается.

Для записи данных в **slave-устройство**, master-устройство использует **регистры хранения** (регистры канала **Inputs**). Переменные, привязанные к этим регистрам, могут считываться и записываться **master-устройством** с помощью функций **3, 6, 16** (см. [рис. 11.7](#)) При этом значения таких переменных **не могут изменяться** самим slave-устройством.

Для чтения данных из **slave-устройства**, master-устройство использует **регистры входа** (регистры канала **Outputs**). Переменные, привязанные к этим регистрам, могут считываться master-устройством с помощью функции **4** и **не могут** им **записываться**. При этом значения таких переменных **могут изменяться** самим **slave-устройством** (если в нем запущена соответствующая программа).

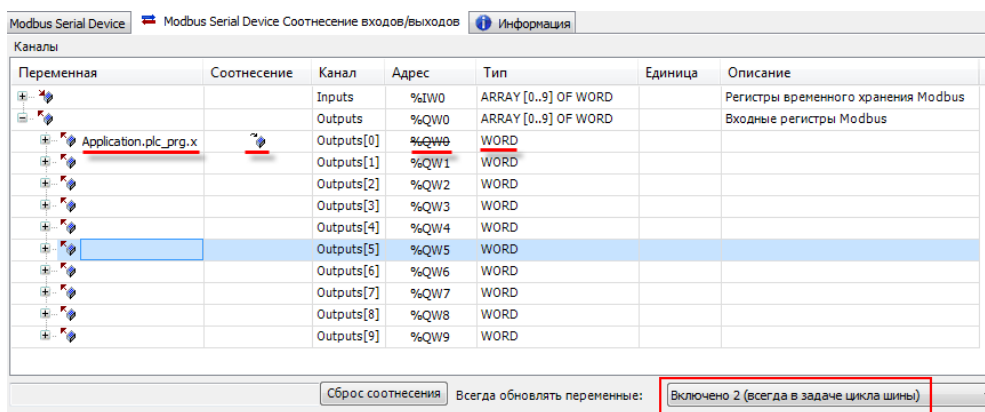


Рис. 11.13. Привязка переменных к регистрам

Крайне важно обратить внимание на то, что во вкладке **Всегда обновлять переменные** для корректного обмена должно быть выставлено значение **Включено 2 (всегда в задаче цикла шины)**.

6. В случае наличия связи между СПК и другим устройством во время запуска программы в **CODESYS** у сетевых компонентов будут отображаться зеленые пиктограммы:

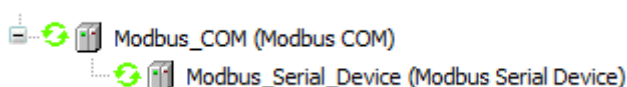


Рис. 11.14. Внешний вид сетевых компонентов slave-устройства Modbus COM при наличии связи со master-устройством

11.5. Основные особенности настройки обмена по Modbus ASCII

В настоящий момент протокол **Modbus ASCII** не поддерживается средой программирования **CODESYS**, но может быть добавлен с помощью библиотеки **Modbus V3**, доступной на сайте компании OВЕН в разделе **CODESYS V3/Библиотеки CODESYS**.

11.6. Основные особенности настройки обмена по Modbus TCP

11.6.1. Основные особенности настройки обмена по Modbus TCP в режиме Master

Обратите внимание, что приведенная инструкция справедлива для CODESYS V3.5 SP6. В других версиях CODESYS количество возможных настроек для сетевых компонентов может отличаться.

1. Убедитесь, что устройства находятся в одной в сети и между ними есть пинг – для этого необходимо подключиться к контроллеру с помощью утилиты **Putty** (см. [п. 14.7](#)) и в терминале выполнить команду **ping <IP-адрес устройства>**. Остановить проверку пинга можно сочетанием клавиш **Ctrl+C**.

2. Убедитесь, что в проект добавлены все необходимые компоненты:

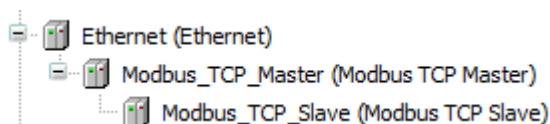


Рис. 11.15. Компоненты master-устройства **Modbus TCP** с опросом одного slave-устройства

3. Версии сетевых компонентов, используемые в проекте, не должны превышать версию **target-файла** контроллера (**рекомендуется** по возможности использовать идентичные версии).

4. Проверьте тип передаваемых переменных – это обязательно должен быть **BOOL** или **WORD**.

5. Проверьте настройку компонента **Modbus TCP Master** (см. [рис. 11.15](#))

5.1. Вкладка Конфигурация **Modbus_TCP_Master**

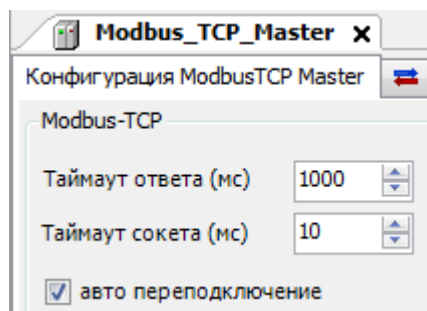


Рис. 11.16. Конфигурация **ModbusTCP Master**

Таймаут ответа – время, которое master дает slave-устройству на ответ. По истечению этого времени, master переходит к опросу следующего slave-устройства. Значение, введенное здесь, будет по умолчанию использоваться для всех slave-устройств. На вкладке **Конфигурация Slave** (см. [рис. 11.18](#)) для каждого устройства можно задать индивидуальное время ожидания;

Таймаут сокета – максимальное время ожидания master'ом входящих пакетов **TCP/IP**. *Не рекомендуется* изменять значение по умолчанию;

Авто переподключение – при **отсутствии** галочки, не ответившее slave-устройство исключается из дальнейшего опроса. **Настоятельно рекомендуется** всегда включать эту опцию.

5.2. Вкладка ModbusTCPMaster Соотнесение входов/выходов

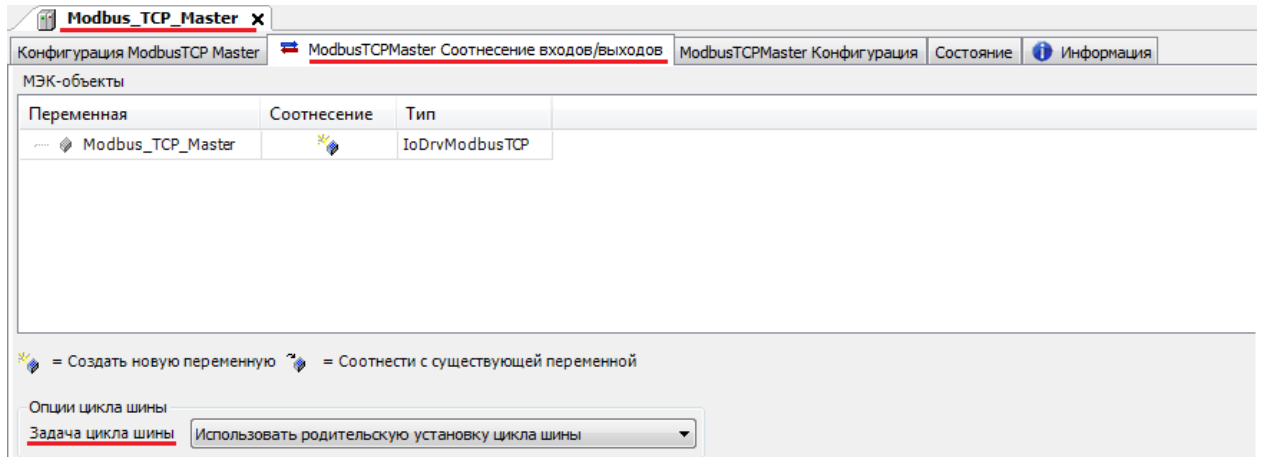


Рис. 11.17. Конфигурация ModbusTCPMaster Соотнесение входов/выходов

Задача цикла шины – позволяет выбрать задачу, которая будет определять время цикла вызова Modbus TCP Master. При значении **Использовать родительскую установку цикла шины** используется задача, указанная во вкладке **Device – Установки ПЛК – Bus Cycle Task**.

6. Настройки компонента ModbusTCP Slave (см. [рис. 11.15](#))

6.1. Вкладка ModbusTCP Slave

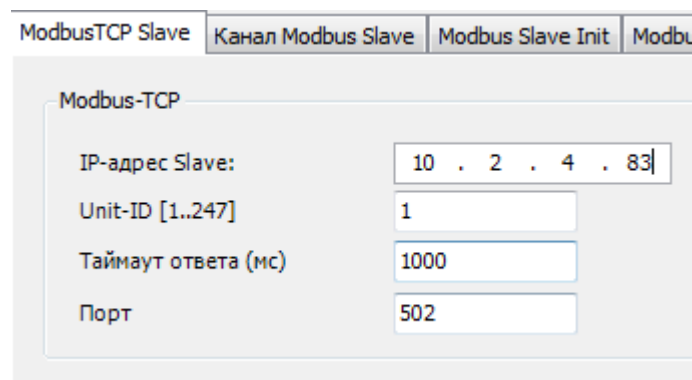


Рис. 11.18. Конфигурация ModbusTCP Slave

IP-адрес Slave – проверьте соответствие выставленного здесь адреса с IP-адресом slave-устройства;

Unit ID – номер slave-устройства в сети Modbus. Может принимать значение от 1 до 247. Должен соответствовать **ID** в конфигурации slave-устройства (см. [рис. 11.24](#));

Таймаут ответа – время, которое master дает slave-устройству на ответ. Спустя это время, master переходит к опросу следующего slave-устройства. Имеет приоритет по сравнению с аналогичным параметром компонента **Modbus TCP Master** (см. [рис 11.16](#));

Порт – ethernet порт, используемый для обмена данными. **Не рекомендуется** изменять значение по умолчанию.

6.2. Вкладка **Канал Modbus Slave** (см. [рис. 11.15](#))

Регистр – ячейка памяти, в которой хранится переменная, передаваемая по протоколу Modbus. Существует четыре типа регистров, отличающихся типом хранимым переменных и проводимыми над ними операциями:

1. регистры флагов (coils). Тип переменных – BOOL, возможные операции – чтение и запись.
2. дискретные входы (discrete inputs). Тип переменных – BOOL, возможные операции – только чтение.
3. регистры хранения (holding registers). Тип переменных – WORD, возможные операции – чтение и запись.
4. регистры входа (input registers). Тип переменных – WORD, возможные операции – только чтение.

Канал – неразрывная последовательность регистров (в простейшем случае – один регистр), опрашиваемая определенной функцией с заданным интервалом опроса.

Функция определяет тип операции, производимой с каналом.

Master-устройство может как запрашивать данные у slave-устройства, так и записывать их в него.

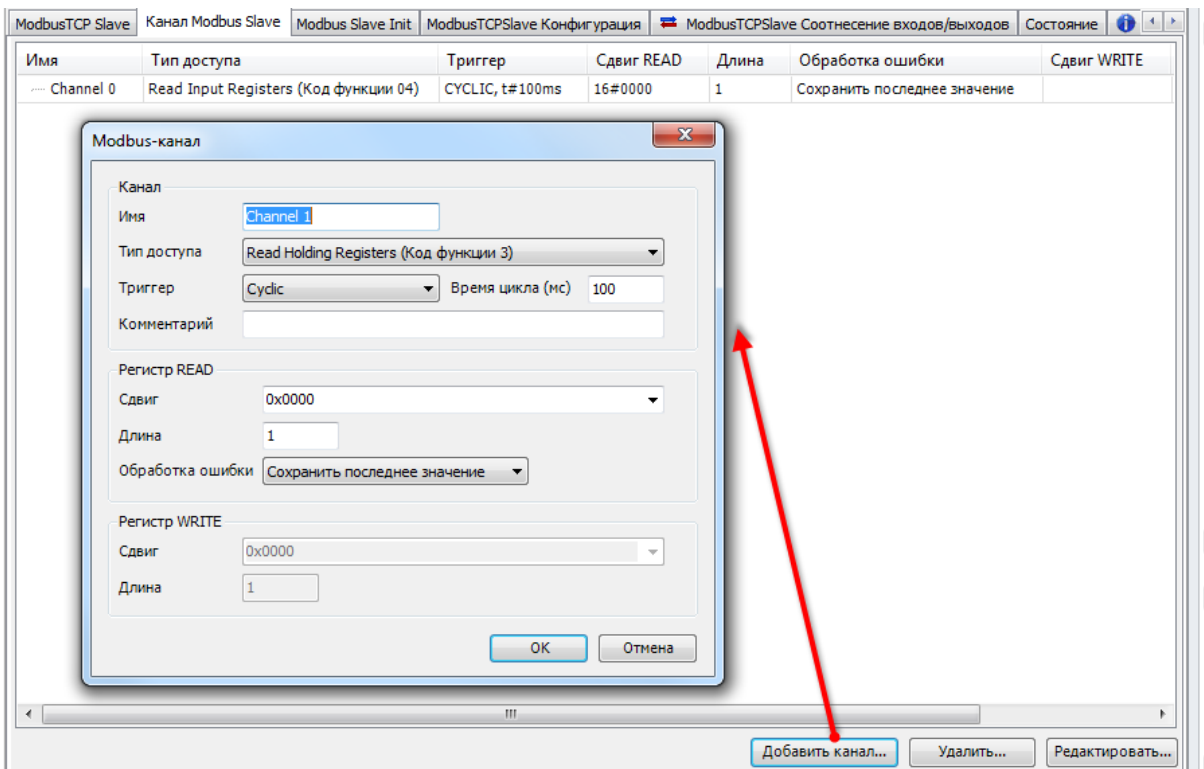


Рис. 11.19. Настройки канала **Modbus Slave**

Тип доступа – функция Modbus, применяемая к данному каналу. Список доступных функций приведен на рис. 11.20;

Триггер – позволяет настроить тип опроса slave-устройства: циклический (**Cyclic**) или по событию (**Rising edge**);

Время цикла – частота опроса slave-устройства. Должно быть равным или кратным времени цикла приложения (см. [рис. 11.17](#)). Также время цикла должно выбираться в зависимости от опрашиваемого устройства – например, для модулей MB110.8A время обновления данных одного канала для термодпары типа ТХК составляет 0.4 секунды, соответственно, разумное время цикла в секундах равно произведению 0.4 на число используемых каналов.

Сдвиг – адрес первого из опрашиваемых регистров в шестнадцатеричном формате;

Длина – количество опрашиваемых регистров;

Обработка ошибки – обработка данных в случае возникновения ошибок соединения: сохранить последние значения или сбросить значения в ноль.

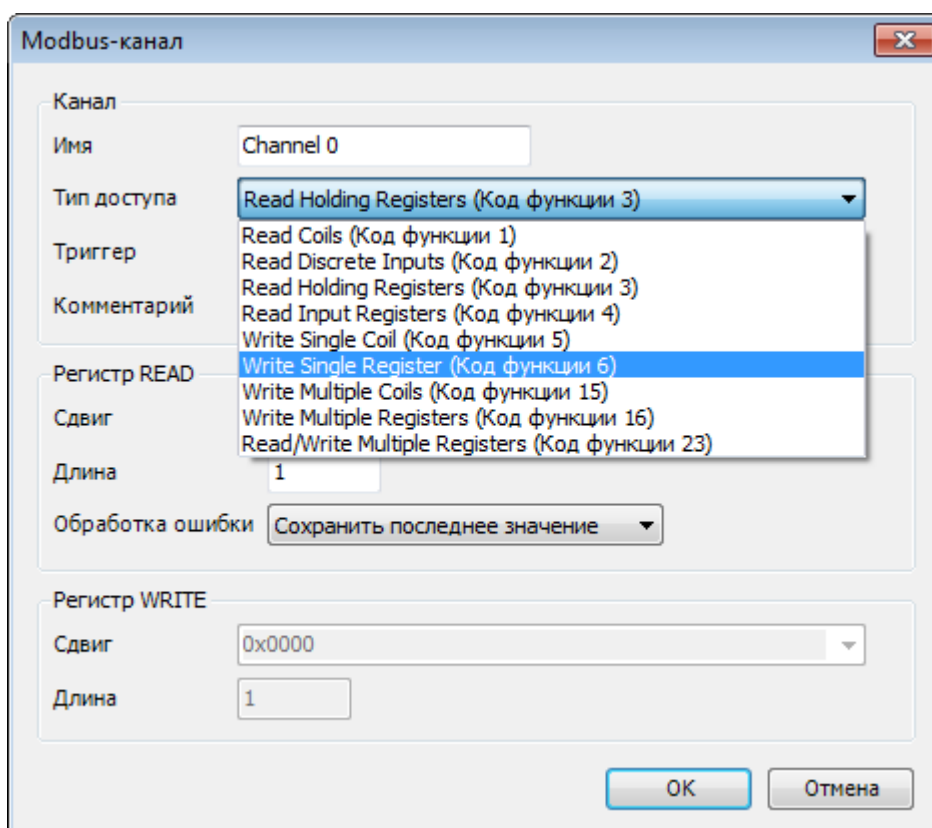


Рис. 11.20. Список функций Modbus, доступных в CODESYS

№ функции (10-ричный формат)	№ функции (16-ричный формат)	Название функции	Тип данных	Описание
1	(0x01)	Read Coils	BOOL	чтение значений из нескольких регистров флагов
2	0x02	Read Discrete Inputs	BOOL	чтение значений из нескольких дискретных входов
3	0x03	Read Holding Registers	WORD	чтение значений из нескольких регистров хранения
4	0x04	Read Input Registers	WORD	чтение значений из нескольких регистров ввода
5	0x05	Write Single Coil	BOOL	запись значения в один регистр флага
6	0x06	Write Single Registers	WORD	запись значения в один регистр хранения
15	0x0F	Write Multiple Coils	BOOL	запись значений в несколько регистров флагов
16	0x10	Write Multiple Registers	WORD	запись значений в несколько регистров хранения
23	0x17	Read/Write Multiple Registers	WORD	чтение/запись значений в несколько регистров хранения

6.3. Вкладка ModbusTCP Slave Соотношение входов/выходов

На этой вкладке происходит привязка переменных программы (типа BOOL или WORD) к переменным Modbus. В столбце канал указывается номер канала, в квадратных скобках – номер регистра. После привязывания переменной к регистру, его адрес зачеркивается.

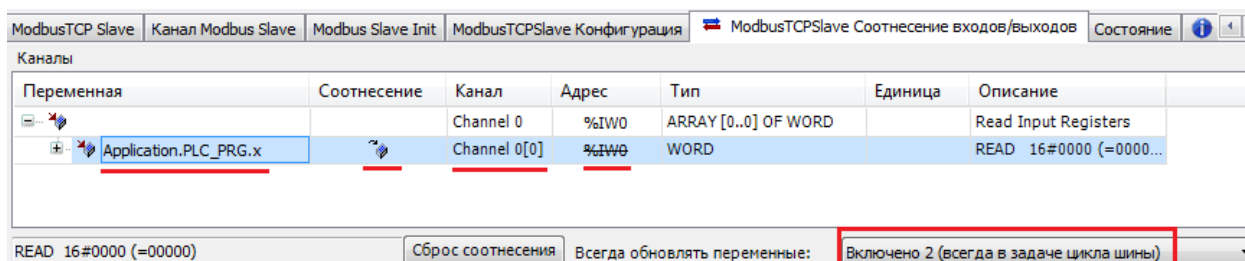


Рис. 11.21. Привязка переменных к регистрам

Крайне важно обратить внимание на то, что во вкладке **Всегда обновлять переменные** для корректного обмена должно быть выставлено значение **Включено 2 (всегда в задаче цикла шины)**.

7. В случае наличия связи между СПК и другим устройством во время запуска программы в CODESYS у сетевых компонентов будут отображаться зеленые пиктограммы:

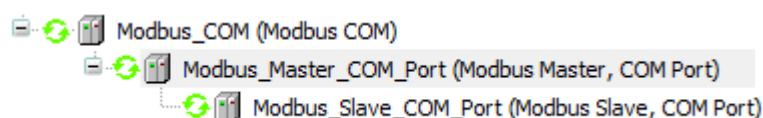


Рис. 11.22. Внешний вид сетевых компонентов master-устройства Modbus COM при наличии связи со slave-устройством

11.6.2. Основные особенности настройки обмена по Modbus TCP в режиме Slave

Обратите внимание, что приведенная инструкция справедлива для CODESYS V3.5 SP6. В других версиях CODESYS количество возможных настроек для сетевых компонентов может отличаться.

1. Убедитесь, что устройства находятся в одной в сети и между ними есть пинг – для этого необходимо подключится к контроллеру с помощью утилиты **Putty** (см. [п. 14.7](#)) и в терминале выполнить команду **ping <IP-адрес устройства>**. Остановить проверку пинга можно сочетанием клавиш **Ctrl+C**.

2. Убедитесь, что в проект добавлены все необходимые компоненты:

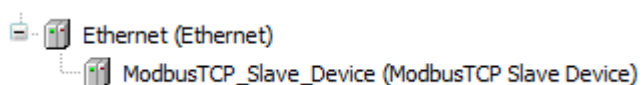


Рис. 11.23. Компоненты slave-устройства **Modbus TCP**

3. Версии сетевых компонентов, используемые в проекте, не должны превышать версию **target-файла** контроллера (*рекомендуется* по возможности использовать идентичные версии).

4. Проверьте тип передаваемых переменных – это обязательно должен быть **BOOL** или **WORD**.

5. Проверьте настройки компонента **ModbusTCP_Slave_Device** (см. [рис. 11.23](#))

5.1. Вкладка **Страница Конфигурации**

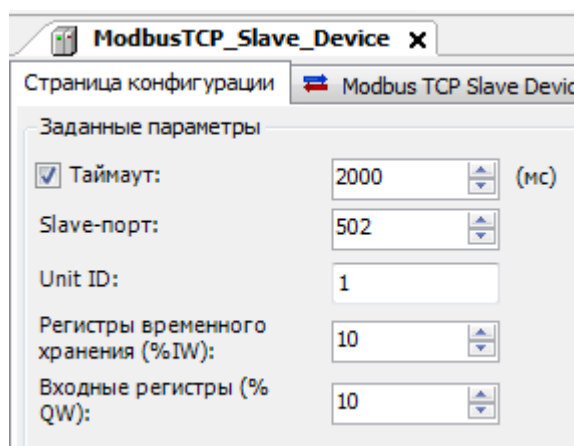


Рис. 11.24. Конфигурация **ModbusTCP_Slave_Device**

Таймаут – время ожидания запроса от master-устройства. Если за это время запроса не происходит, то данные обнуляются. При отсутствии галочки обнуления данных не происходит;

Slave-порт – номер порта ethernet, через который осуществляется передача данных. Должен соответствовать аналогичному параметру master-устройства (см. [рис. 11.18](#)). *Не рекомендуется* изменять значение по умолчанию;

Unit ID – номер slave-устройства в протоколе Modbus. Может принимать значение от 1 до 247. Должен соответствовать **ID** в конфигурации master-устройства (см. [рис. 11.18](#));

Регистры временного хранения – определяют количество **holding** регистров канала. Параметр может принимать значения от 2 до 40 (см. [рис. 11.25](#));

Входные регистры – определяют количество input **регистров** канала. Параметр может принимать значения от 2 до 40 (см. [рис. 11.25](#)).

5.2. Вкладка Modbus TCP Slave Device Соотнесение входов/выходов

На этой вкладке происходит привязка переменных (типа BOOL или WORD) slave-устройства к переменным Modbus. В столбце **Канал** указывается номер канала, в квадратных скобках – номер регистра. Адресация ведется с нулевого регистра. Адрес привязанного регистра зачеркивается.

Для записи данных в **slave-устройство**, master-устройство использует **регистры хранения** (регистры канала **Inputs**). Переменные, привязанные к этим регистрам, могут считываться и записываться **master-устройством** с помощью функций **3, 6, 16** (см. [рис. 11.20](#)) При этом значения таких переменных **не могут изменяться** самим slave-устройством.

Для чтения данных из **slave-устройства**, master-устройство использует **регистры входа** (регистры канала **Outputs**). Переменные, привязанные к этим регистрам, могут считываться master-устройством с помощью функции **4** и **не могут** им **записываться**. При этом значения таких переменных **могут изменяться** самим **slave-устройством** (если в нем запущена соответствующая программа).

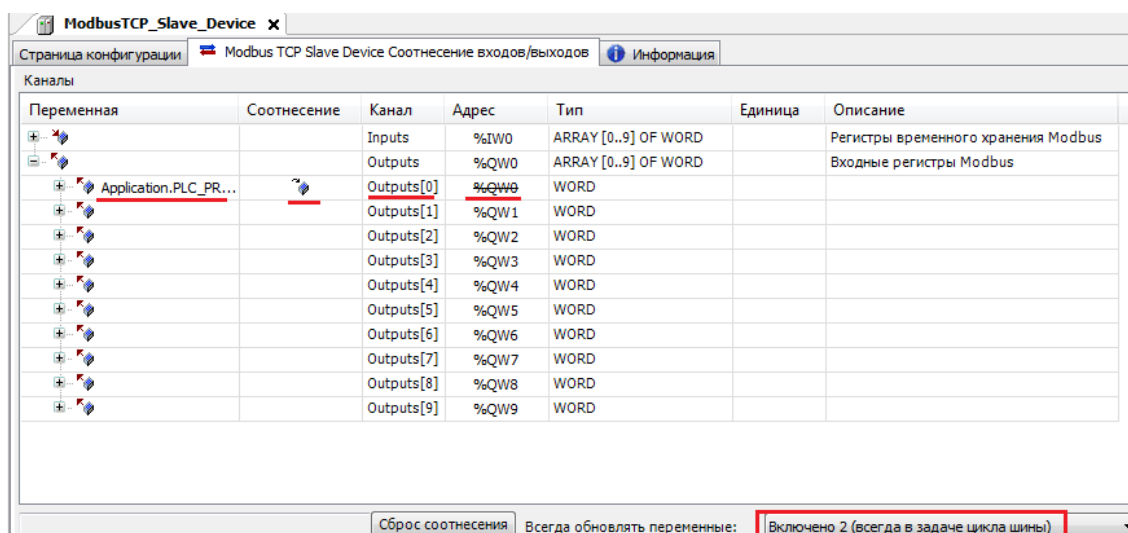


Рис. 11.25. Привязка переменных к регистрам

Крайне важно обратить внимание на то, что во вкладке **Всегда обновлять переменные** для корректного обмена должно быть выставлено значение **Включено 2 (всегда в задаче цикла шины)**.

7. В случае наличия связи между СПК и другим устройством во время запуска программы в **CODESYS** у сетевых компонентов будут отображаться зеленые пиктограммы:

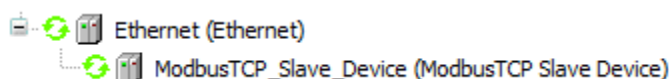


Рис. 11.26. Внешний вид сетевых компонентов slave-устройства Modbus TCP при наличии связи со master-устройством

11.7. Протокол OBD

Настройка обмена данными по протоколу OBD реализуется с помощью библиотеки OBD, доступной на [сайте компании OBD](#) в разделе **CODESYS V3/Библиотеки CODESYS**. Подробное описание протокола можно найти на сайте OBD в разделе **Поддержка/Документация/Сетевые протоколы обмена по RS-485**.

11.8. Протокол CAN OPEN

В **СПК207.04** поддержан стандартный протокол **CAN OPEN**. Для настройки в **CODESYS** обмена данными СПК с каким-либо устройством по данному протоколу, необходим файл описания этого устройства (**EDS файл**).

12. Вопросы по удаленному доступу: WEB-визуализация, модем ПМ01

12.1. Почему не работает WEB-визуализация?

Для просмотра WEB-визуализации необходимо в браузере открыть ссылку:

<IP-адрес контроллера>:8080/<имя web-страницы>.htm

По умолчанию имя web-страницы визуализации – **webvisu**.

Если WEB-визуализация не открывается, необходимо проверить:

1. наличие компонента **WEB-визуализация** в проекте **CODESYS**. Он создается по умолчанию при добавлении компонента **Менеджер визуализации**, но может быть удален пользователем;
2. сетевую видимость СПК и устройства, на котором открывается браузер – они должны находиться в одной локальной сети либо между ними должна быть настроена маршрутизация;
3. поддержку браузером **HTML5** – это необходимо для отображения WEB-визуализации;

Следует заметить, что WEB-визуализация [виртуального контроллера](#) (CODESYS Control Win V3), запускаемого на ПК, имеет ограничение на время отображения (30 минут). Перезапуск виртуального контроллера сбрасывает счетчик.

12.2. Как посмотреть WEB-визуализацию из интернета?

Для этого достаточно настроить маршрутизацию для IP-адреса контроллера – никаких дополнительных действий с СПК совершать не требуется.

12.3. Как настроить обмен данными через модем ПМ01?

Настройка обмена данными через СПК с помощью модема **ПМ01** реализуется с использованием библиотеки **OwenModemLib**. Библиотека и инструкция по ее использованию доступны на [форуме ОВЕН в разделе СПК](#).

12.4. Как с помощью модема ПМ01 отправлять sms?

Передача sms реализуется с помощью библиотеки **OwenModemLib**. Библиотека и инструкция по ее использованию доступны на [форуме ОВЕН в разделе СПК](#).

13. Вопросы по системному времени

13.1. Как отобразить системное время в визуализации?

Проще всего это сделать с помощью задания элементу визуализации (например, прямоугольнику) соответствующего формата вывода: `%t[<заполнители>]`.

Список заполнителей приведен в справке **CODESYS** и документе «СПК. Первый старт» (табл. 5), доступном на [форуме ОВЕН в разделе СПК](#).

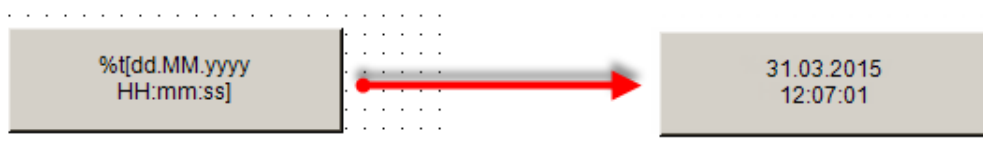


Рис. 10.1. Вывод системного времени с помощью системной переменной

13.2. Как изменить системное время контроллера?

Изменить системное время можно в [конфигураторе СПК](#).

13.3. Как считать системное время из кода программы?

Проще всего это сделать с помощью функционального блока **GetDateAndTime** и функции **DTSplit** из системной библиотеки **CAA_DTUtil**, расположенной в группе библиотек **Time and Date**.

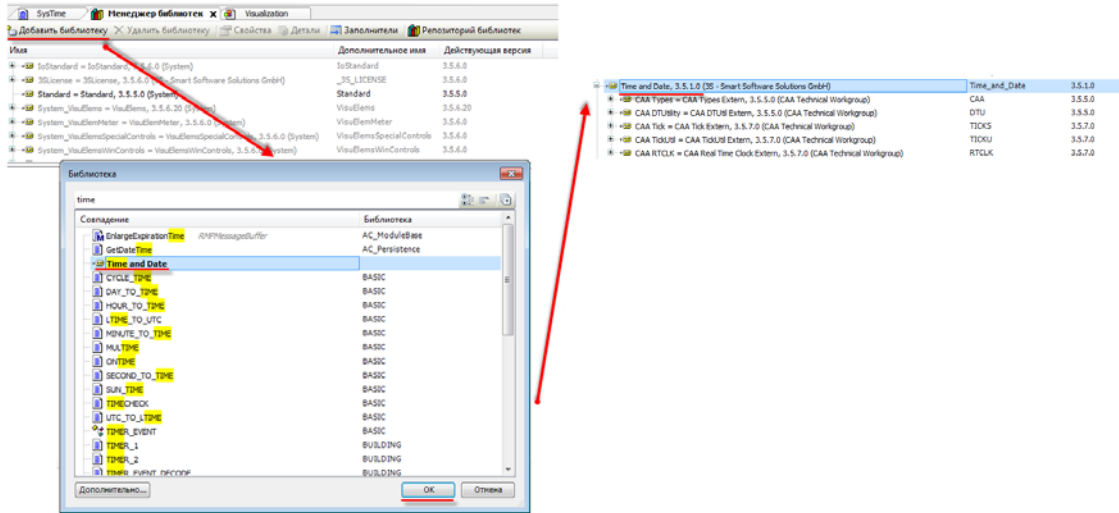


Рис. 10.2. Добавление в проект группы библиотек **Time and Date**

Пример программы считывания времени приведен ниже:

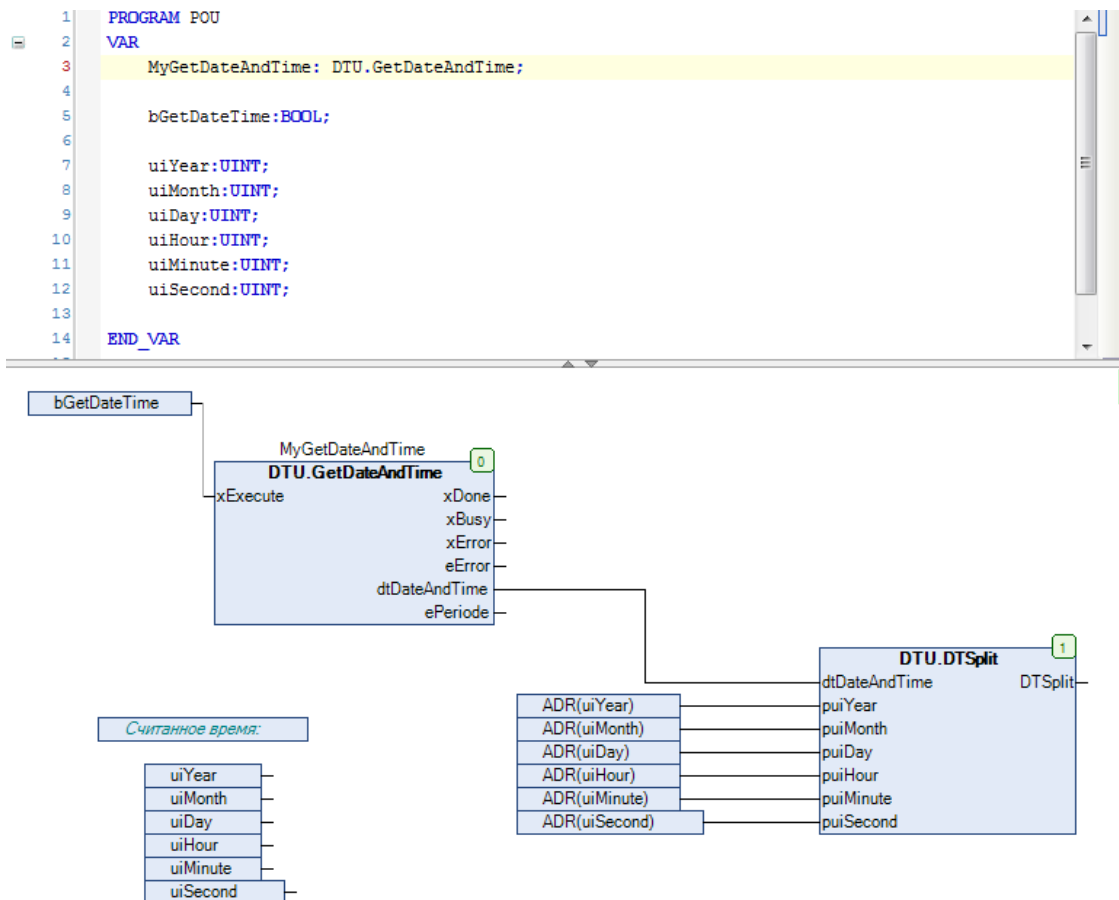


Рис. 10.3. Программа считывания системного времени в редакторе **CFC**

Когда переменная **bGetDateTime** принимает значение **TRUE**, на выходе экземпляра **MyGetDateAndTime** функционального блока **GetDateAndTime** формируется системное время в формате **DT (DATE_AND_TIME)**, которое с помощью функции **DTSplit** разбивается на разряды, каждый из которых записывается в переменную типа **UINT** с помощью указателя на ее адрес.

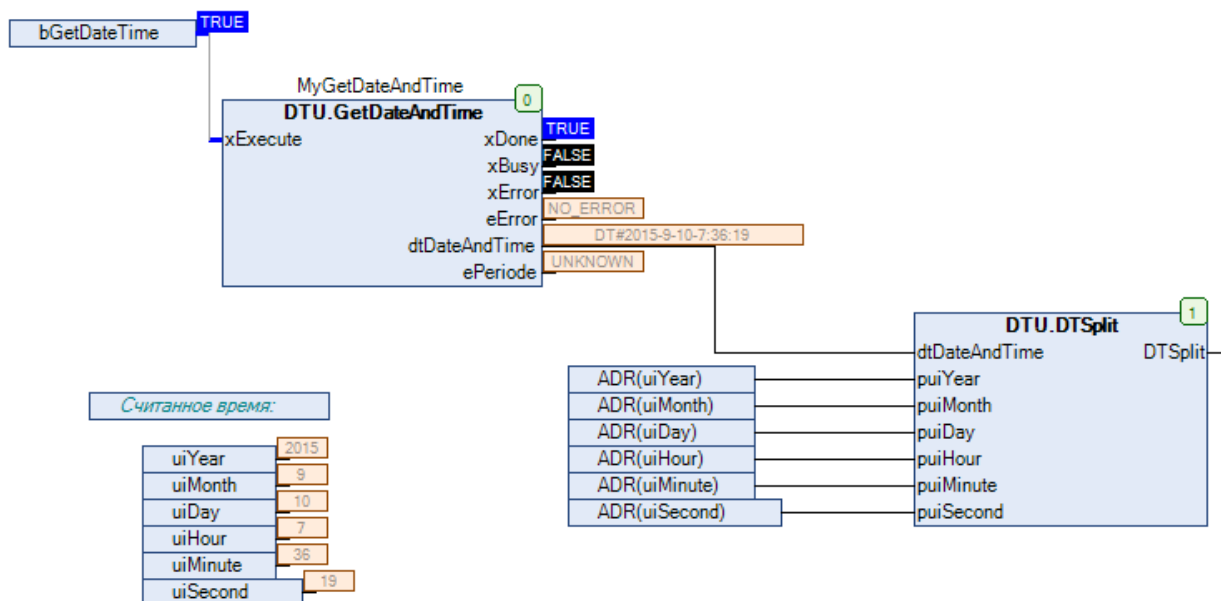


Рис. 10.4. Программа считывания системного времени в процессе работы

Обратите внимание, что считывание системного времени происходит не циклически, а по переднему фронту переменной **bGetDateTime**.

13.4. Как изменить системное время из кода программы?

Проще всего это сделать с помощью функции **DTConcat** и функционального блока **SetDateAndTime** из системной библиотеки **CAA_DTUtil**, расположенной в группе библиотек **Time and Date**.

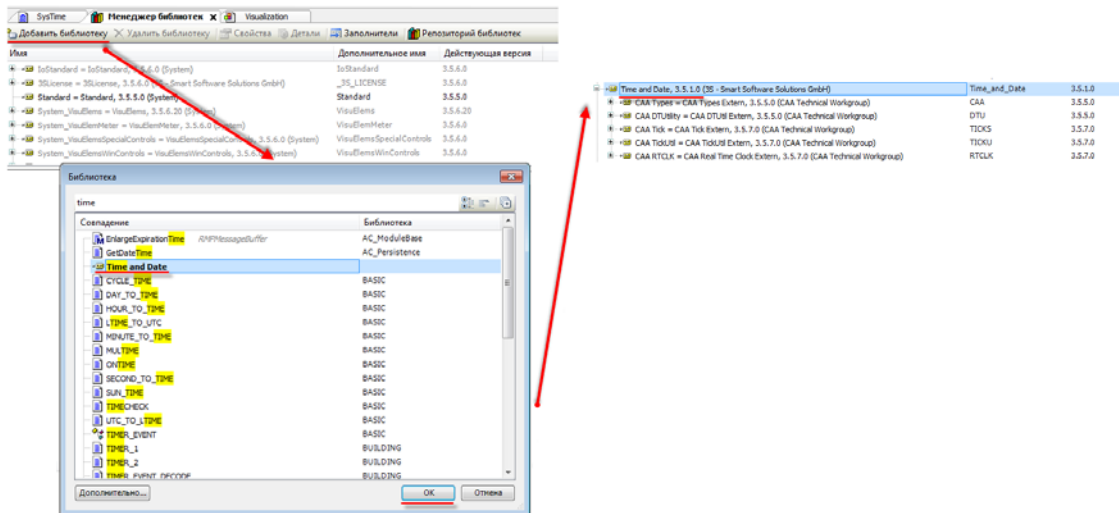


Рис. 10.5. Добавление в проект группы библиотек **Time and Date**

Пример программы изменения системного времени приведен ниже:

```

1  PROGRAM POU_2
2  VAR
3      uiYear:UINT;
4      uiMonth:UINT;
5      uiDay:UINT;
6      uiHour:UINT;
7      uiMinute:UINT;
8      uiSecond:UINT;
9
10     eError:WORD;
11
12     bSetDateTime:BOOL;
13
14     MySetDateAndTime: RTCLK.SetDateAndTime;
15 END_VAR
16
17

```

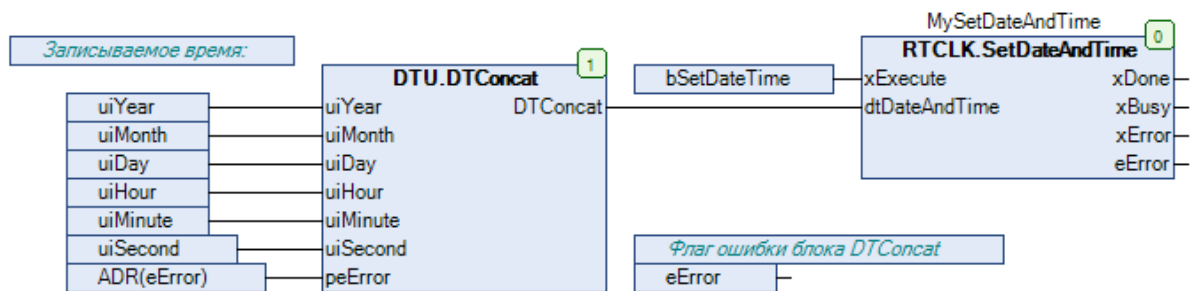


Рис. 10.6. Программа изменения системного времени в редакторе **CFC**

Когда переменная **bSetDateTime** принимает значение **TRUE**, время в формате **DT (DATE_AND_TIME)**, сформированное из значений отдельных разрядов (**uiYear**, **uiMonth** и т.д.) поступает на вход экземпляра **MySetDateAndTime** функционального блока **SetDateAndTime**, что приводит к изменению системного времени контроллера.

Обратите внимание, что переменные разрядов времени могут быть использованы в визуализации для изменения системного времени оператором в процессе работы проекта.

13.5. Что делать, если системное время постоянно сбивается?

Если системное время постоянно сбивается, то вероятнее всего разряжен элемент питания. Замените элемент питания, если проблема сохранится – обратитесь в [техподдержку](#) компании ОВЕН.

13.6. Как настроить синхронизацию контроллера с источником точного времени по протоколу SNTP?

В настоящее время данный функционал не поддерживается. Его появление ожидается в прошивке **3.950**, запланированной к выходу в августе-сентябре 2015 года.

14. Остальные вопросы

14.1. Для чего используется функция SysExecute?

Функция **SysExecute** из [библиотеки CmpSysExec](#) позволяет обращаться к командной оболочке **Linux** контроллера и с помощью нее выполнять системные команды. Записывая через эту функцию информацию в системные файлы, можно из кода программы в **CODESYS** реализовать управление подсветкой дисплея контроллера, индикаторами, аппаратными кнопками и т.д. Для того чтобы узнать путь к нужному файлу, необходимо подключиться к терминалу СПК с помощью утилиты **Putty** (см. [п. 14.8.](#)).

Поскольку функция **SysExecute** является довольно ресурсоемкой, **крайне не рекомендуется** вызывать ее циклически. Если по каким-то причинам это все же необходимо, то рекомендуемое минимальное время цикла задачи – 500 мс. В процессе работы необходимо проверить максимальное время цикла на вкладке **Monitor** компонента **Конфигурация задач** и, при необходимости, увеличить время цикла задачи.

В целом, **рекомендуется** применять функцию совместно с инструкцией **IF**:

```
IF LogicCondition = TRUE THEN
    SysExecute('системная команда');
    LogicCondition:=FALSE;
END_IF
```

Рис. 14.1. Применение функции **SysExecute**

На данный момент большая часть описанного в главе функционала (управление подсветкой, зуммером, считывание состояния аппаратных кнопок) реализовано в библиотеке **SPKtools**, доступной на сайте [Овен](#) в разделе **CODESYS V3/Библиотеки**.

14.2. Как настроить яркость подсветки?

Информация о яркости подсветки хранится в файловой системе СПК в файле **Brightness**. Изменить ее значение можно с помощью функции **SysExecute** (см. [п. 14.1](#) и [рис. 14.1](#)).

1. СПК1xx

Диапазон доступных для записи значений – от **0** (черный экран) до **100** (нормальная яркость). Тусклому экрану соответствует значение **30**.

Пример вызова функции:

```
SysExecute('echo "30" > /sys/class/backlight/pwm-backlight/brightness &');
```

2. СПК207 [M04]

Диапазон доступных для записи значений – от **0** (черный экран) до **255** (нормальная яркость). Тусклому экрану соответствует значение **70**.

Пример вызова функции:

```
SysExecute('echo "70" > /sys/class/backlight/som02-backlight/brightness &');
```

3. СПК207 [M02]

Диапазон доступных для записи значений – от **0** (черный экран) до **255** (нормальная яркость). Тусклому экрану соответствует значение **70**.

Пример вызова функции:

```
SysExecute('echo "70" > /sys/class/backlight/pwm-backlight/brightness &');
```

4. СПК207 [M01]

Диапазон доступных для записи значений – от **0** (черный экран) до **100** (нормальная яркость). Тусклому экрану соответствует значение **30**.

Пример вызова функции:

```
SysExecute('echo "30" > /sys/class/backlight/backlight/brightness &');
```

Обратите внимание, что путь к файлу может различаться в зависимости от модели и модификации СПК. Определить путь к файлу можно с помощью утилиты **WinSCP** (см. [п. 14.10](#)).

Упростить работу с данным функционалом может использование ФБ **Brightness** и **Touch_Activity** библиотеки [SPKtools](#), доступной на сайте [Овен](#) в разделе **CODESYS V3/Библиотеки**.

14.3. Как использовать аппаратные кнопки?

Аппаратные кнопки присутствуют только у СПК2хх. Каждой аппаратной кнопке соответствует свое значение: **SYS** – 0, **F1** – 1, **F2** – 2, ..., **F5** – 5. В **CODESYS** можно привязать значение аппаратной кнопки к элементу визуализации во вкладке **Inputconfiguration** – **Горячая клавиша**:

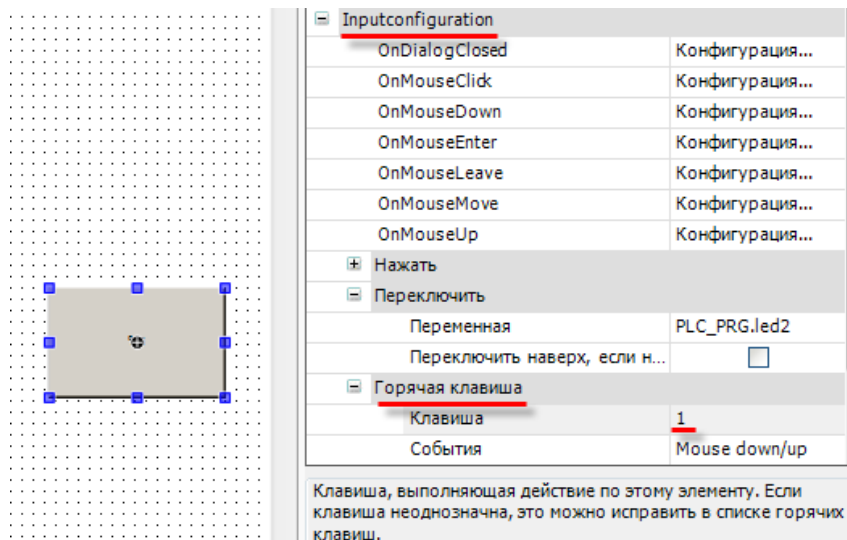


Рис. 14.2.Привязка аппаратной кнопки **F1** к элементу

В данном случае, по нажатию аппаратной кнопки **F1**, будет переключаться логическая переменная, привязанная к элементу визуализации.

Осуществить чтение состояния аппаратных кнопок можно с помощью ФБ **Keys** библиотеки [SPKtools](#), доступной на сайте [Овен](#) в разделе **CODESYS V3/Библиотеки**

14.4. Как использовать индикаторы аппаратных кнопок?

Индикаторы аппаратных кнопок присутствуют *только* у СПК2хх. Информация о состоянии индикаторов (включены/отключены) хранится в папках **sv1**, **sv2** и т.д. (где **sv1** – индикатор кнопки **SYS**, **sv2** – индикатор кнопки **F1** и т.д.) в файле **Brightness**. Включить/выключить индикатор можно с помощью функции **SysExecute** (см. [п. 14.1](#) и [рис. 14.1](#)). *Обратите внимание*, что нажатие аппаратной кнопки приводит к включению соответствующего индикатора, отпускание – к его отключению.

Пример вызова функции на СПК1хх / СПК207 [M01] / СПК207 [M02]:

```
SysExecute('echo "1" > /sys/class/leds/sv1/brightness &'); // индикатор включен
```

```
SysExecute('echo "0" > /sys/class/leds/sv1/brightness &'); // индикатор отключен
```

Пример вызова функции на СПК207 [M04]:

```
SysExecute('echo "255" > /sys/class/leds/sv1/brightness &'); // индикатор включен
```

```
SysExecute('echo "0" > /sys/class/leds/sv1/brightness &'); // индикатор отключен
```

Обратите внимание, что путь к файлу может различаться в зависимости от модели и модификации СПК. Определить путь к файлу можно с помощью утилиты **WinSCP** (см. [п. 14.10](#)).

14.5. Как использовать звукоизлучатель (зуммер, «пищалку»)?

Информация о состоянии звукоизлучателя (включен/отключен) хранится в папке **Buzzer** в файле **Brightness**. Включить/выключить индикатор можно с помощью функции **SysExecute** (см. [п. 14.1](#) и [рис. 14.1](#)).

Пример вызова функции на **СПК1хх**:

```
SysExecute('echo "1" > /sys/class/leds/pwm-beeper/brightness &'); // зуммер включен
```

```
SysExecute('echo "0" > /sys/class/leds/pwm-beeper/brightness &'); // зуммер отключен
```

Пример вызова функции на **СПК207 [M04]**:

```
SysExecute('echo "100" > /sys/class/leds/buzzer/brightness &'); // зуммер включен
```

```
SysExecute('echo "0" > /sys/class/leds/buzzer/brightness &'); // зуммер отключен
```

Пример вызова функции на **СПК207 [M01]/СПК207 [M02]**:

```
SysExecute('echo "1" > /sys/class/leds/buzzer/brightness &'); // зуммер включен
```

```
SysExecute('echo "0" > /sys/class/leds/buzzer/brightness &'); // зуммер отключен
```

Обратите внимание, что путь к файлу может различаться в зависимости от модели и модификации СПК. Определить путь к файлу можно с помощью утилиты **WinSCP** (см. [п. 14.10](#)).

Упростить работу с данным функционалом может использование ФБ **Buzzer** библиотеки [SPKtools](#), доступной на сайте [Овен](#) в разделе **CODESYS V3/Библиотеки**.

14.6. Как управлять аудиовыходом?

СПК поддерживает воспроизведение аудиофайлов формата **.wav**. При этом к аудиовыходу СПК должно быть подключено соответствующее устройство (колонки, наушники и т.д.). Воспроизведение аудио осуществляется с помощью функции **SysExecute** (см. [п. 14.1](#) и [рис. 14.1](#)). Предварительно в файловую систему СПК должен быть загружен соответствующий файл формата **.wav**. Это можно сделать с помощью утилиты **WinSCP** (см. [п. 14.10](#)).

Пример вызова функции:

```
6 IF PlaySound=TRUE THEN
7     SysExecute('/usr/bin/aplay -c2 -t raw -r 44100 /mnt/ufs/root/violin.wav &');
8     PlaySound:=FALSE;
9 END_IF
```

Когда логическая переменная **PlaySound** принимает значение **TRUE**, контроллер начинает воспроизведение файла **violin.wav**, расположенного в папке **/mnt/ufs/root**. Частота дискретизации файла – 44100 Гц.

Чтобы прервать воспроизведение файла, воспользуйтесь следующим кодом:

```
4 IF BreakSound=TRUE THEN
5     SysExecute('killall aplay &');
6     BreakSound:=FALSE;
7 END_IF
```

Когда логическая переменная **BreakSound** принимает значение **TRUE**, воспроизведение аудиофайла прерывается.

Для управления громкостью необходимо использовать ФБ **Volume** из библиотеки [SPKtools](#), доступной на сайте [Овен](#) в разделе **CODESYS V3/Библиотеки**.

14.7. Как перезагрузить контроллер из кода программы?

Перезагрузить контроллер из кода программы можно с помощью функции **SysExecute** (см. [п. 14.1.1.](#) и [рис. 14.1.](#)).

Пример вызова функции:

```
IF reset=TRUE THEN SysExecute('/sbin/reboot');  
END_IF
```

Когда логическая переменная **RESET** принимает значение **TRUE**, контроллер перезагружается. При необходимости можно привязать переменную к кнопке на экране визуализации.

14.8. Как организовать терминальное подключение к контроллеру?

Рекомендуется использовать утилиту **Putty**, которая находится на диске с ПО из комплекта поставки. Если СПК подключен к ПК кросс-кабелем, то в настройках **Putty** (вкладка **Сеанс**) следует выбрать тип соединения **SSH** и указать IP-адрес контроллера; номер порта оставить по умолчанию – 22. Нажать кнопку **Соединиться**.

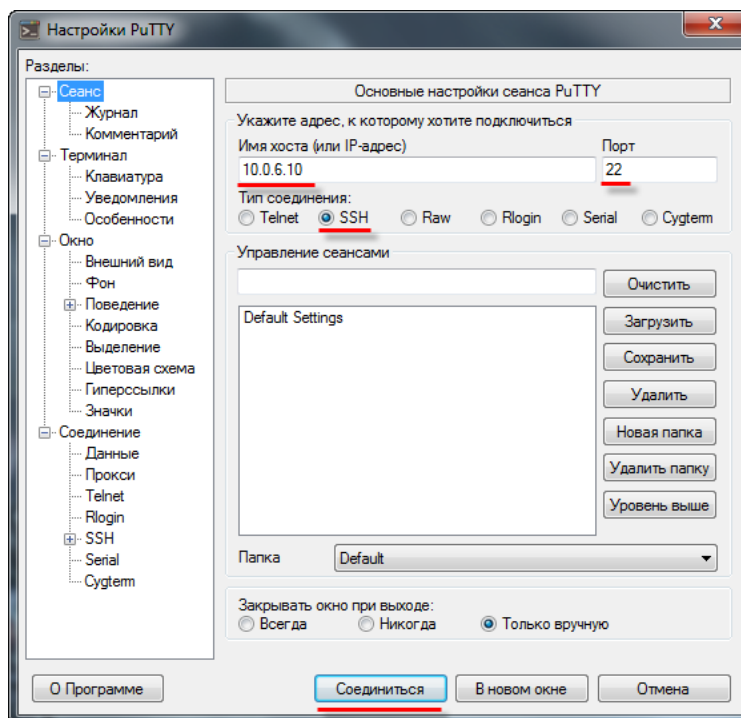


Рис. 14.3. Настройки подключения в программе **Putty** (с использованием ethernet порта)

Если появится окно **Предупреждение безопасности**, следует нажать **ОК**.

В открывшемся терминале в поле **login as** ввести **root**, поле **password** оставить пустым, нажать **Enter**.

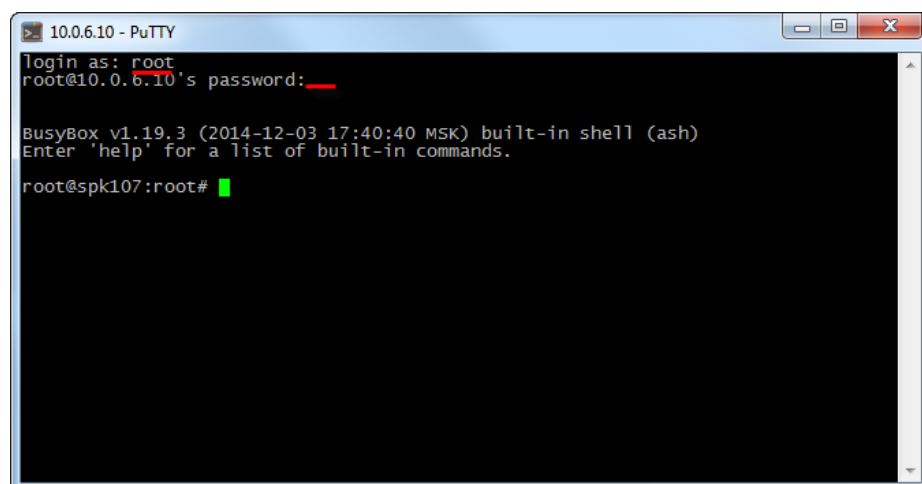


Рис. 14.4. Аутентификация пользователя при подключении к терминалу

Если использовать Ethernet-порт не представляется возможным (например, он занят другим устройством), то можно подключить СПК к ПК при помощи debug кабеля (**КС2**) из комплекта поставки и в настройках **PuTTY** (вкладка **Сеанс**) выбрать тип соединения **Serial**, скорость **115200** и указать номер COM-порта компьютера (отображается в **Диспетчере устройств**):

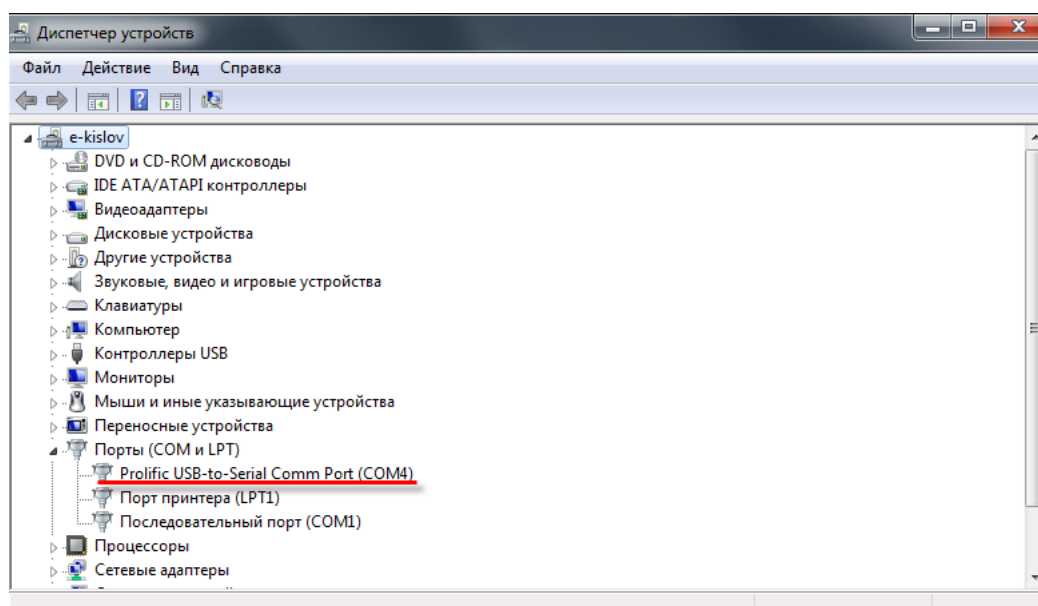


Рис. 14.5 Номер COM-порта компьютера в Диспетчере устройств

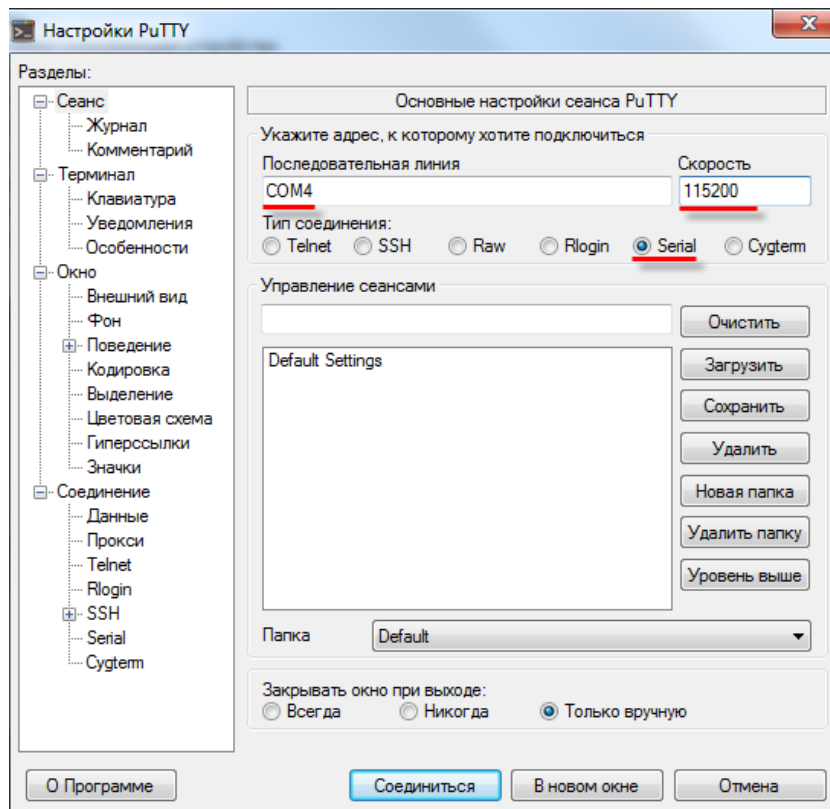
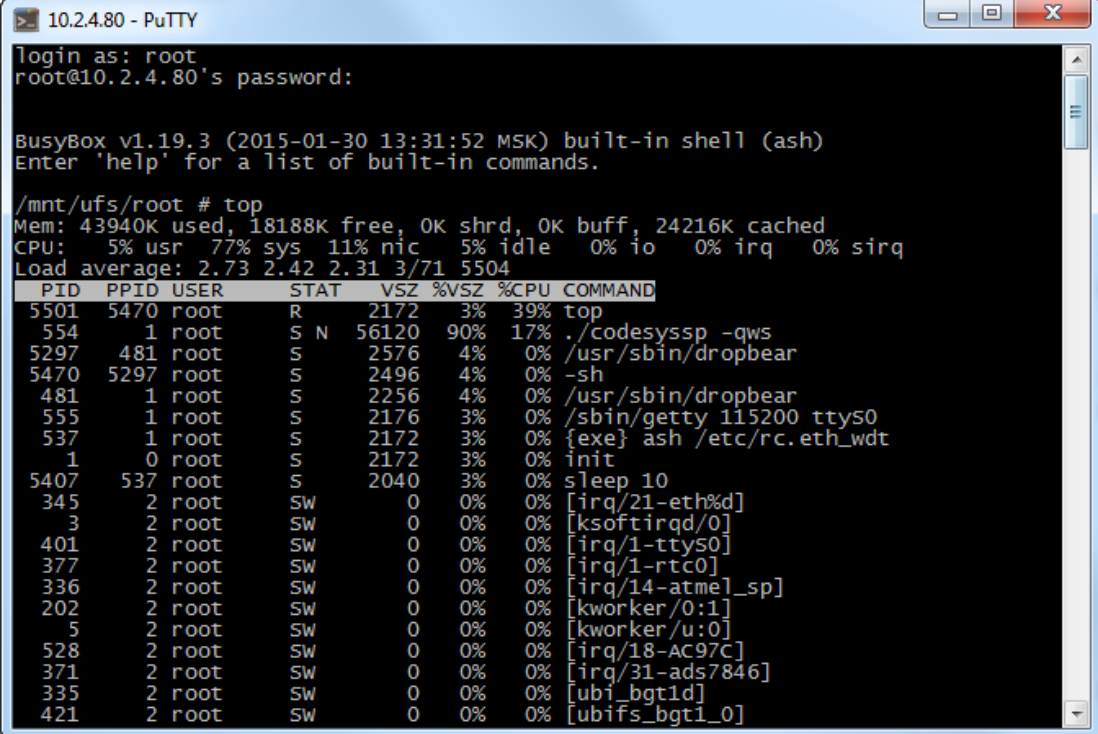


Рис. 14.6. Настройки подключения в программе **PuTTY** (с использованием debug порта)

14.9. Как открыть в Linux контроллера диспетчер задач?

Для этого необходимо подключиться с помощью утилиты **Putty** (см. [п. 14.8](#)) и выполнить команду **top**. Выход из диспетчера задач осуществляется по нажатию клавиши **Q**.



```
10.2.4.80 - PuTTY
login as: root
root@10.2.4.80's password:

BusyBox v1.19.3 (2015-01-30 13:31:52 MSK) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/mnt/ufs/root # top
Mem: 43940K used, 18188K free, 0K shrd, 0K buff, 24216K cached
CPU:  5% usr 77% sys 11% nic  5% idle  0% io  0% irq  0% sirq
Load average: 2.73 2.42 2.31 3/71 5504
  PID  PPID  USER  STAT  VSZ  %VSZ  %CPU  COMMAND
 5501  5470  root   R      2172   3%   39%  top
   554     1  root   S N    56120  90%  17%  ./codesysssp -qws
 5297  481  root   S      2576   4%   0%  /usr/sbin/dropbear
 5470  5297  root   S      2496   4%   0%  -sh
  481     1  root   S      2256   4%   0%  /usr/sbin/dropbear
  555     1  root   S      2176   3%   0%  /sbin/getty 115200 ttys0
  537     1  root   S      2172   3%   0%  {exe} ash /etc/rc.eth_wdt
    1     0  root   S      2172   3%   0%  init
 5407  537  root   S      2040   3%   0%  sleep 10
  345     2  root   SW      0   0%   0%  [irq/21-eth%d]
    3     2  root   SW      0   0%   0%  [ksoftirqd/0]
  401     2  root   SW      0   0%   0%  [irq/1-ttys0]
  377     2  root   SW      0   0%   0%  [irq/1-rtc0]
  336     2  root   SW      0   0%   0%  [irq/14-atmel_sp]
  202     2  root   SW      0   0%   0%  [kworker/0:1]
    5     2  root   SW      0   0%   0%  [kworker/u:0]
  528     2  root   SW      0   0%   0%  [irq/18-AC97C]
  371     2  root   SW      0   0%   0%  [irq/31-ads7846]
  335     2  root   SW      0   0%   0%  [ubi_bgt1d]
  421     2  root   SW      0   0%   0%  [ubifs_bgt1_0]
```

Рис. 14.7. Результат выполнения команды **top** в терминале контроллера

14.10. Как подключиться к файловой системе контроллера?

Рекомендуется использовать программу **WinSCP**, которая находится на диске с ПО из комплекта поставки. В ее настройках следует выбрать тип соединения **SCP** и указать IP-адрес контроллера; номер порта оставить по умолчанию – **22**. Имя пользователя – **root**, поле **Пароль** следует оставить пустым. Нажать кнопку **Войти**.

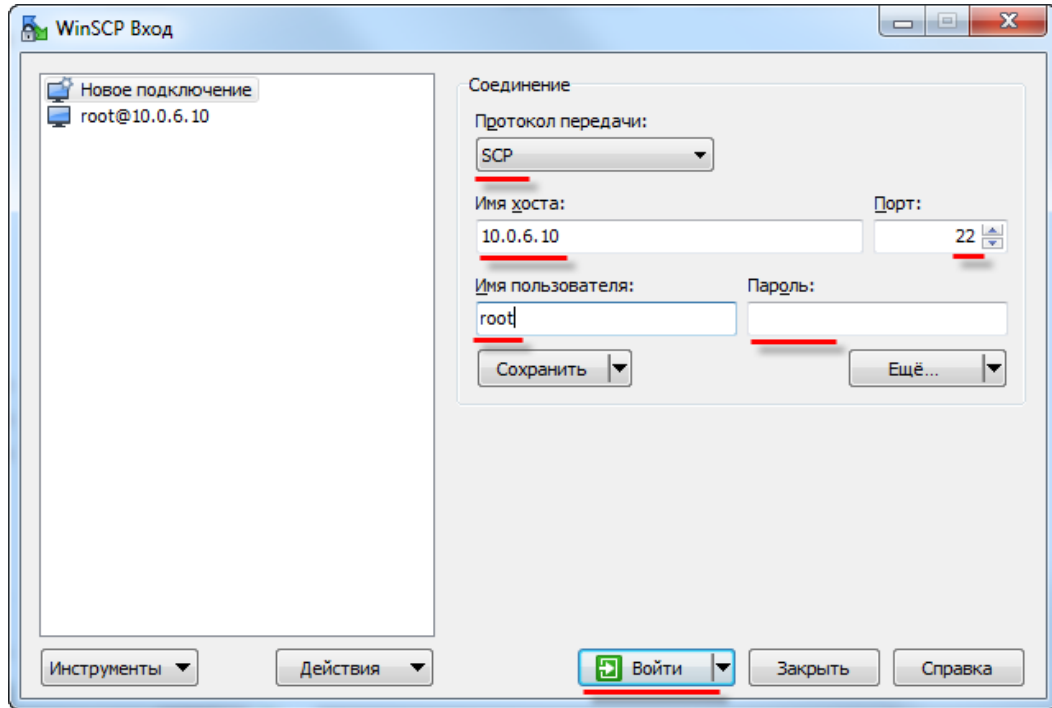


Рис. 14.8. Настройки подключения в программе WinSCP

В появившемся окне аутентификации нажать **ОК**.

В случае появления ошибки «**Не могу сменить каталог на '/'**» нажать **ОК**.

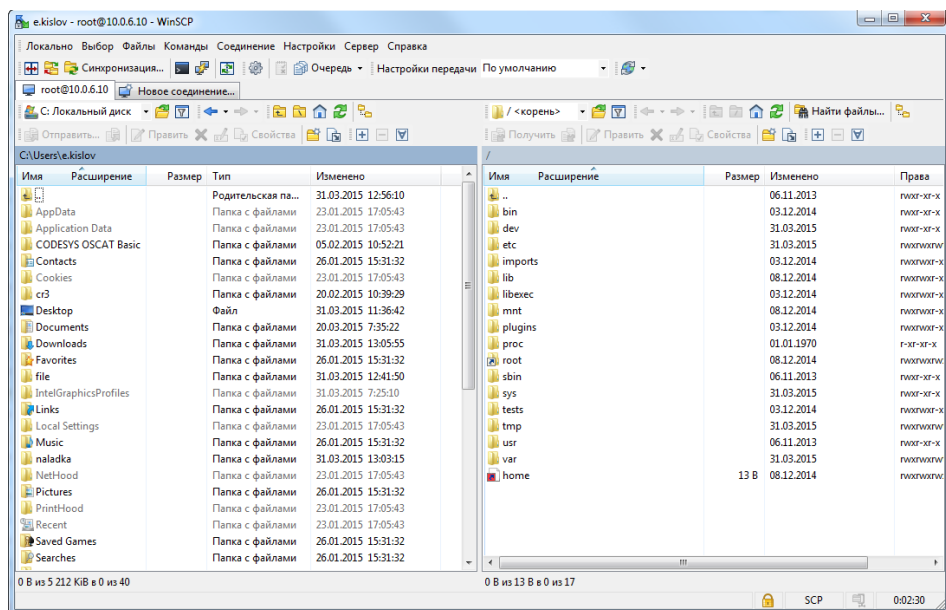


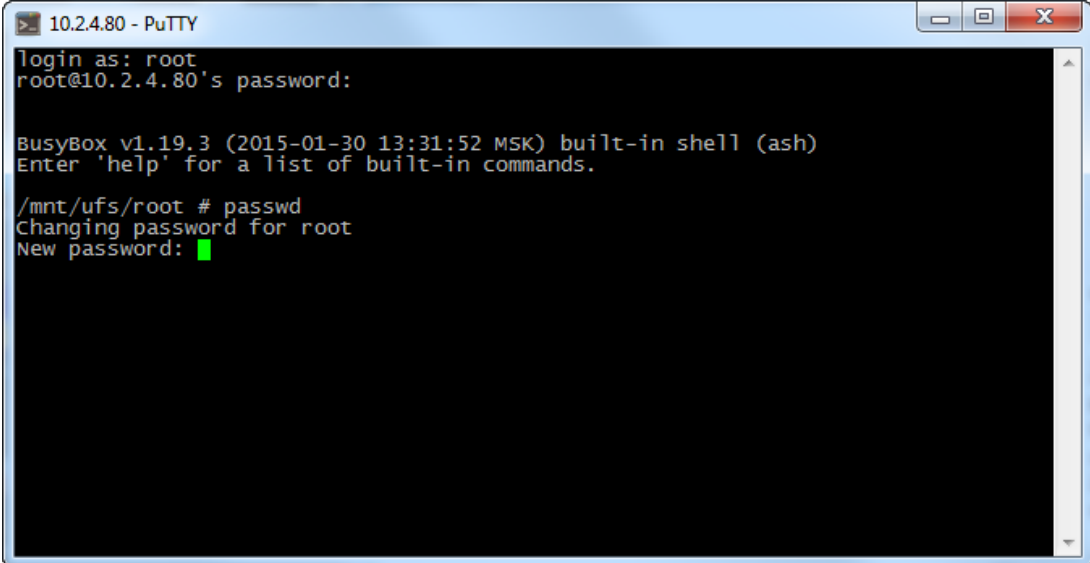
Рис. 14.9. Окно программы WinSCP

14.11. Как защитить доступ к Linux контроллера паролем?

Для этого необходимо подключиться с помощью утилиты **Putty** (см. [п. 14.8](#)) и выполнить команду **passwd**. После этого необходимо два раза ввести новый пароль.

Рекомендации по выбору пароля:

1. пароль должен содержать не менее 6 символов;
2. пароль должен содержать как минимум две буквы (большие или малые) и хотя бы одну цифру или знак;
3. пароль должен отличаться от имени пользователя, прочитанного слева направо или задом наперед, и от его циклических сдвигов. При сравнении не делается различий между большими и малыми буквами;
4. новый пароль должен отличаться от старого хотя бы тремя символами. При сравнении не делается различий между большими и малыми буквами.



```
10.2.4.80 - PuTTY
login as: root
root@10.2.4.80's password:

BusyBox v1.19.3 (2015-01-30 13:31:52 MSK) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/mnt/ufs/root # passwd
Changing password for root
New password: █
```

Рис. 14.10. Результат выполнения команды **passwd** в терминале контроллера