

Комплекс программно-технический Fastwel I/O-2

**Контроллер программируемый универсальный
СРМ810-03**

Руководство по конфигурированию и программированию

ИМЕС.00300-03 33 02-2

Версия 2.0

Настоящий документ содержит информацию о принципах функционирования, а также указания по разработке прикладных программ и настройке системных параметров контроллера программируемого СРМ810-03.

Торговые марки

ДОЛОМАНТ™, ФАСТВЕЛ™, Fastwel™ – официально зарегистрированные торговые марки ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ», Москва, Российская Федерация.

Кроме того, настоящий документ может содержать наименования, фирменные логотипы и торговые марки, являющиеся зарегистрированными торговыми марками, а следовательно, права собственности на них принадлежат их законным владельцам.

Права собственности

Настоящий документ содержит информацию, которая является интеллектуальной собственностью ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ».

Настоящий документ не может быть скопирован или передан с использованием известных средств, а также не может храниться в системах хранения и поиска информации без предварительного письменного согласия ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ» или одного из ее уполномоченных агентов. Информация, содержащаяся в настоящем документе, насколько нам известно, не содержит ошибок, однако, ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ» не может принять на себя ответственность за какие-либо неточности и их последствия, а также ответственность, возникающую в результате использования или применения любой схемы, продукта или примера, приведенного в настоящем документе. ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ» оставляет за собой право изменять и усовершенствовать как настоящий документ, так и представленный в нем продукт по своему усмотрению без дополнительно извещения.

Контактная информация

Изготовитель – ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ»:

Почтовый адрес: Россия, 117342, Москва, ул. Введенского, д.3

Телефон: +7 (495) 232-2033

Факс: +7 (495) 232-1654

Электронная почта: info@dolomant.ru

Web: <http://www.dolomant.ru>

Служба технической поддержки:

Телефон: +7 (495) 232-1698

Электронная почта: fi@fastwel.ru
support@fastwel.ru

Авторское право

Это Руководство не может быть скопировано, воспроизведено, переведено или конвертировано в любую электронную или машиночитаемую форму без предварительного письменного разрешения ЗАО «НАУЧНО-ПРОИЗВОДСТВЕННАЯ ФИРМА «ДОЛОМАНТ».

СОДЕРЖАНИЕ

1.	ВВЕДЕНИЕ	7
1.1.	ИНФОРМАЦИЯ О ДОКУМЕНТЕ	7
1.2.	ЭЛЕМЕНТЫ ОФОРМЛЕНИЯ ТЕКСТА	7
1.3.	ПРИНЯТЫЕ СОКРАЩЕНИЯ И ТЕРМИНЫ	7
1.4.	CIDR-НОТАЦИЯ ДЛЯ ПРЕДСТАВЛЕНИЯ IP-АДРЕСОВ	9
2.	ОБЩИЕ СВЕДЕНИЯ	10
2.1.	НАЗНАЧЕНИЕ FASTWEL I/O-2	10
2.2.	НАЗНАЧЕНИЕ СРМ810-03	10
2.3.	СТРУКТУРА АППАРАТНЫХ СРЕДСТВ	10
2.3.1.	Микропроцессор	10
2.3.2.	Межмодульная шина	10
2.3.3.	Порты Ethernet	11
2.3.4.	Порты RS-232C	11
2.3.5.	Порты RS-485	12
2.3.6.	Порты USB	12
2.3.7.	Порт дискретного ввода-вывода	12
2.3.8.	Порт VGA	12
2.3.9.	Дисковые накопители	12
2.3.10.	Энергонезависимая память	12
2.3.11.	Часы-календарь	12
2.3.12.	Органы управления и индикации	13
2.4.	СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	13
2.4.1.	Состав программного обеспечения	13
2.4.2.	Пакет инструментальных средств ПЛК Fastwel (Fastwel PLC Application Toolkit)	13
2.4.2.1.	Назначение	13
2.4.2.2.	Варианты программы	14
2.4.2.3.	Состав	14
2.4.3.	Системное программное обеспечение контроллера	15
2.4.4.	Основные характеристики системного программного обеспечения	16
2.5.	СИСТЕМНЫЕ ТРЕБОВАНИЯ К РАБОЧЕМУ МЕСТУ РАЗРАБОТЧИКА	17
3.	УСКОРЕННОЕ ОЗНАКОМЛЕНИЕ	18
3.1.	ОБЩИЕ СВЕДЕНИЯ	18
3.2.	ПОДГОТОВКА К РАБОТЕ	19
3.2.1.	Цель работы	19
3.2.2.	Конфигурация аппаратных средств для учебного проекта	19
3.2.3.	Сетевые параметры контроллера	20
3.3.	НАСТРОЙКА ПАРАМЕТРОВ СИСТЕМНЫХ СЕРВИСОВ КОНТРОЛЛЕРА	21
3.4.	СОЗДАНИЕ ПРОЕКТА	21
3.5.	СОЗДАНИЕ КОНФИГУРАЦИИ МОДУЛЕЙ ВВОДА-ВЫВОДА	23
3.5.1.	Общие сведения	23
3.5.2.	Создание описания модулей ввода-вывода вручную	23
3.5.3.	Создание описания модулей ввода-вывода из списка сканирования контроллера	25
3.5.4.	Настройка параметров межмодульной шины и модулей ввода-вывода	26
3.5.5.	Создание связей приложения с каналами модуля ввода-вывода	27
3.5.5.1.	Общие сведения	27
3.5.5.2.	Соотнесение переменных функционального блока с каналами модуля ввода-вывода	29
3.6.	СОЗДАНИЕ КОНФИГУРАЦИИ СЕРВЕРА MODBUS TCP	31
3.6.1.	Общие сведения	31
3.6.2.	Создание и настройка объекта сервера MODBUS TCP	32
3.6.3.	Создание объектов доступа к данным	33
3.6.3.1.	Общие сведения	33
3.6.3.2.	Создание областей ввода и вывода сервера MODBUS	33
3.6.3.3.	Создание объектов доступа к данным сервера	34
3.6.4.	Создание связи приложения с коммуникационными объектами	37
3.7.	СОЗДАНИЕ КОНФИГУРАЦИИ МАСТЕРА MODBUS TCP	39
3.7.1.	Общие сведения	39
3.7.2.	Создание и настройка объекта мастера MODBUS TCP	40
3.7.3.	Создание связи приложения с коммуникационными объектами мастера MODBUS TCP	45
3.8.	РАЗРАБОТКА ПРОГРАММЫ	46
3.8.1.	Общие сведения	46
3.8.2.	Функциональный блок ACTIVITY_CHECKER	47
3.8.2.1.	Общие сведения	47
3.8.2.2.	Создание функционального блока (FUNCTION_BLOCK)	48

3.8.2.3.	Использование функционального блока	50
3.8.3.	<i>Алгоритм сервера MODBUS TCP</i>	51
3.8.3.1.	Общие сведения	51
3.8.3.2.	Декларация переменных PLC_PRG	51
3.8.3.3.	Исходный текст алгоритма PLC_PRG	52
3.8.3.4.	Соотнесение счетчиков транзакций и ошибок сервера MODBUS TCP	53
3.8.4.	<i>Алгоритм мастера MODBUS TCP</i>	54
3.8.4.1.	Общие сведения	54
3.8.4.2.	Создание программы TCP_CLIENT	54
3.8.4.3.	Создание задачи <i>ClientTask</i> и установка вызова <i>TCP_CLIENT</i>	55
3.8.4.4.	Декларация переменных TCP_CLIENT	56
3.8.4.5.	Исходный текст TCP_CLIENT	57
3.8.4.6.	Соотнесение переменных алгоритма <i>TCP_CLIENT</i> с объектами данных клиента MODBUS TCP	58
3.9.	ТРАНСЛЯЦИЯ ПРИЛОЖЕНИЯ И ЗАГРУЗКА В КОНТРОЛЛЕР	59
3.9.1.	<i>Общие сведения</i>	59
3.9.2.	<i>Загрузка приложения учебного проекта в контроллер</i>	60
3.10.	ОТЛАДКА ПРИЛОЖЕНИЯ	62
3.10.1.	<i>Общие сведения</i>	62
3.10.2.	<i>Мониторинг переменных</i>	62
3.10.3.	<i>Просмотр статистики выполнения задач</i>	65
3.10.4.	<i>Пошаговая отладка</i>	65
3.11.	МОДИФИКАЦИЯ ПРИЛОЖЕНИЯ В КОНТРОЛЛЕРЕ	67
3.11.1.	<i>Общие сведения</i>	67
3.11.2.	<i>Модификация приложения методом полной загрузки</i>	68
3.11.2.1.	Общие сведения	68
3.11.2.2.	Создание функционального блока с наследованием интерфейса и реализации	68
3.11.2.3.	Создание метода FB_Exit функционального блока	70
3.11.2.4.	Использование USER_LED_EXT и загрузка приложения в контроллер	71
3.11.3.	<i>Модификация приложения онлайн-изменением (Online Change)</i>	71
3.11.3.1.	Общие сведения	71
3.11.3.2.	Изменение функционального блока USER_LED_EXT и онлайн-изменение приложения	72
3.11.4.	<i>Идентификационная информация проекта</i>	73
3.12.	ДИАГНОСТИКА КРИТИЧЕСКИХ ОТКАЗОВ ПРИЛОЖЕНИЯ	75
3.12.1.	<i>Общие сведения</i>	75
3.12.2.	<i>Поиск местоположения бесконечного цикла в коде приложения</i>	76
3.12.2.1.	Общие сведения	76
3.12.2.2.	Ошибка заикливания в учебном проекте	77
3.12.2.3.	Получение информации о местоположении заикливания	77
3.12.3.	<i>Поиск местоположения ошибки доступа к памяти в коде приложения</i>	79
3.12.3.1.	Общие сведения	79
3.12.3.2.	Ошибка доступа к памяти в учебном проекте	79
3.12.3.3.	Получение информации о местоположении неправильного доступа к памяти	80
3.12.4.	<i>Поиск местоположения ошибки деления на 0 в коде приложения</i>	81
3.12.4.1.	Общие сведения	81
3.12.4.2.	Ошибка деления на 0 в учебном проекте	81
3.12.4.3.	Получение информации о местоположении ошибки деления на 0	82
4.	ПРИНЦИП РАБОТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОНТРОЛЛЕРА	84
4.1.	ОБЩИЕ СВЕДЕНИЯ	84
4.2.	РЕЖИМЫ РАБОТЫ КОНТРОЛЛЕРА	84
4.2.1.	<i>Режим с исходными (заводскими) настройками</i>	84
4.2.2.	<i>Нормальный режим</i>	85
4.2.3.	<i>Безопасный режим</i>	86
4.2.4.	<i>Получение информации о режиме работы контроллера по сети</i>	89
4.2.4.1.	Общие сведения	89
4.2.4.2.	Получение информации о режиме работы контроллера по MODBUS	89
4.2.4.3.	Получение информации о режиме работы контроллера через OPC UA	89
4.3.	ПРОЦЕСС ЗАПУСКА КОНТРОЛЛЕРА	90
4.3.1.	<i>Запуск при первом включении питания</i>	90
4.3.2.	<i>Запуск при наличии загруженного приложения</i>	90
4.4.	СИСТЕМА ИСПОЛНЕНИЯ ПРИЛОЖЕНИЙ МЭК 61131-3	91
4.4.1.	<i>Общие сведения</i>	91
4.4.2.	<i>Использование подсистемы целевой визуализации (Target Visualization)</i>	91
4.4.2.1.	Общие сведения	91
4.4.2.2.	Активация подсистемы целевой визуализации	92
4.4.2.3.	Удаленный доступ к экранным формам целевой визуализации	93
4.4.3.	<i>Диагностические каналы системы исполнения</i>	93
4.5.	СЕРВИС ВВОДА-ВЫВОДА	95
4.5.1.	<i>Общие сведения</i>	95

4.5.2.	<i>Конфигурация сервиса ввода-вывода</i>	95
4.5.2.1.	Параметры сервиса ввода-вывода	95
4.5.2.2.	Соотнесение удаленного адаптера шины FBUS с IP-адресом сетевого интерфейса	97
4.5.2.3.	Создание конфигурации модулей ввода-вывода	99
4.5.3.	<i>Диагностика</i>	103
4.5.3.1.	Индикация	103
4.5.3.2.	Диагностические каналы сервиса ввода-вывода	103
4.5.4.	<i>Порты последовательной передачи данных на базе модулей NIM741/2, NIM841/2</i>	105
4.5.4.1.	Общие сведения	105
4.5.4.2.	Идентификация последовательных портов на базе NIM741/841, NIM742/842	105
4.5.4.3.	Доступ к последовательным портам из приложения	106
4.5.5.	<i>Получение информации о подключенных модулях ввода-вывода</i>	107
4.6.	СЕТЕВЫЕ СЕРВИСЫ	110
4.6.1.	<i>Общие сведения о веб-конфигураторе контроллера</i>	110
4.6.2.	<i>Настройка IP-параметров</i>	111
4.6.2.1.	Общие сведения	111
4.6.2.2.	Статическое назначение IP-адресов	111
4.6.2.3.	Динамическое назначение IP-адресов	114
4.6.2.4.	Настройка IP-маршрутизации	114
4.6.3.	<i>Сервис подчиненного узла протокола MODBUS</i>	116
4.6.3.1.	Общие сведения	116
4.6.3.2.	Сервис подчиненного узла протокола MODBUS TCP	118
4.6.3.3.	Сервис подчиненного узла протокола MODBUS RTU/ASCII	120
4.6.4.	<i>Сервис мастера протокола MODBUS</i>	121
4.6.4.1.	Общие сведения	121
4.6.4.2.	Сервис мастера протокола MODBUS TCP	123
4.6.4.3.	Сервис мастера протокола MODBUS RTU/ASCII	125
4.6.5.	<i>Сервис протокола NTP</i>	127
4.6.5.1.	Общие сведения	127
4.6.5.2.	Настройка клиента NTP	127
4.6.5.3.	Настройка сервера NTP	129
4.6.6.	<i>Сервис протокола RTP</i>	129
4.6.7.	<i>Сервис протокола FTP</i>	131
5.	НАСТРОЙКА СИСТЕМНЫХ ПАРАМЕТРОВ	133
5.1.	ОБЩИЕ СВЕДЕНИЯ	133
5.2.	ПАРАМЕТРЫ СЕТИ	134
5.3.	IP-МАРШРУТИЗАЦИЯ	135
5.4.	МАРШРУТИЗАЦИЯ CODESYS	135
5.5.	ПАРАМЕТРЫ MODBUS	135
5.5.1.	<i>Общие сведения</i>	135
5.5.2.	<i>Настройка параметров серверов MODBUS TCP</i>	136
5.5.3.	<i>Настройка параметров мастеров MODBUS TCP</i>	137
5.6.	УДАЛЕННЫЙ FBUS	137
5.7.	УСТАНОВКА ВРЕМЕНИ	137
5.8.	ПАРАМЕТРЫ NTP	138
5.9.	ПАРАМЕТРЫ RTP	138
5.10.	ПАРАМЕТРЫ OPC UA	139
5.11.	СЕРТИФИКАТЫ БЕЗОПАСНОСТИ	140
5.12.	ВИДЕО	144
5.13.	ВЕБ-КОНФИГУРАТОР	145
5.14.	УСТАНОВКА ПАРОЛЕЙ	147
5.15.	ОБОЛОЧКА ПЛК	148
5.16.	СИСТЕМА	149
5.16.1.	<i>Общие сведения</i>	149
5.16.2.	<i>Включение и отключение поддержки USB-устройств</i>	149
5.16.3.	<i>Загрузка системных файлов</i>	149
5.16.4.	<i>Восстановление системных параметров контроллера</i>	150
5.16.5.	<i>Перезапуск контроллера</i>	150
5.17.	НАСТРОЙКА ПУТЕЙ	150
5.18.	ЖУРНАЛ ПЛК	153
6.	СИСТЕМНЫЕ БИБЛИОТЕКИ	158
6.1.	ОБЩИЕ СВЕДЕНИЯ	158
6.2.	БИБЛИОТЕКА ДОСТУПА К ПОРТУ ДИСКРЕТНОГО ВВОДА-ВЫВОДА FASTWELDIO	158
6.2.1.	<i>Общие сведения</i>	158
6.2.2.	<i>Функции</i>	159
6.2.2.1.	<i>FwDIO_getPortsCount</i>	159

6.2.2.2.	FwDIO_getLinesCount	159
6.2.2.3.	FwDIO_setDirection	160
6.2.2.4.	FwDIO_getDirection.....	160
6.2.2.5.	FwDIO_setLineDirection	161
6.2.2.6.	FwDIO_getLineDirection.....	161
6.2.2.7.	FwDIO_getPort	161
6.2.2.8.	FwDIO_readPort	161
6.2.2.9.	FwDIO_setPort.....	162
6.2.2.10.	FwDIO_writePort.....	162
6.2.2.11.	FwDIO_getLine	163
6.2.2.12.	FwDIO_readLine.....	163
6.2.2.13.	FwDIO_setLine	163
6.2.2.14.	FwDIO_writeLine	164
ПРИЛОЖЕНИЕ 1 . ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....		165

1. Введение

1.1. Информация о документе

Настоящий документ содержит сведения о принципах функционирования, а также указания по настройке и программированию контроллера программируемого СРМ810-03.

Контроллер программируемый СРМ810-03 будет далее называться СРМ810-03, контроллер СРМ810-03 или контроллер.

При работе с настоящим документом следует также пользоваться документами:

1. ИМЕС.00390-03 33 01. Пакет инструментальных средств ПЛК Fastwel. Fastwel PLC Application Toolkit. Руководство пользователя
2. ИМЕС.421459.252РЭ. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 1. Общие сведения
3. ИМЕС.421459.252РЭ1. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 2. Составные части комплекса
4. ИМЕС.00300-03 33 03. Протокол MODBUS. Руководство по конфигурированию и программированию
5. ИМЕС.00300-03 33 04. Протокол ГОСТ Р МЭК 60870-5-104. Руководство по конфигурированию и программированию

Перед началом разработки прикладного программного обеспечения контроллера следует ознакомиться с указаниями документа *ИМЕС.00390-03 33 01. Пакет инструментальных средств ПЛК Fastwel. Fastwel PLC Application Toolkit. Руководство пользователя.*

1.2. Элементы оформления текста

В настоящем документе используются следующие элементы дополнительного оформления текста:

<i>Текст</i>	Понятие или значение параметра.
Текст	Заголовок элемента пользовательского интерфейса (заголовок элемента управления, меню, окна и т.п.), заголовок рисунка или таблицы.
Текст	Фрагмент исходного текста программы, консольной команды или результата выполнения команды.
<i>// Текст</i>	Комментарий в исходном тексте программы.

1.3. Принятые сокращения и термины

В настоящем документе используются следующие сокращения и термины:

Astra.IDE	инструментальное программное обеспечение (интегрированная среда разработки) версии 1.7.0.0 и выше компании Prosoft-Systems Ltd., предназначенное для разработки приложений контроллеров, программируемых на языках стандарта IEC 61131-3 (ГОСТ Р МЭК 61131-3)
CDC	Communications Device Class, класс устройств последовательной передачи данных через интерфейс USB
CODESYS V3	инструментальное программное обеспечение (интегрированная среда разработки) версии 3.5 и выше компании CODESYS Development GmbH, предназначенное для разработки приложений контроллеров, программируемых на языках стандарта IEC 61131-3 (ГОСТ Р МЭК 61131-3)

DHCP	Dynamic Host Configuration Protocol, протокол динамического назначения IP-адресов
DNS	Domain Name System, система доменных имен, как правило используемая для получения IP-адреса узла сети по имени
Fastwel PLC Application Toolkit	Программный комплекс "ИМЕС.00390-03. Пакет инструментальных средств ПЛК Fastwel", предназначенный разработки прикладного программного обеспечения для контроллеров программируемых, выпускаемых под торговой маркой Fastwel, на языках программирования, определенных стандартом ГОСТ Р МЭК 61131-3 (IEC 61131-3), в совместимых интегрированных средах разработки
FTP	File Transfer Protocol, протокол передачи файлов по сети
HTTP	Hypertext Transfer Protocol, гипертекстовый протокол передачи данных по сети
IDE МЭК 61131-3	интегрированная среда разработки на языках стандарта IEC 61131-3 (ГОСТ Р МЭК 61131-3), совместимая с CODESYS V3 и/или Astra.IDE
IDE Gateway	коммуникационный сервис, обеспечивающий взаимодействие между IDE МЭК 61131-3 и контроллерами для загрузки и отладки приложений, передачи файлов и выполнения диагностических операций
NTP	Network Time Protocol, протокол сетевого времени для синхронизации внутренних часов на узлах сети с переменной латентностью
OPC UA	Open Platform Communications Unified Architecture – спецификация протокола информационного обмена между устройствами в промышленных сетях, разработанная организацией OPC Foundation
POU	Program Organization Unit, программная единица или программный компонент по ГОСТ Р МЭК 61131-3 – элемент программной модели, представляющий обособленный именованный набор данных и операций по их преобразованию. Стандартом определены четыре вида POU: программа (PROGRAM), функциональный блок (FUNCTION_BLOCK), функция (FUNCTION) и класс (CLASS).
PTP	Precision Time Protocol, протокол точного времени для синхронизации внутренних часов на узлах сети
ST	Structured Text, язык структурированного текста – текстовый язык реализации приложений ПЛК согласно IEC 61131-3 (ГОСТ Р МЭК 61131-3)
LD	Ladder Diagram, язык релейно-контактных схем (диаграмм) – графический язык реализации приложений ПЛК согласно IEC 61131-3 (ГОСТ Р МЭК 61131-3), основными примитивами которого являются графические представления катушек и контактов реле.
FBD	Function Block Diagram, язык диаграмм функциональных блоков – графический язык реализации приложений ПЛК согласно IEC 61131-3 (ГОСТ Р МЭК 61131-3), основными примитивами которого являются графические представления функциональных блоков, имеющих входы, выходы, внутренние переменные состояния и, как минимум, функцию, вызываемую при вызове экземпляра блока, оперирующую значениями на входах, переменных состояния и формирующую значения выходов.
SFC	Sequential Function Chart, язык последовательных функциональных схем – графический язык реализации приложений ПЛК согласно IEC 61131-3 (ГОСТ Р

	МЭК 61131-3), основными примитивами которого являются графические представления шагов, действий, переходов и условий переходов.
SSL	Secure Sockets Layer, уровень защищенных сокетов – защищенный криптографический протокол с асимметричной криптографией для аутентификации ключей обмена и симметричным шифрованием.
АСУТП, АСУ ТП	автоматизированная система управления технологическим процессом
КП	контроллер программируемый универсальный – вычислительное устройство, входящее в состав программируемого (логического) контроллера с переменным составом модулей и выполняющее функции модуля центрального процессора
ПЛК, контроллер	совокупность контроллера программируемого, объединенного с периферийными модулями межмодульной шиной, образующая многофункциональный многоканальный программируемый контроллер с переменным составом модулей, который по структуре и назначению относится к программируемым контроллерам по ГОСТ Р МЭК 61131-1
ПК	персональный компьютер, портативный компьютер (ноутбук), рабочая станция
ПТК	комплекс программно-технический – изделия, не соединенные друг с другом при изготовлении, и программное обеспечение, поставляемые потребителю и относящиеся к семейству продуктов, объединяемых одним или несколькими конструктивно-техническими признаками, принципами работы и правилами применения
ППО, приложение	прикладное программное обеспечение, реализующее специфические пользовательские алгоритмы сбора, обработки данных и управления технологическим объектом управления
РЭ	руководство по эксплуатации
Руководство пользователя Fastwel PLC Application Toolkit	документ <i>ИМЕС.00390-03 33 01. Пакет инструментальных средств ПЛК Fastwel. Fastwel PLC Application Toolkit. Руководство пользователя</i>
СПО	системное программное обеспечение, микропрограмма – встроенное программное обеспечение вычислительного или периферийного устройства
Система исполнения приложений МЭК 61131-3	часть СПО контроллеров Fastwel, обеспечивающая выполнение кода ППО, загруженного в ПЛК, взаимодействие ППО с периферийными модулями ПЛК и обмен данными по сети с использованием поддерживаемых протоколов прикладного уровня

1.4. CIDR-нотация для представления IP-адресов

В настоящем документе используется [CIDR-нотация](#) адреса узла IP-сети: X.Y.Z.Q/L, где X, Y, Z, Q – значения октетов IPv4-адреса в десятичном представлении, а L – длина маски подсети, равная количеству установленных в 1 битовых полей в маске подсети, если считать от крайней левой позиции значения маски.

Например, *10.0.0.100/8* означает IP-адрес 10.0.0.100 и маску подсети длиной 8 бит, т.е. *255.0.0.0* или *2#11111111_00000000_00000000_00000000*.

Для маски *255.255.252.0* длина маски равна 22 (*2#11111111_11111111_11111100_00000000*).

2. Общие сведения

2.1. Назначение Fastwel I/O-2

Fastwel I/O-2 является программно-техническим комплексом, предназначенным для создания автоматизированных систем сбора данных и управления.

Аппаратно-программные средства Fastwel I/O-2 могут использоваться для построения как автономных программируемых контроллеров, так и распределенных систем сбора данных и управления.

2.2. Назначение CPM810-03

CPM810-03 предназначен для применения в составе программируемых логических контроллеров с переменным составом модулей Fastwel I/O и/или Fastwel I/O-2 в качестве модуля центрального процессора.

Контроллер содержит встроенную систему исполнения прикладных программ (приложений), разрабатываемых в IDE МЭК 61131-3, обеспечивает обмен данными по сети Ethernet с использованием протоколов OPC UA, MODBUS TCP (мастер и подчиненный), через интерфейсы RS-232C и RS-485 с использованием протоколов MODBUS RTU или ASCII (мастер или подчиненный), управление модулями ввода-вывода Fastwel I/O, подключенными к межмодульной шине FBUS контроллера, а также, при наличии подключенных видеомонитора и клавиатуры, отображение графических мнемосхем и взаимодействие с оператором средствами встроенной подсистемы целевой визуализации.

Внешний вид контроллера показан на рисунке 1.

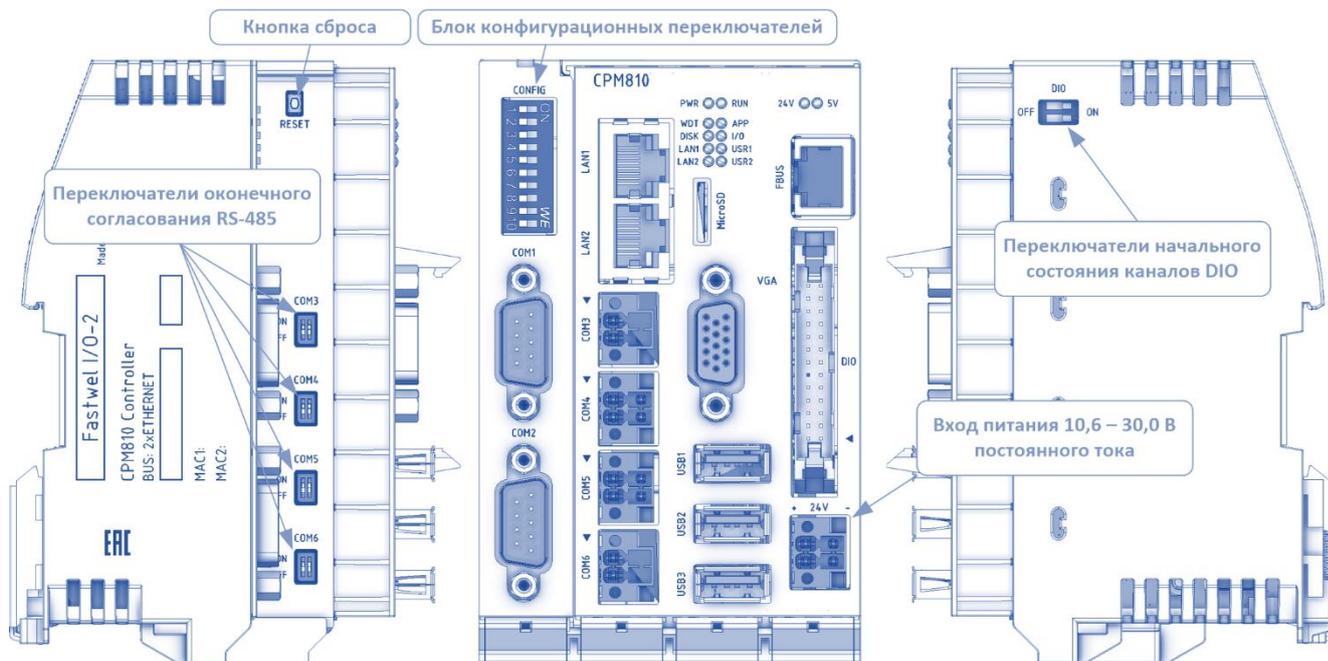


Рисунок 1 – Внешний вид CPM810-03

2.3. Структура аппаратных средств

2.3.1. Микропроцессор

CPM810-03 является вычислительным устройством на базе микропроцессора Vortex86DX3 с архитектурой команд x86 (i486) и тактовой частотой 800 МГц.

2.3.2. Межмодульная шина

CPM810-03 имеет интерфейс с периферийными модулями Fastwel I/O и Fastwel I/O-2, далее называемый внутренней шиной FBUS.

Для подключения набора периферийных модулей Fastwel I/O и Fastwel I/O-2 к СРМ810-03 через модули расширения шины FBUS OM757 или OM857 соответственно может использоваться розетка RJ-45 "FBUS" на передней панели.

Для непосредственного подключения набора периферийных модулей Fastwel I/O-2 к СРМ810-03 должен использоваться тыльный соединитель совместно с соединителем шины, входящим в комплект поставки СРМ810-03, как показано на рисунке 2.

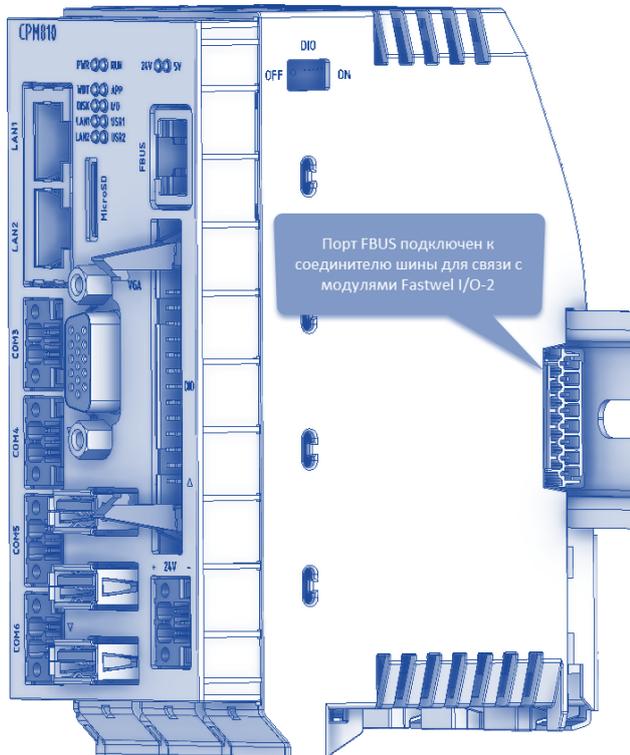


Рисунок 2 – Порт FBUS для связи с модулями Fastwel I/O-2 (присоединен к соединителю шины)

Встроенный submodule питания имеет соединитель "24V" и преобразует входное напряжение 24 В постоянного тока в 5 В для электропитания цифровой части контроллера и периферийных модулей Fastwel I/O-2, подключенных к тыльному соединителю FBUS.

Обратите внимание, что периферийные модуля Fastwel I/O или Fastwel I/O-2 должны подключаться к одному, и только одному, соединителю шины FBUS.

ВНИМАНИЕ!

Не допускается одновременное подключение двух наборов периферийных модулей Fastwel I/O и/или Fastwel I/O-2 к фронтальному и тыльному портам FBUS.

2.3.3. Порты Ethernet

Два порта интерфейса Ethernet LAN1 и LAN2 в составе контроллера реализованы на базе двух независимых контроллеров с поддержкой обмена на скоростях 10/100/1000 Мбит/с и предназначены для обмена данными с другими контроллерами, с рабочими станциями и автоматизированными рабочими местами верхнего уровня автоматизированных систем сбора данных и управления.

2.3.4. Порты RS-232C

Контроллер имеет в своем составе два порта интерфейса RS-232C COM1 и COM2, которые могут использоваться для обмена данными между приложением и другими устройствами по протоколу MODBUS RTU или ASCII в режиме мастера или подчиненного узла, либо с использованием функций системной библиотеки SysCom.

Один из портов может быть настроен на взаимодействие с IDE МЭК 61131-3 в веб-конфигураторе на странице **Маршрутизация CODESYS**. При поставке порт COM2 настроен на взаимодействие со средой разработки IDE МЭК 61131-3 с параметрами 115200-8-n-1.

2.3.5. Порты RS-485

Четыре порта RS-485 (COM3 – COM6) предназначены для информационного обмена между приложением CODESYS V3, загруженным в контроллер, и другими устройствами по протоколу MODBUS RTU или ASCII в режиме мастера или подчиненного узла, либо с использованием функций системной библиотеки SysCom.

2.3.6. Порты USB

Порты USB1 – USB3 с поддержкой спецификаций USB 1.1 и USB 2.0 предназначены для подключения к контроллеру клавиатуры, мыши или дисковых накопителей.

2.3.7. Порт дискретного ввода-вывода

На передней панели имеется 26-контактный соединитель "DIO", содержащий 24 канала дискретного ввода-вывода. Каналы могут использоваться для ввода или вывода дискретных сигналов с уровнями, совместимыми с КМОП-логикой.

Для доступа к каналам дискретного ввода-вывода из приложения IDE МЭК 61131-3, загружаемого в CPM810-03, должна использоваться библиотека FastweldIO.

2.3.8. Порт VGA

CPM810-03 имеет в своем составе порт VGA встроенной видеоподсистемы, который может использоваться для подключения видеомонитора с разрешением до 1920 × 1440/60 Гц/32 бит.

Включение и настройка порта осуществляются в BIOS и в веб-конфигураторе на странице **Видео**, более подробная информация приведена в п. 5.12.

2.3.9. Дисковые накопители

В состав CPM810-03 входит встроенный NAND флэш-накопитель (SATA NAND Flash) и гнездо для установки карты microSD (SDSC, SDHC, Spec. v1.1, v2.0) размером до 32 Гбайт с файловой системой FAT16/FAT32.

ВНИМАНИЕ!

Не допускается установка или извлечение карты microSD без выключения питания контроллера.

Встроенный флэш-накопитель диск является загрузочным, содержит системное программное обеспечение контроллера, и может использоваться приложением для сохранения данных средствами библиотеки SysFile, входящей в IDE МЭК 61131-3. Объем пространства на встроенном NAND флэш-диске, доступного приложению, составляет не менее 6 Гбайт.

Порты USB1 – USB3 могут использоваться для подключения и отключения дисковых накопителей без выключения питания контроллера.

2.3.10. Энергонезависимая память

Контроллер имеет в своем составе энергонезависимую память с линейным доступом размером 65512 байт, реализованную по технологии MRAM и не требующую питания от батареи при выключении или пропадании основного питания.

Данная память доступна приложению в виде сегмента энергонезависимых (RETAIN) переменных, которые сохраняют свои значения после выключения питания контроллера.

2.3.11. Часы-календарь

В состав контроллера входят часы-календарь с питанием от встроенной литиевой батареи. Питание от батареи используется только при отсутствии основного питания контроллера.

Погрешность хода часов при нормальных условиях составляет не хуже ± 2 с/сутки.

Доступ к системному времени осуществляется средствами библиотек SysTime, входящих в IDE МЭК 61131-3.

2.3.12. Органы управления и индикации

Контроллер имеет в своем составе 10 переключателей, предназначенных для выбора режима работы и настройки параметров контроллера. Первые восемь переключателей (1–8) предназначены для настройки IP-параметров сетевых интерфейсов контроллера, а переключатели 9 и 10 могут использоваться приложением при помощи функций библиотеки FastwelBoard, входящей в состав Fastwel PLC Application Toolkit. Информация о назначении переключателей приведена в пп. 4.2.1 и 4.6.2 настоящего документа.

Двенадцать светодиодных индикаторов в составе контроллера служат для отображения режимов работы его основных подсистем. Информация о назначении светодиодов приведена в пп. 4.2.2 и 4.2.3 настоящего документа. Двухцветные индикаторы USR1 и USR2 могут использоваться приложением при помощи функций библиотеки FastwelBoard. Информация о библиотеке FastwelBoard приведена в руководстве пользователя Fastwel PLC Application Toolkit.

2.4. Структура программного обеспечения

2.4.1. Состав программного обеспечения

Программные средства СРМ810-03 состоят из следующего системного и инструментального программного обеспечения:

1. Пакет инструментальных средств ПЛК Fastwel (далее *Fastwel PLC Application Toolkit*).
2. Системное программное обеспечение контроллера СРМ723-01.

Структура программного обеспечения СРМ810-03 показана на рисунке 3.

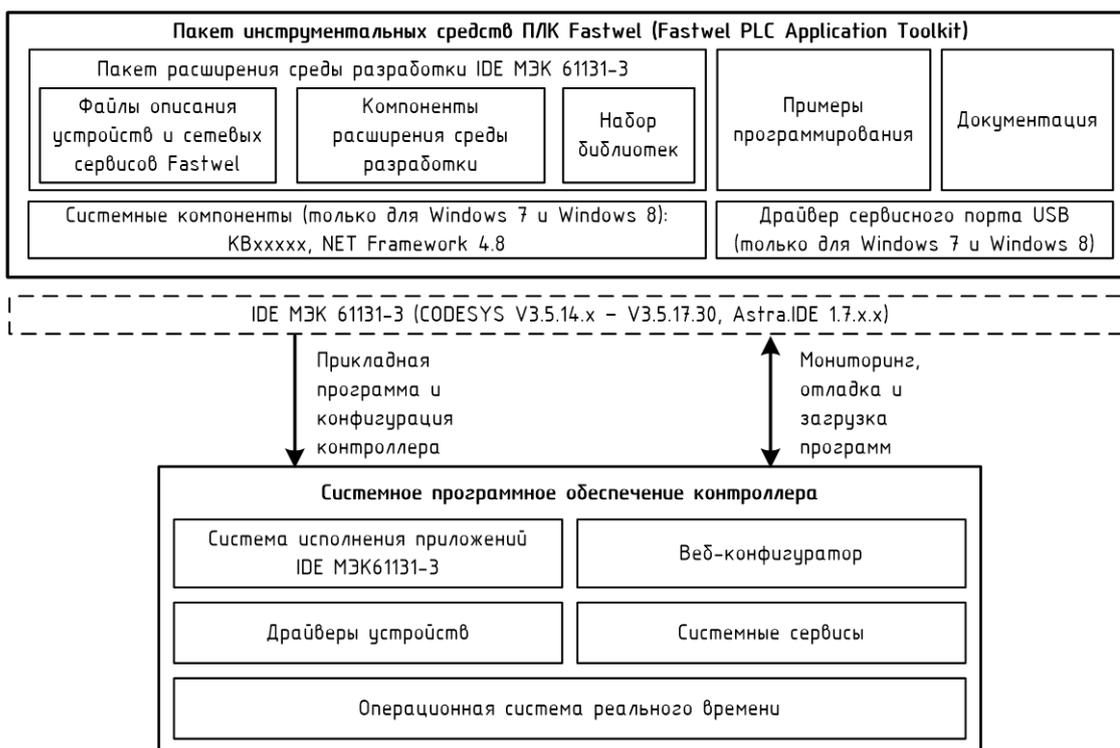


Рисунок 3 – Структура программного обеспечения СРМ810-03

2.4.2. Пакет инструментальных средств ПЛК Fastwel (Fastwel PLC Application Toolkit)

2.4.2.1. Назначение

Fastwel PLC Application Toolkit является программным обеспечением, устанавливаемым на ПК, интегрируемым с одной или несколькими совместимыми IDE МЭК 61131-3 и обеспечивающим возможность разработки приложений сбора, обработки данных и управления для контроллеров, выпускаемых под торговой маркой Fastwel, в том числе:

- Создание и редактирование проектов приложений для контроллеров Fastwel на базе процессоров с архитектурой x86 и ARM.

- Создание и редактирование конфигурации периферийных модулей, которые предполагается иметь в составе в контроллера, в том числе типы модулей, местоположение модулей на межмодульной шине, а также настройку параметров модулей в специальных редакторах, интегрированных в IDE МЭК 61131-3, в удобной для восприятия форме.
- Создание и редактирование конфигурации сетевых сервисов, которые должны функционировать в контроллере для обмена данными реального времени с другими контроллерами и/или системами сбора данных и оперативного диспетчерского управления и другими системами верхнего уровня, а также настройку параметров коммуникационных.

2.4.2.2. Варианты программы

Fastwel PLC Application Toolkit поставляется в виде двух вариантов программы установки (инсталлятора) на ПК с операционной системой MS Windows, в том числе: Windows 7 Service Pack 1, Windows 8, Windows 8.1, Windows 10 или Windows 11:

1. Базовый: *FastwelPLCApplicationToolkit.exe* – предполагает наличие на ПК ранее установленной совместимой IDE МЭК 61131-3 и содержит только программные компоненты, документацию и примеры программирования для контроллеров Fastwel. Совместимыми IDE МЭК 61131-3 являются:
CODESYS версий от V3.5 SP14 Patch 1 до V3.5 SP17 Patch 3;
Astra.IDE версии 1.7.x.x.
2. Расширенный: *FastwelPLCApplicationToolkitFull.exe* – помимо программных компонентов, документации и примеров программирования для контроллеров Fastwel, включает в себя IDE МЭК 61131-3 CODESYS версии не ниже V3.5 SP16 Patch 7 (3.5.16.700).

Обе программы установки поставляются на компакт-диске Fastwel DVD или могут быть загружены по адресу ftp://ftp.fastwel.ru/pub/Hardware/Fastwel/Fastwel_IO/Version3/Setup.

2.4.2.3. Состав

В состав обоих вариантов Fastwel PLC Application Toolkit входят следующие компоненты:

1. Системные компоненты: обновления операционной системы Windows 7 Service Pack 1, необходимые для установки .NET Framework v4.8, а также автономный инсталлятор .NET Framework v4.8, который должен быть установлен перед установкой IDE МЭК 61131-3 на ПК с операционными системами Windows 7 Service Pack 1, Windows 8/8.1.
Установка системных компонентов происходит автоматически только при необходимости в процессе работы инсталлятора Fastwel PLC Application Toolkit на ПК с операционными системами Windows 7 Service Pack 1 или Windows 8/8.1.
При установке обновлений Windows 7 Service Pack 1 может потребоваться перезапуск ПК, после которого работа инсталлятора Fastwel PLC Application Toolkit будет продолжена.
Лицензионное соглашение .NET Framework v4.8 предполагает наличие у пользователя корректной активной лицензии на используемую операционную систему Windows.
2. Драйвер сервисного порта USB: файл установки конфигурации драйвера Windows последовательного порта, реализуемого через интерфейс USB ПК, к которому подключен сервисный порт USB контроллера устанавливается только в операционных системах Windows 7 Service Pack 1 и Windows 8/8.1.
В операционных системах Windows 10 и Windows 11 используется драйвер CDC со стандартной конфигурацией.
3. Пакет расширения IDE МЭК 61131-3 *FastwelTargets.package*, в состав которого входят:
 - файлы описания целевых устройств, устройств ввода-вывода и сетевых сервисов, представляющих контроллеры Fastwel в IDE МЭК 61131-3 для генерации исполняемого кода целевых процессоров и бинарной конфигурации устройств ввода-вывода и сетевых сервисов;
 - набор библиотек, содержащих функциональные блоки, функции и другие программные единицы, обеспечивающие доступ к специфическим функциональным возможностям контроллеров Fastwel из приложений;

- наборы системных библиотек, обеспечивающих совместимость системных вызовов, формируемых IDE МЭК 61131-3 при компиляции и кодогенерации пользовательского приложения, с соответствующими реализациями системных вызовов в системе исполнения приложений в контроллерах Fastwel;
- компоненты расширения (плагины) IDE МЭК 61131-3, позволяющие создавать и редактировать в специальных интерактивных редакторах конфигурационную информацию для периферийных модулей и сетевых сервисов в составе контроллера, а также генерировать дополнительную информацию времени выполнения.

Данный пакет расширения устанавливается автоматически в выбранные пользователем профили IDE МЭК 61131-3 в процессе работы инсталлятора Fastwel PLC Application Toolkit.

4. Примеры программирования: проекты IDE МЭК 61131-3 с примерами программирования, демонстрирующими приемы программного доступа к подсистемам контроллера, управления ими, обработки данных периферийных модулей, диагностики и оценки вычислительной загрузки контроллера.

Для поставляемых примеров программирования в меню запуска Windows создаются ярлыки, облегчающие поиск и доступ к проектам.

Данные примеры программирования могут использоваться в качестве основы при реализации пользовательских приложений.

5. Документация: файлы эксплуатационной и программной документации для контроллеров Fastwel в формате pdf. Для документации в меню запуска Windows создаются ярлыки, облегчающие доступ к файлам.

Более подробная информация о составе Fastwel PLC Application Toolkit приведена в руководстве пользователя Fastwel PLC Application Toolkit.

2.4.3. Системное программное обеспечение контроллера

Системное программное обеспечение контроллера включает в себя систему исполнения приложений МЭК 61131-3 и системные сервисы, функционирующие под управлением операционной системы реального времени.

Система исполнения приложений МЭК 61131-3 выполняет следующие функции:

1. Исполнение приложения, загруженного пользователем из IDE МЭК 61131-3.
2. Обмен данными между приложением и модулями ввода-вывода.
3. Прием данных и команд по сети и передачу их приложению.
4. Передачу по сети данных приложения.
5. Отображение графических мнемосхем оператора, входящих в приложение, на видеомониторе, подключенном к контроллеру, и прием и обработку информации, вводимой оператором на подключенной клавиатуре и с помощью указательного устройства.
6. Светодиодную индикацию режима работы контроллера и его отдельных подсистем.
7. Диагностику функционирования основных подсистем системы исполнения и предоставление диагностической информации приложению.
8. Управление режимами работы контроллера, обработку ошибок и нестандартных ситуаций.

В состав системных сервисов, в том числе, входят:

1. Веб-конфигуратор – облегченный веб-сервер, обеспечивающий возможность удаленной настройки системных параметров контроллера при помощи веб-браузера.
2. Сервис сторожевого таймера – взаимодействует с аппаратным сторожевым таймером, входящим в состав контроллера.
3. Сервисы протоколов NTP и RTP – реализуют функции установки и синхронизации часов с использованием протоколов NTP и IEEE 1588 (RTPv2).
4. Сервис обновления системного программного обеспечения – служит для обновления системного программного обеспечения контроллера и модулей ввода-вывода.

2.4.4. Основные характеристики системного программного обеспечения

Перечень основных характеристик программного обеспечения контроллера приведен в таблице 1.

Таблица 1 – Основные характеристики системного программного обеспечения СРМ810-03

Характеристика	Значение
Параметры системы исполнения приложений	
Среда разработки приложений	Astra.IDE (версия 1.7.0.0 и выше), CODESYS V3 (версия 3.5.14.100 и выше)
Размер области памяти входных переменных приложения, байт, не более	524288
Размер области памяти выходных переменных приложения, байт, не более	524288
Размер области памяти внутренних переменных приложения, байт, не более	16252928
Размер области памяти исполняемого кода приложения, байт, не более	16777216
Размер области памяти энергонезависимых переменных, байт, не более	65512
Количество циклических задач, не более	16
Количество ациклических задач, не более	64
Период циклической задачи, мс	1 – 65530
Количество уровней приоритета циклических задач	32
Количество уровней приоритета жесткого реального времени	8
Количество уровней нормального приоритета	8
Количество уровней фонового приоритета	16
Специальные функции системы исполнения приложений	
Целевая визуализация	√
Поддержка встроенного сервера OPC UA	√
Количество сеансов с клиентами OPC UA, не более	32
Файловая система сменных карт microSD и USB-накопителей	FAT16, FAT32
Управление доступом	√
Механизм обмена сетевыми переменными CODESYS V3	√
Мониторинг переменных	√
Пошаговая отладка	√
Горячее обновление приложения (Online Change)	√
Просмотр стека вызовов при отладке	√
Встроенная трассировка значений	√
Браузер ПЛК	√
Журнал ПЛК	√
Подсистема ввода-вывода	
Количество локальных шин FBUS	1
Количество удаленных шин FBUS (через NIM745-01), не более	2
Количество модулей ввода-вывода на одну шину FBUS, не более	64
Период опроса модулей ввода-вывода по локальной шине, мс	1 – 1000
Период опроса модулей ввода-вывода по удаленной шине, мс	5 – 1000
Количество последовательных портов через коммуникационные модули Fastwel I/O и/или Fastwel I/O-2 RS-232C/RS-422/RS-485, не менее	32
Сетевая подсистема	
Поддерживаемые протоколы:	
MODBUS TCP (клиент и сервер)	√
MODBUS RTU/ASCII (мастер и подчиненный узлы)	√
ГОСТ Р МЭК 60870-5-104 (ведущий и подчиненный узлы)	√
Сервер FTP	√ (чтение и запись файлов)
Сервер HTTP	√ (настройка системных параметров)
NTP v4 (клиент и сервер)	√
IEEE 1588v2 (PTP) (ведущий и подчиненный узлы)	√

Характеристика	Значение
Сервис сервера (подчиненного узла) MODBUS TCP:	
Количество независимых экземпляров сервера в конфигурации, не более	2
Количество одновременных с клиентами на один экземпляр сервера, не более	32
Размер данных, публикуемых одним экземпляром сервера, байт, не более	65534
Размер данных, получаемых одним экземпляром сервера, байт, не более	65534
Маршрутизация запросов MODBUS TCP в MODBUS RTU	√
Поддерживаемые типы запросов MODBUS	01, 02, 03, 04, 05, 06, 0Fh, 10h, 17h
Управление функционированием сервера из кода приложения	√
Сервис клиента (мастера) MODBUS TCP:	
Суммарный размер данных, получаемых клиентом от подчиненных узлов, Кбайт, не более	128
Поддерживаемые типы запросов MODBUS	01, 02, 03, 04, 05, 06, 0Fh, 10h, 17h
Управление функционированием сервера из кода приложения	√
Сервис подчиненного узла MODBUS RTU/ASCII:	
Скорость передачи данных, бит/с	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Количество независимых экземпляров сервера в конфигурации, не более	16
Размер данных, публикуемых одним экземпляром сервера, байт, не более	65534
Размер данных, получаемых одним экземпляром сервера, байт, не более	65534
Поддерживаемые типы запросов MODBUS	01, 02, 03, 04, 05, 06, 0Fh, 10h, 17h
Управление функционированием сервера из кода приложения	√
Сервис мастера (ведущего узла) MODBUS RTU/ASCII:	
Скорость передачи данных, бит/с	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200
Количество независимых экземпляров мастера в конфигурации, не более	16
Суммарный размер данных, получаемых клиентом от подчиненных узлов, Кбайт, не более	128
Поддерживаемые типы запросов MODBUS	01, 02, 03, 04, 05, 06, 0Fh, 10h, 17h
Управление функционированием сервера из кода приложения	√

Параметры производительности определены для операций, представленных выражениями вида:

$$Z := X \text{ op } Y,$$

где *op* – код операции: '+', '-', '*', '/' и выполняемых в контексте циклической задачи наивысшего приоритета с уровнем 0.

2.5. Системные требования к рабочему месту разработчика

Требования к аппаратным и программным средствам, а также указания по установке среды разработки IDE МЭК 61131-3 и Fastwel PLC Application Toolkit, настройке связи между средой разработки и контроллером приведены в руководстве пользователя Fastwel PLC Application Toolkit.

3. Ускоренное ознакомление

3.1. Общие сведения

Процесс разработки прикладного программного обеспечения для контроллера СРМ810-03 состоит из следующих основных операций:

1. Настройка системных сервисов контроллера, в том числе настройка параметров IP сетевых интерфейсов, сетевых сервисов синхронизации времени и т.д. при помощи переключателей и/или веб-браузера.
2. Создание проекта IDE МЭК 61131-3 для платформы *СРМ810-03 Programmable Automation Controller*.
3. Создание конфигурации модулей ввода-вывода.
4. Создание конфигурации сервисов внешней сети, включая описание параметров и коммуникационных объектов клиентов (мастеров) и серверов (подчиненных узлов) протокола MODBUS TCP и/или MODBUS RTU.
5. Разработка программных единиц (POU), реализующих требуемые прикладные алгоритмы, включая программы, функциональные блоки и функции.
6. Отображение (сопоставление) переменных приложения на входные и выходные каналы модулей ввода-вывода и коммуникационных объектов обмена данными сетевых сервисов.
7. Создание циклических, ациклических и свободно-исполняемых задач, под управлением которых будут исполняться программы.
8. При необходимости, создание функций обработки системных событий, происходящих в системе исполнения контроллера.
9. Трансляция (компиляция) приложения для получения исполняемого кода и бинарной конфигурации, подлежащих загрузке в контроллер.
10. Загрузка прикладной программы в контроллер.
11. Мониторинг переменных и отладка приложения в контроллере.

В данном разделе будут рассмотрены перечисленные операции на примере небольшого учебного проекта.

Для выполнения указаний настоящего раздела потребуются следующие оборудование, принадлежности и программное обеспечение:

1. Контроллер СРМ810-03 с исходными (заводскими) установками сетевых интерфейсов:
IP адрес LAN1 (база): 10.0.0.0/8
IP адрес LAN2 (база): 10.0.0.0/8
2. Кабель соединительный нуль-модемный ASC00092 или аналогичный.
3. Один или два кабеля TIA/EIA-568-B (патч-корд).
4. Неуправляемый или управляемый коммутатор сети Ethernet.
5. Блок питания с выходным напряжением 24 В постоянного тока.
6. Модуль реле DIM713.
7. ПК с веб-браузером, а также установленной IDE МЭК 61131-3 и Fastwel PLC Application Toolkit в соответствии с требованиями руководства пользователя Fastwel PLC Application Toolkit.

3.2. Подготовка к работе

3.2.1. Цель работы

В результате работы над проектом будет создано приложение для контроллера CPM810-03 со следующими функциями:

1. Прерывистое свечение индикатора USR1 контроллера красным или зеленым цветом. Красный цвет будет использоваться при отсутствии запросов к серверу MODBUS TCP от удаленных клиентов, а зеленый – при периодическом поступлении запросов к серверу.
2. Включение и выключение первого канала модуля реле DIM713 при включении и выключении 10-го переключателя.
3. Включение и выключение второго канала модуля реле DIM713 по команде, поступающей от удаленного клиента протокола MODBUS TCP.
4. Включение и выключение второго канала модуля реле DIM713 по сети MODBUS TCP из другой циклической задачи этого же приложения.

3.2.2. Конфигурация аппаратных средств для учебного проекта

1. Соберите схему, показанную на рисунке 4.
2. Если ПК подключен к сети с динамическим распределением адресов по протоколу DHCP, установите переключатели контроллера в положение 254 ("Динамическое назначение IP-адресов"), как показано на рисунке 5.

Если IP-параметры сетевого интерфейса ПК заданы статически, установите для ПК IP-адрес в подсети 10.0.0.0/8, например, 10.0.0.252, с маской подсети 255.0.0.0, а переключатели контроллера установите в положение 3, как показано на рисунке 5.

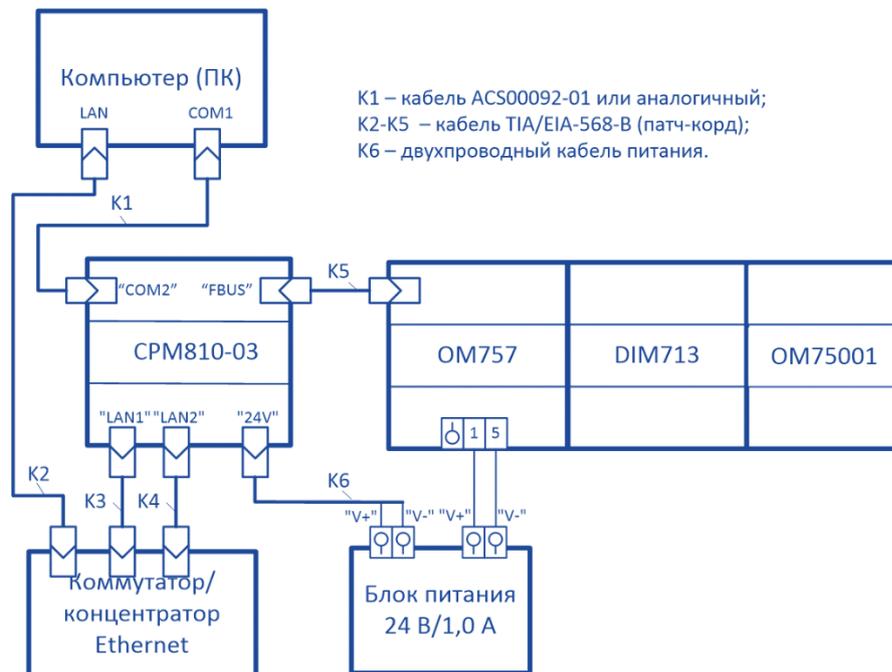


Рисунок 4 – Схема соединений контроллера для реализации учебного проекта

3. Присоедините вилки кабелей K3 и K4 к портам LAN1 и LAN2 контроллера.
4. Включите блок питания.

Если в контроллере отсутствует ранее загруженное приложение, через некоторое время индикатор RUN будет светиться прерывисто, поочередно меняя цвет с зеленого на красный и погасая, что свидетельствует об отсутствии в контроллере исполняющегося приложения и пребывании контроллера в безопасном (SAFE) режиме.

Индикаторы LAN1 и LAN2 при наличии физического соединения с коммутатором будут светиться непрерывно зеленым цветом.

При отсутствии в контроллере ранее загруженного приложения, остальные индикаторы контроллера будут выключены.

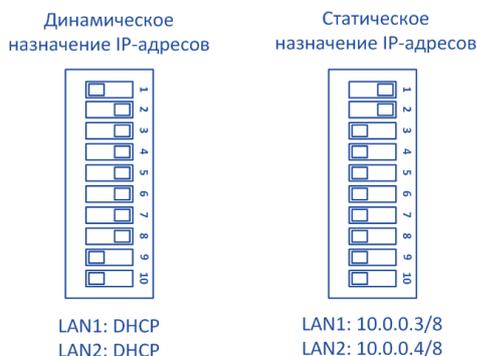


Рисунок 5 – Варианты установки IP-параметров контроллера с заводскими настройками

3.2.3. Сетевые параметры контроллера

Сетевые параметры контроллера представлены, как минимум, IP-адресами сетевых интерфейсов. При работе над учебным проектом потребуются знать IP-адреса сетевых интерфейсов контроллера для конфигурирования мастера (клиента) MODBUS TCP, который будет запускаться на этом же контроллере и опрашивать собственный сервер MODBUS TCP.

Если конфигурация контроллера ранее не менялась с момента извлечения из упаковки, то сетевые интерфейсы LAN1 и LAN2 будут иметь два разных физических (MAC) адреса и настроены для работы одной подсети. Если при этом переключатели установлены в положение 3, то порт LAN1 контроллера будет иметь IP-адрес 10.0.0.3/8, а порт LAN2 – 10.0.0.4/8. При установке переключателей в положение от 1 до 253 фактические IP-адреса сетевых интерфейсов контроллера определяются следующим образом:

IP-адрес LAN1 = базовый IP-адрес 1 + swv1_8

IP-адрес LAN2 = базовый IP-адрес 2 + swv1_8

где swv1_8 – значение, установленное на переключателях 1–8, из диапазона от 1 до 253 включительно;

базовый IP-адрес 1, базовый IP-адрес 2 – значения базового адреса интерфейсов LAN1 и LAN2, определяемые параметрами **IP адрес LAN1 (база)** и **IP адрес LAN2 (база)** на странице **Параметры сети** встроенного веб-конфигуратора контроллера.

Если базовые адреса обоих интерфейсов совпадают и относятся к одной подсети, то фактический IP-адрес второго сетевого интерфейса LAN2 определяется так:

IP-адрес LAN2 = базовый IP-адрес + swv1_8 + 1

По умолчанию значения базового адреса LAN1 и LAN2 равно 10.0.0.0/8, поэтому при установке переключателей в положение 3, сетевой интерфейс LAN1 будет иметь IP-адрес 10.0.0.3/8, а LAN2 – 10.0.0.4/8.

Если переключатели контроллера установлены в положение 254, то IP-адреса сетевых интерфейсов контроллера устанавливаются по протоколу DHCP. При этом рекомендуется задать на используемом в сети сервере DHCP фиксированное сопоставление между MAC-адресами портов LAN1 и LAN2 контроллера и IP-адресами, назначаемыми им сервером DHCP. MAC-адреса портов LAN1 и LAN2 нанесены на левую плоскость контроллера, как показано на рисунке 1.

Если нет возможности установить на сервере DHCP фиксированное сопоставление MAC-адресов портов LAN1 и LAN2 контроллера с назначаемыми IP-адресами, то по значениям MAC-адресов выясните на сервере DHCP, какие IP-адреса были назначены портам LAN1 и LAN2 контроллера.

Информация о настройке IP-параметров контроллера приведена в п. 4.6.2 настоящего документа.

IP-адреса сетевых портов контроллера могут быть выяснены при помощи команды *ipinfo*, выполняемой на вкладке **Оболочка ПЛК (PLC Shell)** редактора устройства (*CPM810-03 Programmable Automation Controller*) в IDE МЭК 61131-3:

1. Настройте параметры сервиса IDE Gateway и установите с контроллером связь через сервисный USB-порт в соответствии с указаниям руководства пользователя Fastwel PLC Application Toolkit.

- В редакторе устройства выберите вкладку **Оболочка ПЛК (PLC Shell)** и в поле ввода команд введите:

```
ipinfo
```

Если ранее связь с контроллером через сервисный порт была успешно установлена, на вкладке **Оболочка ПЛК** будет отображена информация о текущих IP-параметрах и режиме работы сетевых интерфейсов контроллера:

```
=>Adapters: 2
[0]: eth0, interfaces: 1
    [0]:LAN1
        MAC: 00:08:B3:02:9D:E1
        IPv4: 10.0.0.100
        Mask: 255.0.0.0
        Broadcast: 10.255.255.255
        Gateway: 255.255.255.255
[1]: eth1, interfaces: 1
    [0]:LAN2
        MAC: 00:08:B3:02:9D:E2
        IPv4: 10.0.0.101
        Mask: 255.0.0.0
        Broadcast: 10.255.255.255
        Gateway: 255.255.255.255
```

IP-параметры интерфейсов LAN1 и LAN2 – в строках [0]:LAN1 и [1]:LAN2.

3.3. Настройка параметров системных сервисов контроллера

Настройка параметров системных сервисов контроллера выполняется в веб-браузере через встроенный веб-сервер контроллера в соответствии с указаниями раздела 5 настоящего руководства. При разработке учебного проекта будут использованы исходные параметры системных сервисов контроллера, т.е. никаких дополнительных настроек через веб-браузер не потребуется.

3.4. Создание проекта

Для создания проекта запустите среду разработки IDE МЭК 61131-3 и выполните следующие действия:

- В главном меню выберите команду **Файл – Создать новый проект**. На экране будет отображено окно

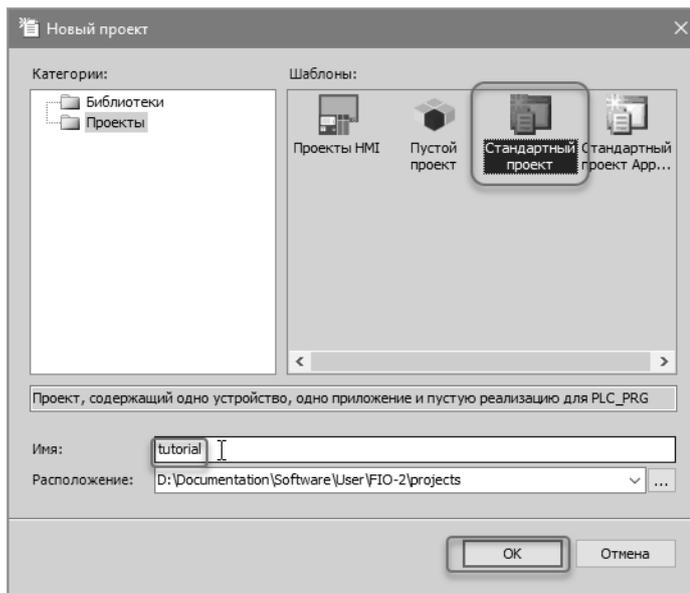


Рисунок 6 – Окно "Новый проект" при создании нового проекта

- В окне **Новый проект**, показанном на рисунке 6, выберите *Стандартный проект* в списке **Шаблоны**.
- В поле **Имя** введите имя файла проекта и нажмите **ОК**.

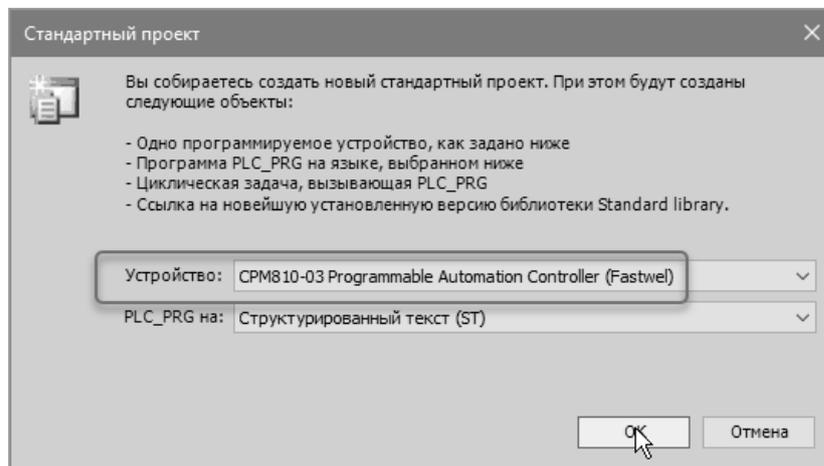


Рисунок 7 – Окно выбора целевой платформы для создаваемого проекта

4. В окно выбора целевой платформы **Стандартный проект**, показанном на рисунке 7, в выпадающем списке **Устройство** выберите *CPM810-03 Programmable Automation Controller (Fastwel)* и нажмите **ОК**.

На вкладке **Устройства**, расположенной в левой области главного окна CODESYS V3, будет отображено дерево проектной информации, относящейся к создаваемому проекту, как показано на рисунке 8.

5. Выполните команду **Компиляция – Генерировать код** главного меню CODESYS V3. В окне **Сообщения** при успешном построении исполняемого кода приложения будет отображено несколько сообщений, завершающихся строкой:
Компиляция завершена – 0 ошибок, 0 предупреждений: готово к загрузке
6. Сохраните проект, выполнив команду **Файл – Сохранить**, или нажатием сочетания клавиш **Ctrl-S**.

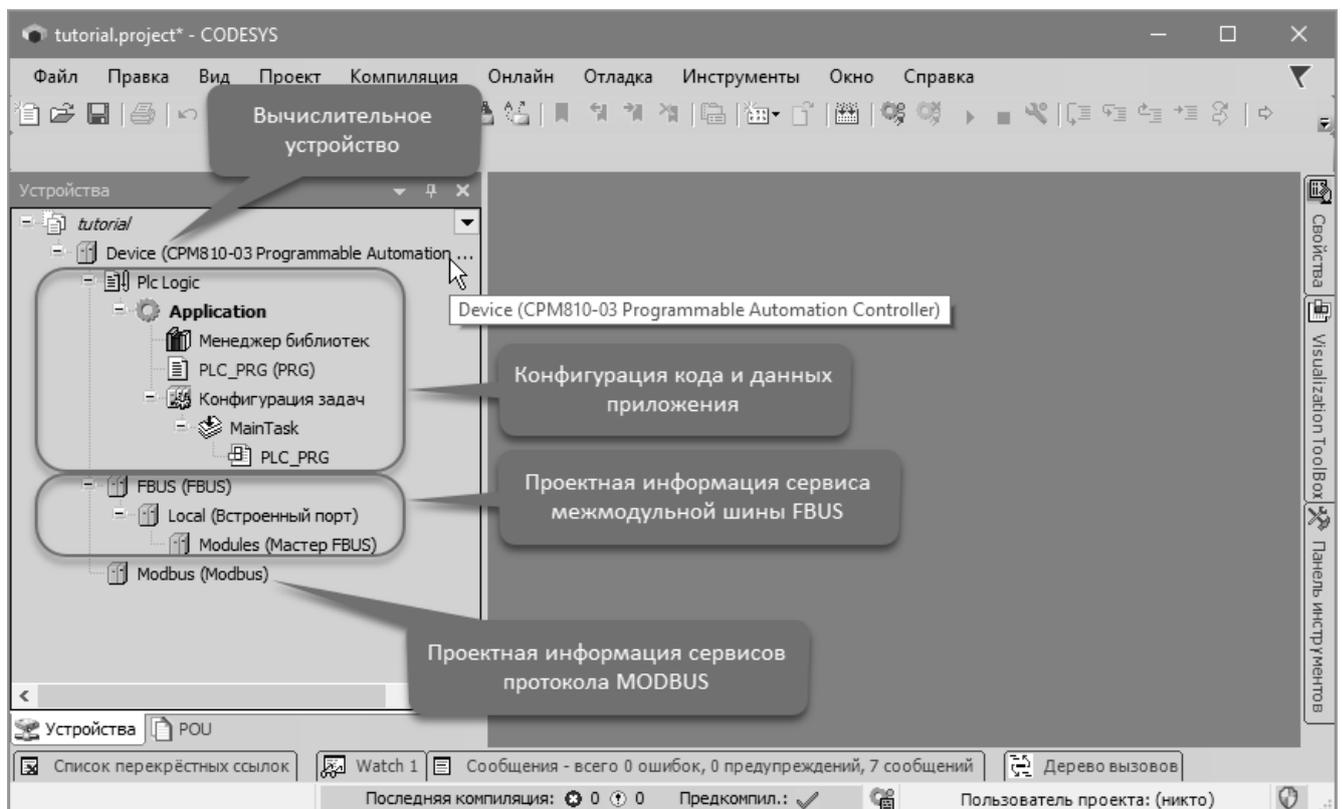


Рисунок 8 – Исходное представление стандартного проекта для CPM810-03

3.5. Создание конфигурации модулей ввода-вывода

3.5.1. Общие сведения

Контроллер СРМ810-03 обеспечивает возможность работы с тремя наборами модулей ввода-вывода Fastwel I/O и/или Fastwel I/O-2, каждый из которых может состоять из 64 активных модулей. Модули первого набора подключаются непосредственно к локальной шине FBUS контроллера, а два других набора – к модулям интерфейсным NIM745-01, доступным контроллеру по сети Ethernet. Модуль NIM745-01 является удаленным адаптером шины FBUS.

Описания модулей ввода-вывода, которые требуется иметь для работы приложения в составе контроллера на локальной шине, добавляются пользователем к элементу дерева проектной информации *FBUS – Local (Встроенный порт) – Modules (Мастер FBUS)*. Обратите внимание, что имена элементов дерева *Local* и *Modules* могут быть изменены в процессе работы над проектом.

В учебном проекте для работы приложения потребуется единственный модуль дискретного вывода DIM713, подключенный к локальной шине FBUS контроллера. Описание данного модуля может быть добавлено в проект вручную либо, если у среды разработки имеется соединение с включенным контроллером, к которому подключен требуемый модуль, путем сканирования шины и вставки описания в проект из списка найденных модулей.

3.5.2. Создание описания модулей ввода-вывода вручную

Для добавления описание модуля DIM713 в дерево проектной информации:

1. Щелкните правой кнопкой мыши над элементом *FBUS (FBUS) – Local (Встроенный порт) – Modules (Мастер FBUS)* и выберите команду **Добавить устройство** в контекстном меню. На экране будет отображено окно **Добавить устройство**, показанное на рисунке 9, со списком описаний типов периферийных модулей, которые могут быть вставлены в конфигурацию приложения.
Обратите внимание, что данное окно позволяет как добавлять описание модуля последним элементом списка (**Действие: Добавить устройство**), так и вставлять в любую позицию имеющегося списка модулей (**Действие: Вставить устройство**).
Столбец **Версия** содержит номер версии описательной информации об устройствах, установленной с пакетом адаптации среды разработки, которая используется для представления параметров и каналов ввода-вывода устройств в редакторах устройств IDE МЭК 61131-3. Обратите внимание, что данный номер версии не имеет ничего общего с версией системного программного обеспечения физических устройств.
2. В списке устройств выберите *DIM713 2-channel AC/DC Relay Output Module* и нажмите кнопку **Добавить устройство**. Описание устройства с именем *m_dim713* (из поля **Имя** в окне **Добавить устройство**) будет добавлено в дерево проекта под элементом *Modules (Мастер FBUS)*, как показано на рисунке 10.
Обратите внимание, что можно продолжить добавление модулей в проект, не закрывая окно **Добавить устройство**, а выбирая требуемые элементы в списке доступных типов модулей и нажимая **Добавить устройство**.
3. Закройте окно **Добавить устройство** нажатием кнопки **Заккрыть** и сохраните проект командой **Файл – Сохранить** в главном меню среды разработки.

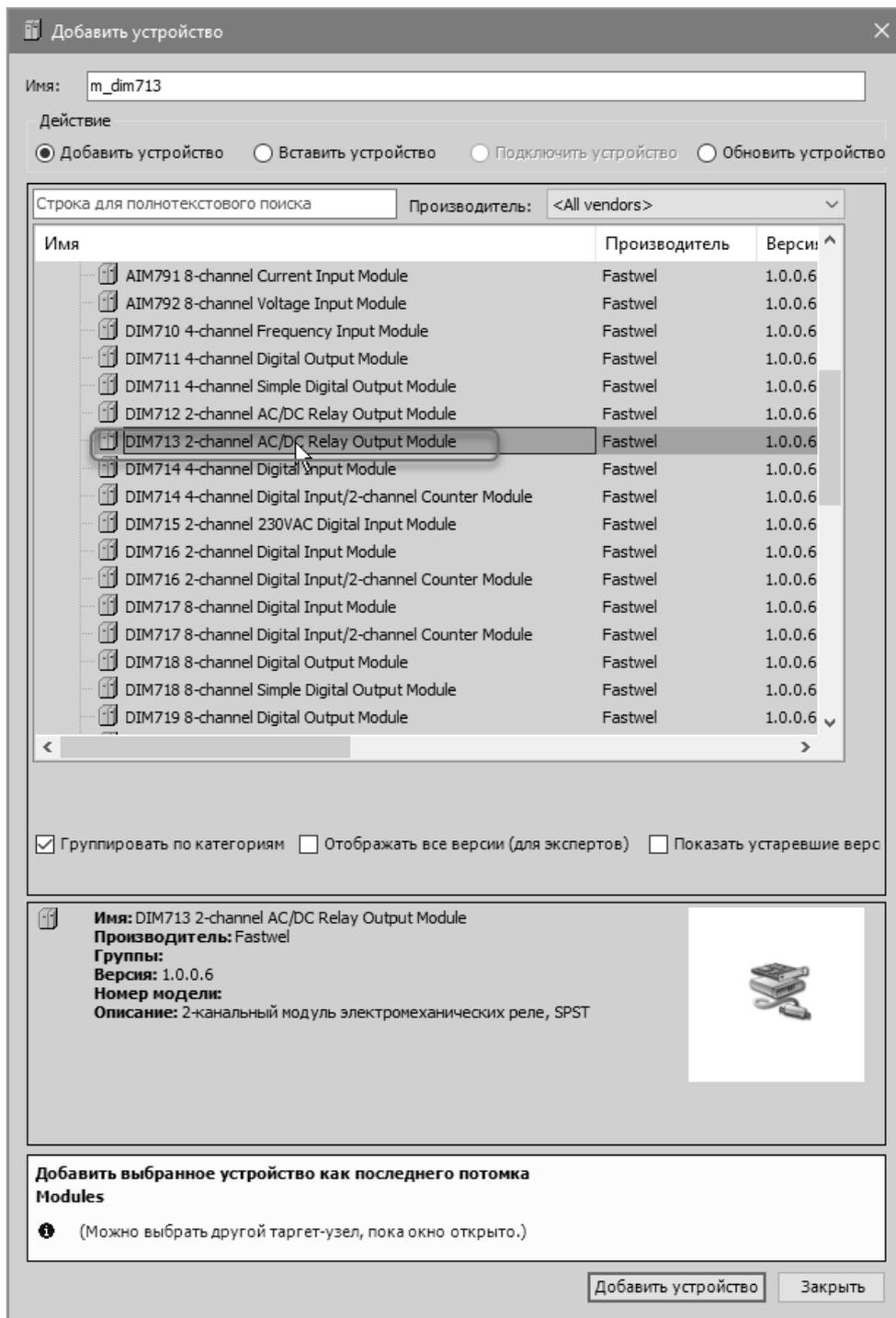


Рисунок 9 – Добавление описания модуля ввода-вывода вручную

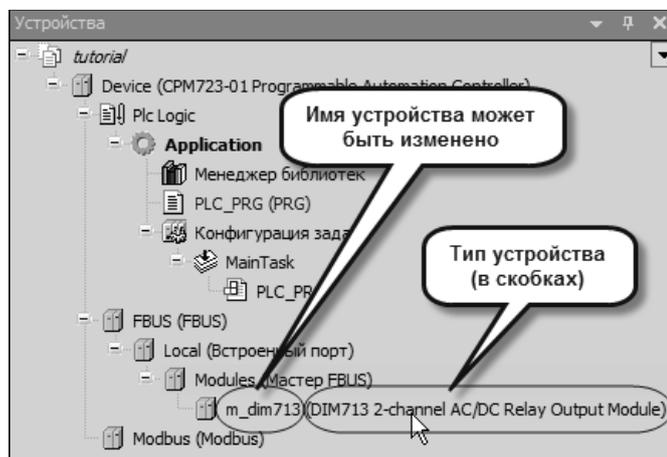


Рисунок 10 – Описание модуля DIM713 в дереве проекта

3.5.3. Создание описания модулей ввода-вывода из списка сканирования контроллера

Для добавления в проект описаний модулей ввода-вывода путем считывания текущей аппаратной конфигурации контроллера необходимо настроить путь связи между средой разработки и контроллером, после чего выбрать команду **Поиск устройств** в контекстном меню над элементом проектной информации *Modules (Мастер FBUS)*.

1. Включите питание контроллера, если это не было сделано ранее.
2. Настройте сетевой путь доступа среды разработки к контроллеру в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.
3. При необходимости загрузить пустой проект в контроллер выполните команду **Онлайн – Логин**. Если в контроллере отсутствует выполняющееся приложение или если приложение в контроллере отличается от открытого в среде разработки, на экран будет выведена диалоговая панель с сообщением о необходимости загрузки измененного приложения в контроллер или обновления приложения в контроллере. Нажмите **Да**, и приложение будет загружено в контроллер, после чего система исполнения приложений в контроллере будет ожидать команды запуска приложения **Отладка – Старт** или нажатием клавиши F5, как показано на рисунке 11.

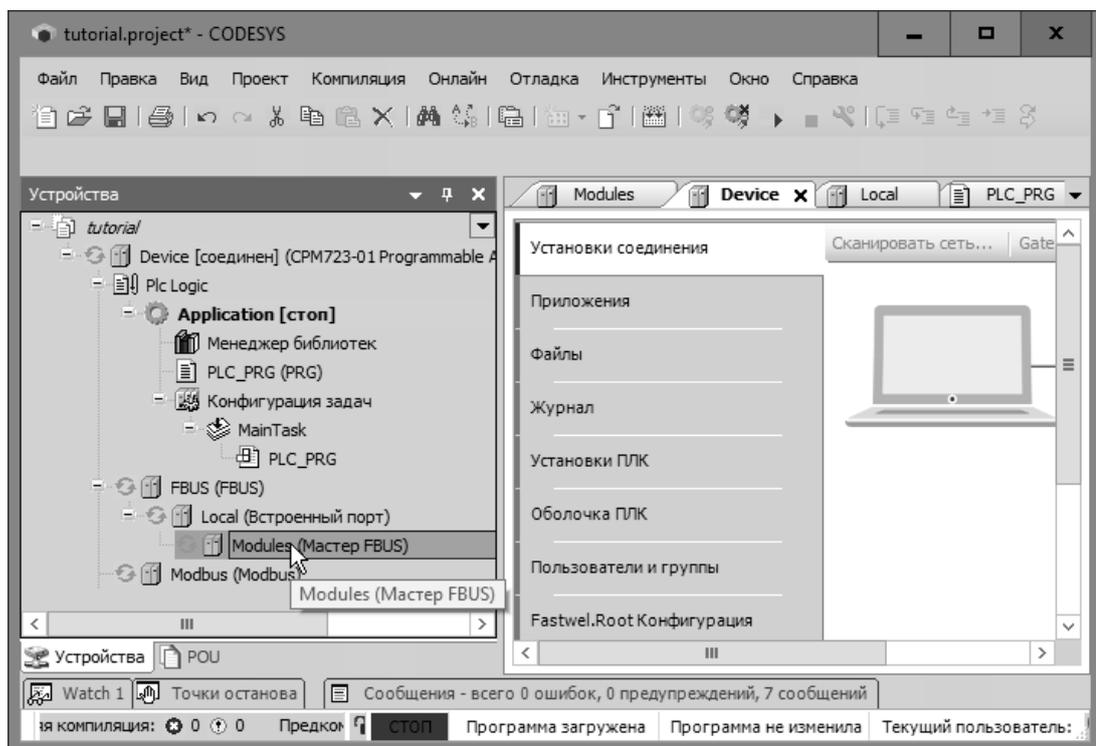


Рисунок 11 – Окно среды разработки после загрузки приложения

4. Щелкните правой кнопкой мыши над элементом *FBUS – Local (Встроенный порт) – Modules (Мастер FBUS)* и в появившемся контекстном меню выполните команду **Поиск устройств**. При наличии связи с контроллером на экран монитора будет выведено окно **Поиск устройств**, показанное на рисунке 12, содержащее описание модулей ввода-вывода, физически подключенных к локальной шине и обнаруженных сервисом ввода-вывода контроллера.
5. Если на ПК установлено несколько версий пакета расширения среды разработки для контроллеров Fastwel, то следует выбрать актуальную версию описания устройства, как показано на рисунке 12.
6. Выберите название модуля, подлежащего вставке в проект, и нажмите кнопку **Копировать в проект**. Окно **Поиск устройств** будет закрыто, а описание модуля появится в дереве проекта.
7. Если установлено соединение между средой разработки и контроллером, для загрузки измененного приложения в контроллер выполните команду **Онлайн – Отключение**, а затем **Онлайн – Логин**, после чего нажмите **Да** в диалоговой панели с предложением выполнить полную загрузку приложения в контроллер.

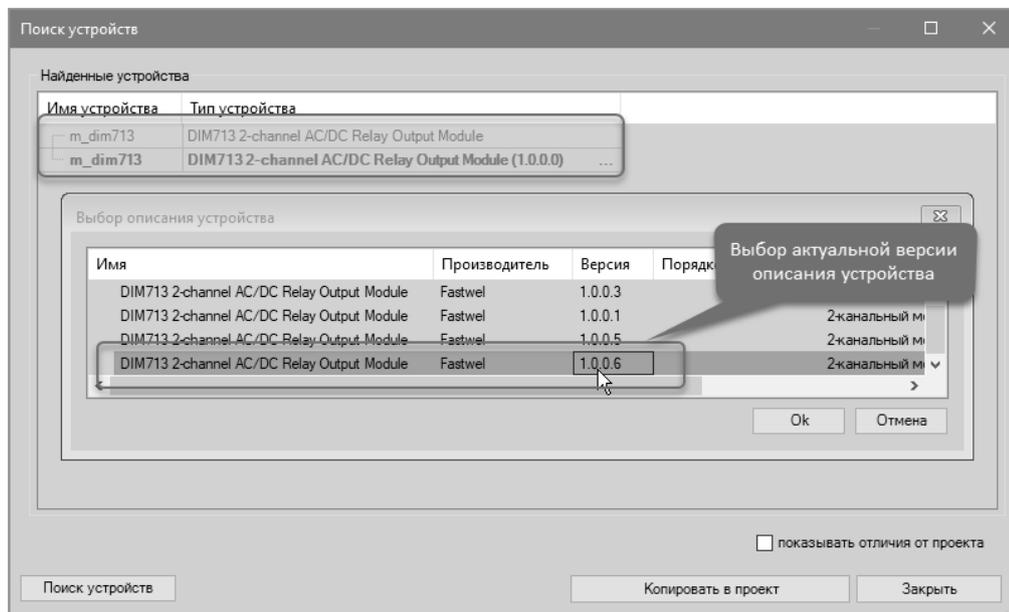


Рисунок 12 – Окно поиска устройств при успешном сканировании шины FBUS

- После успешной загрузки приложения в контроллера в среде разработки нажмите кнопку F5 или выполните команду **Отладка – Старт**. Если описания модулей ввода-вывода в загруженном приложении совпадают с фактической аппаратной конфигурацией модулей ввода-вывода, подключенных к локальной шине контроллера, статус исправности шины и модулей будет отображен символами круглой формы зеленого цвета, как показано на рисунке 13.

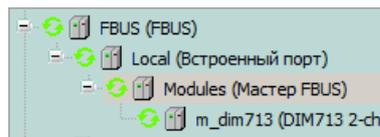


Рисунок 13. Отображение статуса исправности межмодульной шины и модуля ввода-вывода

Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш Ctrl-S. Обратите внимание, что среда разработки с исходными настройками не сохраняет проектную информацию в файле до тех пор, пока пользователь не выполнит команду сохранения вручную. Для изменения данного поведения включите функцию автоматического сохранения проекта путем выбора команды меню **Инструменты (Tools) – Опции (Options)** и установки опции **Автосохранение с интервалом (Automatically save every)** с требуемым интервалом автосохранения в категории параметров **Загрузка и сохранение (Load and Save)** в окне **Опции (Options)**.

- Более подробная информация приведена в руководстве пользователя Fastwel PLC Application Toolkit.

3.5.4. Настройка параметров межмодульной шины и модулей ввода-вывода

Настройка параметров модулей ввода-вывода выполняется в редакторе устройств на вкладке **Конфигурация**, которая отображается по двойному щелчку на описании модуля в дереве проекта, как показано на рисунке 14.

Режим работы и период опроса межмодульной шины устанавливаются редакторе соответствующего устройства на вкладке **Конфигурация**, отображаемой по двойному щелчку на элементе *Modules (Мастер FBUS)*. Внешний вид редактора конфигурации межмодульной шины показан на рисунке 15. Более подробная информация о параметрах межмодульной шины в п. 4.5.2 настоящего документа.

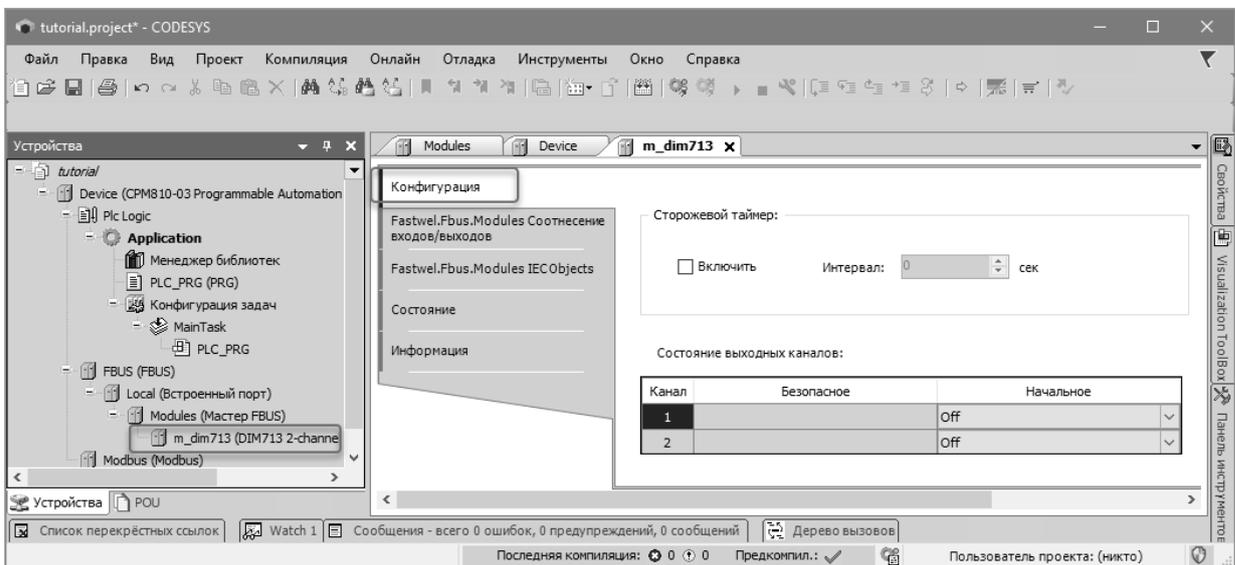


Рисунок 14 – Конфигурация модуля DIM713 в редакторе устройств

В рассматриваемом учебном проекте не требуется изменять исходные параметры межмодульной шины и модуля DIM713.

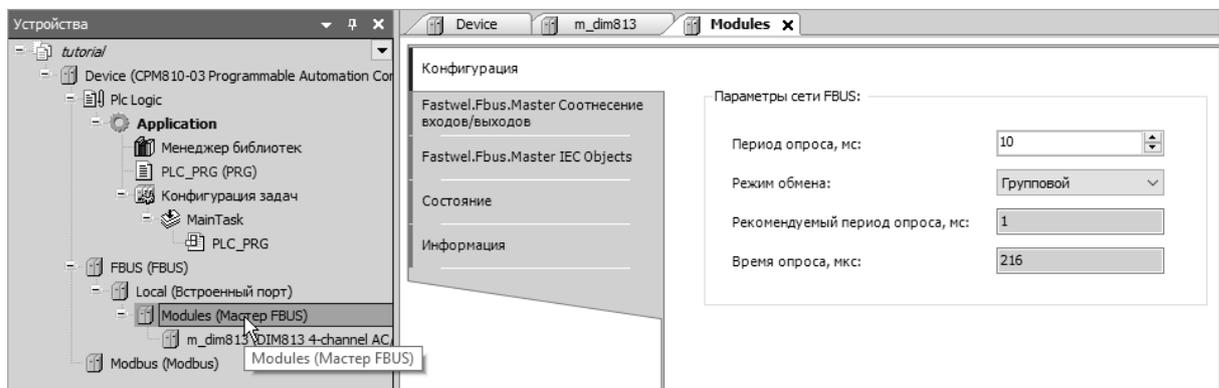


Рисунок 15 – Конфигурация межмодульной шины FBUS

3.5.5. Создание связей приложения с каналами модуля ввода-вывода

3.5.5.1. Общие сведения

Взаимодействие исполняемого кода приложения с каналами модулей ввода-вывода и с коммуникационными объектами сетевых сервисов осуществляется через сегменты входных и выходных данных приложения, называемых далее входным и выходным образом процесса.

IDE МЭК 61131-3 поддерживает несколько способов организации связи программных единиц приложения с каналами модулей ввода-вывода и коммуникационных объектов внешней сети, называемых *соотнесением* или *отображением*:

1. Декларация глобальных переменных или переменных программных единиц типа *PROGRAM*, непосредственно отображенных на адреса входного или выходного образа процесса. Данный способ соотнесения иллюстрируется рисунком 16.
2. Соотнесение имеющихся переменных программных единиц с адресами во входном или выходном образе процесса, как показано на рисунке 17. Двойной щелчок в ячейке **Переменная** в редакторе образа процесса приводит к появлению кнопки , позволяющей выбрать имеющуюся переменную, с которой требуется ассоциировать входной или выходной канал модуля ввода-вывода или коммуникационного объекта.
3. Создание переменных, отображенных на входные и выходные каналы модулей ввода-вывода и коммуникационных объектов, непосредственно в редакторе образа процесса, как показано на рисунке 18.
4. Соотнесение с адресами в образе процесса недоопределенных конфигурируемых переменных программных единиц в секции *VAR_CONFIG*. Данный способ иллюстрируется рисунком 19.

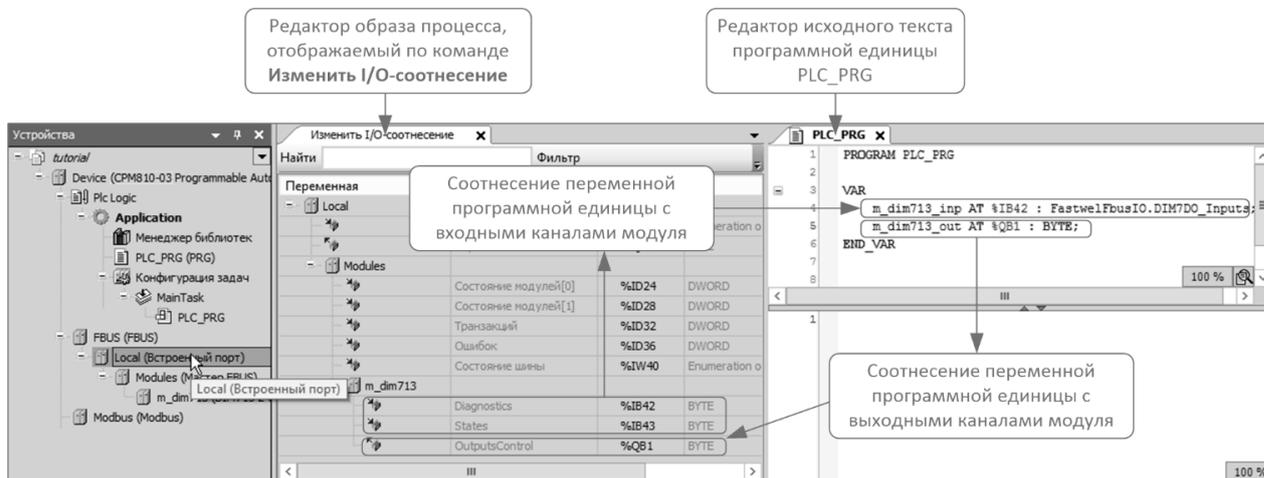


Рисунок 16 – Непосредственное соотнесение переменных PLC_PRG с адресами в образе процесса

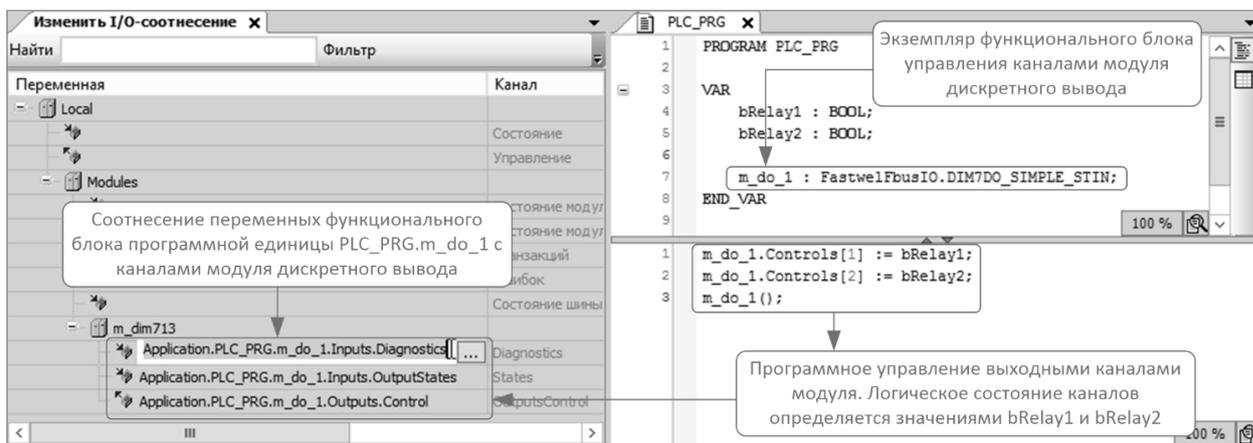


Рисунок 17 – Соотнесение имеющихся переменных экземпляра функционального блока с входными и выходными каналами модуля

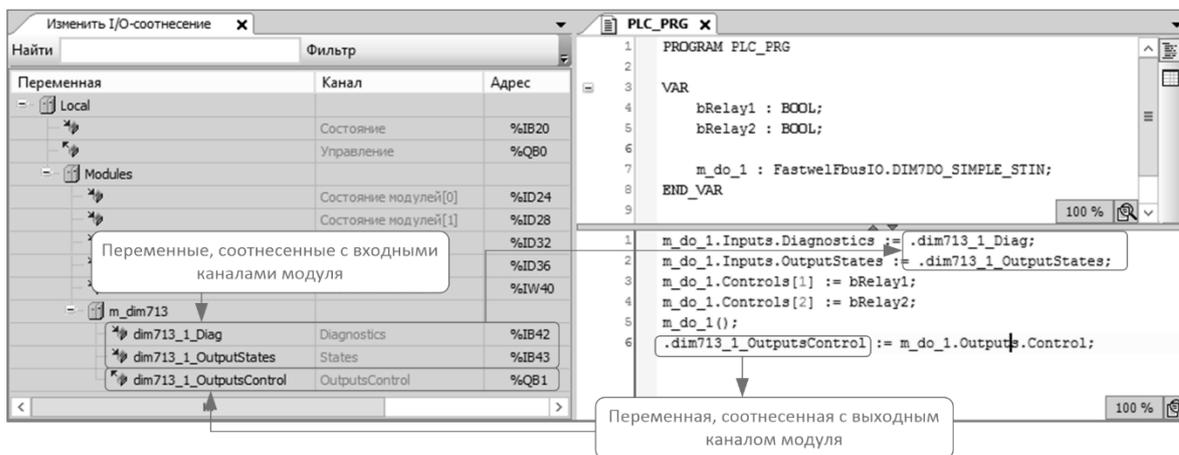


Рисунок 18 – Создание входных и выходных переменных в редакторе образа процесса

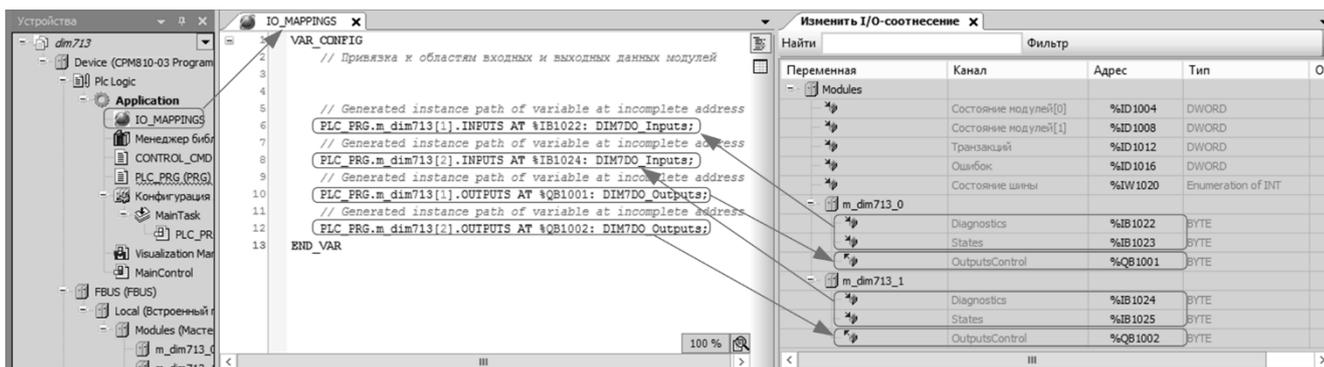


Рисунок 19 – Привязка недоопределенных конфигурируемых переменных в секции VAR_CONFIG

В рассматриваемом учебном проекте будет использован второй способ, при котором имеющиеся переменные приложения соотносятся с адресами каналов модулей ввода-вывода в образе процесса.

3.5.5.2. Соотнесение переменных функционального блока с каналами модуля ввода-вывода

Проект, созданный при выполнении указаний п. 3.4 настоящего документа, содержит пустую программную единицу языка ST (Structured Text) ГОСТ Р МЭК 61131-3 типа *PROGRAM* с именем *PLC_PRG*, которая, после загрузки приложения в контроллер, будет выполняться под управлением циклической задачи *MainTask*. *Выполняться под управлением циклической задачи* означает, что код *PLC_PRG* от начала и до конца программной единицы будет запускаться многократно с периодом, установленным в свойствах задачи *MainTask*.

Программа *PLC_PRG* будет содержать исходный текст приложения, реализующий простой алгоритм, описанный в п. 3.2.1.

В настоящем подпункте в программе *PLC_PRG* будет объявлен именованный экземпляр функционального блока *DIM7DO_SIMPLE_STIN* из библиотеки *FastwelFbusIO*. Данный блок предполагается использовать для управления двумя выходными каналами модуля DIM713, а также для передачи текущего состояния каналов клиентам MODBUS TCP.

1. Откройте редактор программы *PLC_PRG*, для чего дважды щелкните левой кнопкой мыши на элементе *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – PLC_PRG* в дереве проекта.
2. В области деклараций переменных, расположенной в части редактора, под строкой *PROGRAM PLC_PRG* введите декларацию экземпляра функционального блока *DIM7DO_SIMPLE_STIN* из библиотеки *FastwelFbusIo*:

```
VAR
    m_do_1 : FastwelFbusIO.DIM7DO_SIMPLE_STIN;
END_VAR
```

Внешний вид окна редактора *PLC_PRG* с декларацией переменной-экземпляра функционального блока показан на рисунке 20.

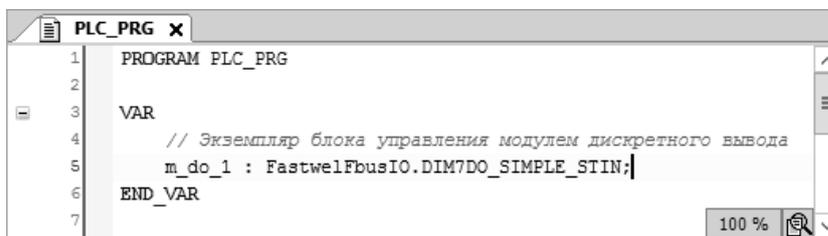


Рисунок 20 – Декларации переменных для управления модулем DIM713

3. Щелкните правой кнопкой мыши на элементе *Device – FBUS – Local (Встроенный порт) – Modules (Мастер FBUS)* в дереве проекта и выполните команду **Изменить ИО соотнесение** в появившемся контекстном меню.
4. На вкладке редактора образа процесса **Изменить И/О-соотнесение** дважды щелкните левой кнопкой мыши в ячейке, расположенной под именем *m_dim713* в столбце **Переменная**, как показано на рисунке 21, после чего нажмите кнопку , появившуюся в ячейке справа.

Переменная	Канал	Адрес	Тип
Modules	Состояние модулей[0]	%ID24	DWORD
	Состояние модулей[1]	%ID28	DWORD
	Транзакций	%ID32	DWORD
	Ошибка	%ID36	DWORD
	Состояние шины	%IW40	Enumeration of INT
m_dim713	Diagnostics	%IB42	BYTE
	States	%IB43	BYTE
	OutputsControl	%QB1	BYTE

Рисунок 21 – Начало соотнесения с имеющейся переменной виртуального диагностического канала модуля

5. В окне **Ассистент ввода**, отображенном на экране, раскройте дерево переменных *Application – PLC_PRG* и выберите переменную *m_do_1.Inputs.Diagnostics*, как показано на рисунке 22, после чего закройте окно **Ассистент ввода** нажатием кнопки **ОК**. В выбранной ячейке редактора образа процесса появится полное имя переменной *Application.PLC_PRG.m_do_1.Inputs.Diagnostics*.

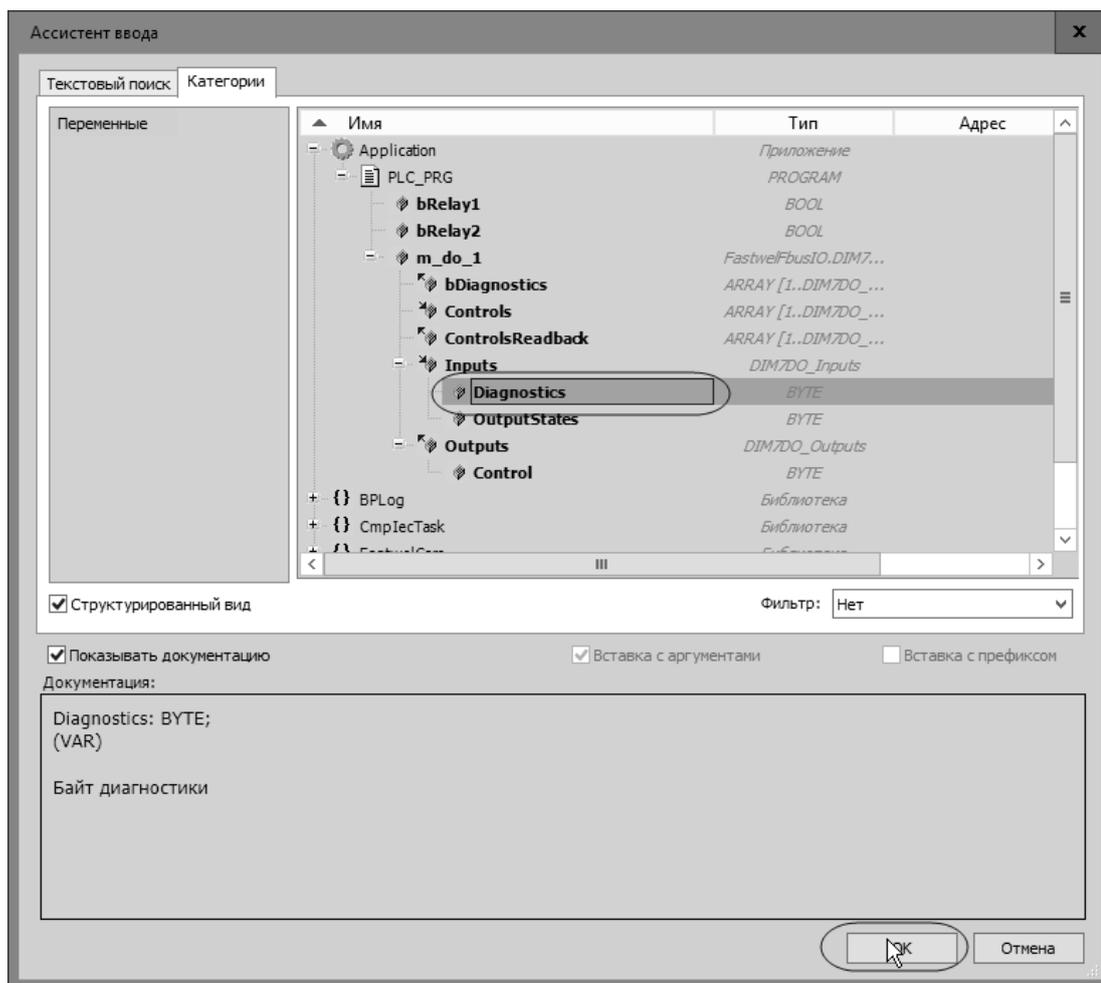


Рисунок 22 – Выбор переменной функционального блока для соотнесения с виртуальным диагностическим каналом модуля

6. Для завершения соотнесения переменной нажмите клавишу Enter или щелкните левой кнопкой мыши во внутренней области редактора образа процесса. Результат соотнесения представлен на рисунке 23.

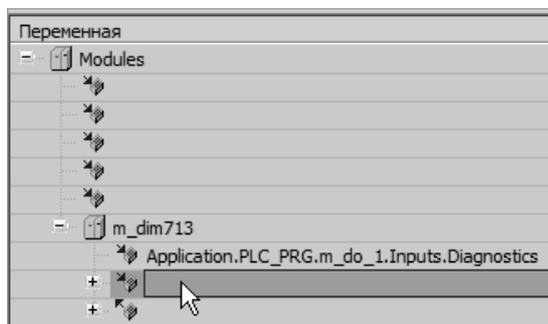


Рисунок 23 – Соотнесение виртуального диагностического канала модуля DIM713 с соответствующим входом функционального блока

7. На вкладке редактора образа процесса **Изменить I/O-соотнесение** дважды щелкните левой кнопкой мыши в ячейке, расположенной слева от имени канала *States*, после чего нажмите кнопку **...**, появившуюся в ячейке справа, как показано на рисунке 24.

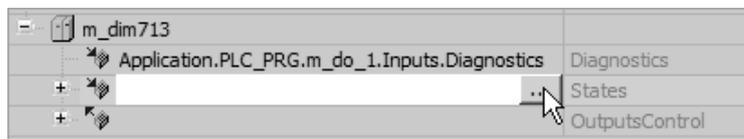


Рисунок 24 – Начало соотнесения с имеющейся переменной канала текущего состояния выходов модуля

8. В отображенном на экране окне **Ассистент ввода** раскройте дерево переменных *Application – PLC_PRG* и выберите переменную *m_do_1.Inputs.OutputStates*, а затем закройте окно **Ассистент ввода** нажатием кнопки **ОК**. В выбранной ячейке редактора образа процесса появится полное имя переменной *Application.PLC_PRG.m_do_1.Inputs.OutputStates*.
9. Для завершения соотнесения переменной нажмите клавишу **Enter** или щелкните левой кнопкой мыши во внутренней области редактора образа процесса.
10. Выполните соотнесение переменной *Application.PLC_PRG.m_do_1.Outputs.Control* с выходным каналом *OutputsControl* в описании модуля *m_dim713* в соответствии с указаниями пп. 7–9. Результат соотнесения каналов модуля с переменными функционального блока представлен на рисунке 25.

m_dim713		
Application.PLC_PRG.m_do_1.Inputs.Diagnostics	Diagnostics	%IB42
Application.PLC_PRG.m_do_1.Inputs.OutputSta...	States	%IB43
Application.PLC_PRG.m_do_1.Outputs.Control	OutputsControl	%QB1

Рисунок 25 – Результат соотнесения переменных приложения с каналами модуля

11. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш **Ctrl-S**.

3.6. Создание конфигурации сервера MODBUS TCP

3.6.1. Общие сведения

Контроллер CPM810-03 обеспечивает возможность запуска до двух серверов (подчиненных устройств) протокола MODBUS TCP и до 16 серверов протокола MODBUS RTU или ASCII через встроенные последовательные порты интерфейсов RS-232C и RS-485, а также через коммуникационные модули NIM741 или NIM742, представленные в конфигурации шины FBUS описаниями модулей *NIM741 RS-485 1xUART Stream Module* или *NIM742 RS-232C 1xUART Stream Module*.

Кроме того, серверы MODBUS TCP могут использоваться в качестве шлюзов между сетью MODBUS TCP и подсетью MODBUS RTU через встроенные последовательные порты и коммуникационные модули NIM741 или NIM742.

В разрабатываемом учебном проекте в конфигурацию приложения будет добавлен один сервер MODBUS TCP без дополнительного конфигурирования в веб-конфигураторе. В результате при запуске приложения сервер будет установлен на все доступные сетевые интерфейсы контроллера.

В таком случае сервер будет принимать входящие соединения от удаленных клиентов через оба сетевых интерфейса, тем самым обеспечивая дублирование соединения между клиентами и сервером на прикладном уровне обмена данными. Дублирование на прикладном уровне означает, что один или несколько клиентов MODBUS TCP должны устанавливать соединение с сервером по двум IP-адресам, назначенным каждому сетевому интерфейсу контроллера, и передавать запросы чтения и записи одних и тех же коммуникационных объектов (регистров и битовых полей) через оба соединения или через соединение, выбранное в качестве основного. Одно из соединений клиент может считать основным, а второе – резервным. В случае отказа связи по основному соединению клиент должен сделать основным исправное резервное соединение и продолжать обмен данными через него. Обратите внимание, что механизм переключения с основного сервера на резервный должен быть реализован на стороне клиента MODBUS TCP или на уровне приложения, взаимодействующего с клиентом.

Если оба физических сетевых порта контроллера достижимы по сети от вычислительного устройства (ПК или другого контроллера), на котором функционирует клиент MODBUS TCP, то клиент устанавливает одно соединение с сервером на контроллере, а отказы связи по любому подключению к физическому порту контроллера будут парированы автоматически на низком уровне стека протоколов TCP/IP путем перенаправления пакетов через второй доступный порт контроллера.

В данном учебном проекте в п. 3.8 будет рассмотрено и реализовано дублирование связи между клиентом MODBUS TCP и сервером контроллера на прикладном уровне.

3.6.2. Создание и настройка объекта сервера MODBUS TCP

Для создания конфигурации сервера MODBUS TCP в учебном проекте:

1. Щелкните правой кнопкой мыши над элементом дерева проектной информации *Device* – *Modbus* и выберите команду **Добавить устройство** в контекстном меню.
2. В окне **Добавить устройство** выберите элемент *Сервер Modbus TCP* и нажмите кнопку **Добавить устройство**. В дереве проектной информации появится элемент *Device* – *Modbus* – *Server (Сервер Modbus TCP)*.
3. Закройте окно **Добавить устройство** нажатием кнопки **Заккрыть**.
4. Для настройки параметров сервера дважды щелкните левой кнопкой мыши над элементом *Device* – *Modbus* – *Server (Сервер Modbus TCP)*. В правой области окна среды разработки будет отображена вкладка редактора устройства, показанная на рисунке 26.

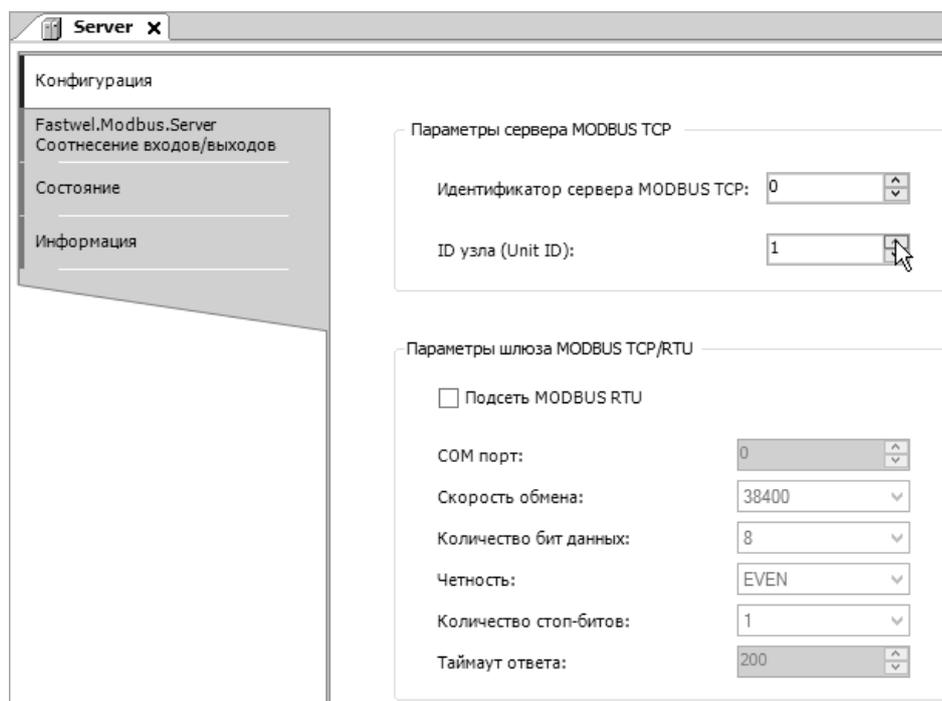


Рисунок 26 – Редактор параметров сервера MODBUS TCP

5. Установите значение параметра **ID узла (Unit ID)** равным 1. Данный параметр определяет идентификатор подчиненного устройства, функционирующего под управлением данного сервера MODBUS TCP, и передается со стороны клиентов (мастеров) в запросах чтения и записи данных сервера.
Если установлен флажок **Подсеть MODBUS RTU**, конфигурируемый сервер MODBUS TCP будет выполнять функцию шлюза между сетью MODBUS TCP и подсетью MODBUS RTU, параметры которой определены группой **Параметры шлюза MODBUS TCP/RTU**. В таком случае все запросы со стороны клиентов со значениями Unit ID, отличными от установленного в поле **ID узла (Unit ID)**, будут передаваться в подсеть MODBUS RTU подчиненным узлам с соответствующими адресами.
Некоторые клиенты MODBUS TCP не позволяют использовать значение Unit ID, равное 0, поэтому в учебном проекте параметр **ID узла (Unit ID)** устанавливается равным 1.
6. Параметр **Идентификатор сервера MODBUS TCP** может принимать значения 0 или 1, и служит для соотнесения экземпляра сервера с IP-адресом, назначенным сетевому интерфейсу контроллера. Данное соотнесение устанавливается при помощи веб-конфигуратора в группе системных параметров **Параметры MODBUS**. Например, если предполагается использовать сетевые интерфейсы контроллера с настройкой IP-адресов в одной подсети и установить сервер с идентификатором 0 на IP-адрес сетевого интерфейса LAN2, то параметры MODBUS в веб-конфигураторе должны быть настроены, как показано на рисунке 27.

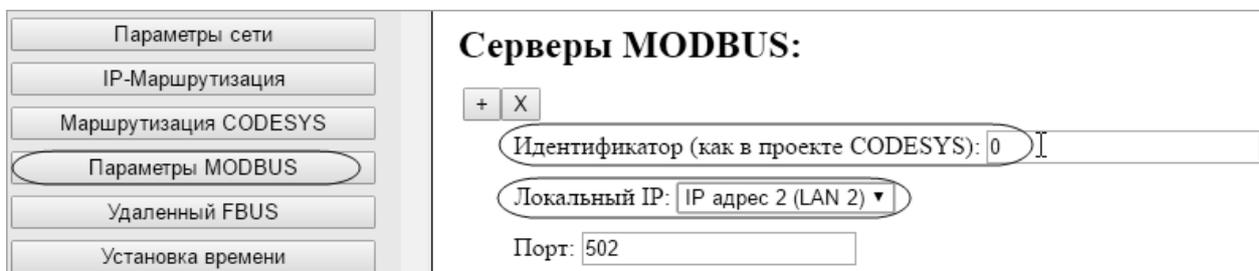


Рисунок 27 – Отображение идентификатора сервера 0 с IP-адресом сетевого интерфейса LAN2

Если приложению для контроллера с двумя независимыми сетевыми интерфейсами требуется использовать два экземпляра сервера MODBUS TCP, то для их работы необходимо присвоить каждому экземпляру собственный идентификатор, а затем в веб-конфигураторе выполнить соотнесение идентификаторов с IP-адресами интерфейсов. Как указывалось в п. 3.6.1, в рассматриваемом учебном проекте создается один сервер MODBUS TCP без дополнительного конфигурирования системных сервисов контроллера при помощи веб-конфигуратора. Данный сервер будет автоматически установлен на все доступные сетевые интерфейсы контроллера.

7. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш **Ctrl-S**.

3.6.3. Создание объектов доступа к данным

3.6.3.1. Общие сведения

Для связи серверов MODBUS TCP и MODBUS RTU/ASCII с приложением, выполняющемся на контроллере, в конфигурации каждого сервера должны быть определены области ввода и вывода данных. Область ввода *Входы* служит для обслуживания запросов записи регистров хранения (Holding Registers) и битовых полей типа Coil, поступающих к серверу со стороны удаленных клиентов MODBUS, а область вывода *Выходы* – для обслуживания запросов чтения регистров Input Registers и битовых полей типа Discrete Input.

Для каждой области ввода или вывода устанавливается начальный адрес регистра, с которого будут нумероваться принадлежащие данной области регистры и битовые поля, отображаемые на объекты доступа к данным приложения, созданные внутри области.

3.6.3.2. Создание областей ввода и вывода сервера MODBUS

Для определения областей ввода и вывода в конфигурации сервера в учебном проекте:

1. Щелкните правой кнопкой мыши над элементом *Device – Modbus – Server (Сервер Modbus TCP)* и выберите команду **Добавить устройство** в контекстном меню.
2. В окне **Добавить устройство** выберите элемент *Входы* и нажмите кнопку **Добавить устройство**. В дереве проектной информации появится элемент *Device – Modbus – Server (Сервер Modbus TCP) – Inputs (Входы)*.
3. В окне **Добавить устройство** выберите элемент *Выходы* и нажмите кнопку **Добавить устройство**. В дереве проектной информации появится элемент *Device – Modbus – Server (Сервер Modbus TCP) – Outputs (Выходы)*.
4. Закройте окно **Добавить устройство** нажатием кнопки **Заккрыть**. Обратите внимание, что имена элементов *Inputs* и *Outputs* могут быть изменены двукратным (с паузой) щелчком левой кнопки мыши над элементом в дереве проекта и вводом нового имени, как показано на рисунке 28.

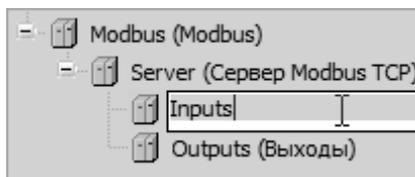


Рисунок 28 – Редактирование имени области ввода данных сервера MODBUS TCP

3.6.3.3. Создание объектов доступа к данным сервера

В рассматриваемом учебном проекте предполагается иметь объекты доступа к данным сервера MODBUS TCP, позволяющие управлять состоянием второго канала модуля DIM713, передавать клиенту MODBUS TCP текущее состояние обоих каналов модуля DIM713 и 32-разрядный счетчик циклов программы PLC_PRG.

Для создания объекта данных в области ввода данных сервера:

1. В дереве проекта откройте редактор устройства *Inputs (Входы)* двойным щелчком левой кнопки мыши на элементе *Device – Modbus – Server (Сервер Modbus TCP) – Inputs (Входы)*. Открытая вкладка редактора показана на рисунке 29.

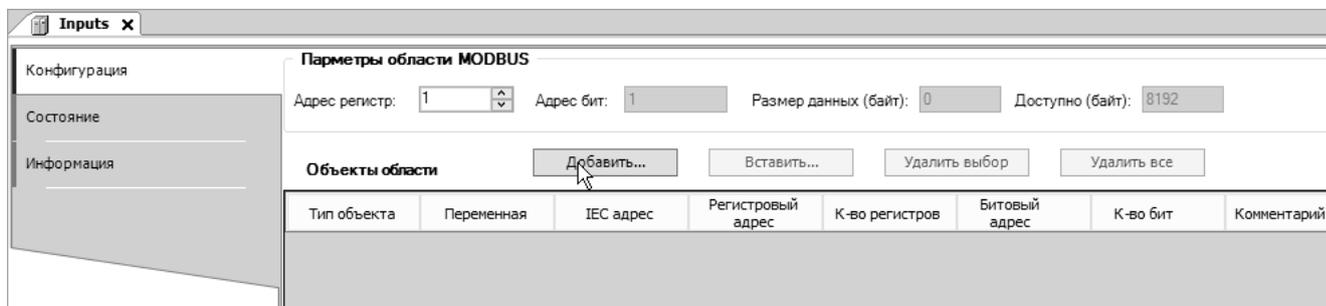


Рисунок 29 – Редактор области ввода данных сервера

2. Нажмите кнопку **Добавить**. На экране будет отображено окно **Добавить объекты данных**, показанное на рисунке 30.

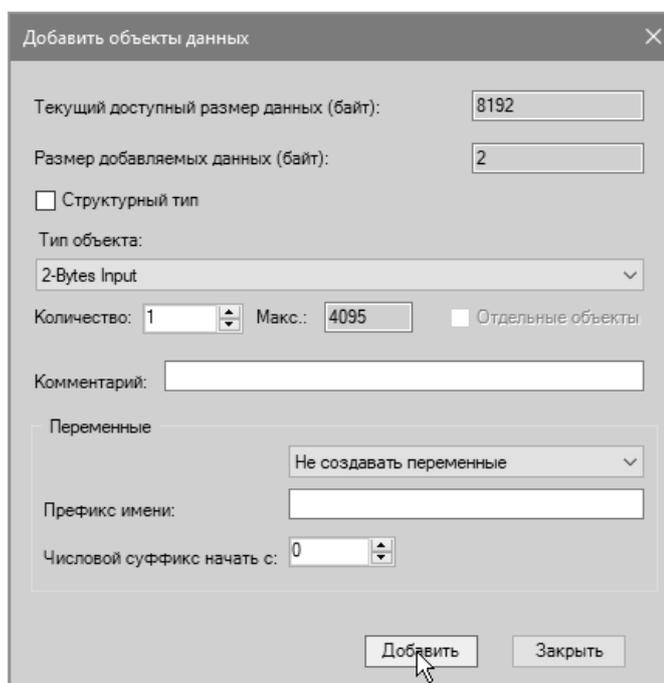


Рисунок 30 – Окно создания объектов данных в области ввода сервера

Выпадающий список **Тип объекта** позволяет выбрать тип создаваемого объекта среди доступных: *WORD Input*, *DWORD Input*, *REAL Input*, *LREAL Input* или *2-Bytes Input*.

Поле **Количество** предназначено для определения количества создаваемых объектов выбранного типа. Если добавляется более одного объекта данных, то по умолчанию объекты будут добавлены в виде массива. Данный способ добавления множества объектов данных наиболее эффективен с точки зрения минимизации влияния на быстродействие среды разработки.

При необходимости можно отметить флажок **Отдельные объекты**, и тогда в область ввода или вывода сервера будут добавлены отдельные объекты данных. Данный способ менее эффективен с точки зрения влияния на быстродействие среды разработки при последующей работе над проектом.

Поле **Префикс имени переменных** позволяет задать общую часть имени одной или нескольких переменных приложения, создаваемых непосредственно в области ввода

или вывода. Если установлена опция **Отдельные объекты**, то создаваемые переменные выбранного типа будут иметь имена, состоящие из заданного префикса и числового суффикса, начальное значение которого может быть задано в поле **Числовой суффикс начать с**.

3. Выберите тип объекта *2-Bytes Input* и, не изменяя текущих значений в остальных окна, нажмите кнопку **Добавить**, а затем **Заккрыть**. Объект *2-Bytes Input* будет отображен в таблице **Объекты области**, как показано на рисунке 31.

Данный объект будет впоследствии использоваться для передачи команды управления вторым выходным каналом модуля DIM713.

Параметры области MODBUS											
Адрес регистр:		1	Адрес бит:		1	Размер данных (байт):	2	Доступно (байт):	8190		
Объекты области								Добавить...	Вставить...	Удалить выбор	Удалить все
Тип объекта	Переменная	IEC адрес	Регистровый адрес	К-во регистров	Битовый адрес	К-во бит	Комментарий				
2-Bytes Input		%IW52	1	1	1	16					

Рисунок 31 – Объект типа *WORD Input* в области ввода сервера

Ячейки таблицы содержат информацию о связи объекта с приложением и параметры запроса к объекту по протоколу MODBUS:

Переменная – содержит имя переменной, созданной во входном образе процесса приложения, или имя существующей переменной приложения, связанной с объектом данных. Обратите внимание, что данная переменная будет доступна в приложении только для чтения.

IEC адрес – содержит смещение объекта данных во входном образе процесса приложения.

Регистровый адрес – начальный адрес регистра хранения (Holding Register) в сетевых запросах чтения или записи объекта по протоколу MODBUS. Обратите внимание, что значение адреса, отображаемое в данном столбце, на единицу превышает значение адреса регистра, которое должно передаваться серверу удаленными клиентами в сетевых запросах.

К-во регистров – максимальное количество регистров, которое может передаваться серверу в сетевых запросах чтения или записи объекта.

Битовый адрес – начальный адрес битового поля типа Coil в сетевых запросах чтения или записи битовых полей объекта. Значение адреса в данном столбце на единицу превышает значение адреса битового поля, передаваемое в сетевом запросе серверу.

К-во бит – максимальное количество битовых полей, которое может передаваться в сетевых запросах чтения или записи объекта.

Для записи значения созданного объекта данных по сети мастером (клиентом) MODBUS TCP могут использоваться следующие запросы протокола MODBUS: (05) Write Single Coil; (06) Write Holding Register; (15) Write Multiple Coils; (16) Write Multiple Registers; (23) Read-Write Multiple Registers.

Для чтения значения созданного объекта данных по сети клиентом могут использоваться запросы MODBUS: (01) Read Coil, (03) Read Holding Registers, (23) Read-Write Multiple Registers.

Адреса регистров и битовых полей и количество объектов в запросах должны соответствовать указанным в таблице объектов области с учетом отличия значений регистровых и битовых адресов на 1 от фактически передаваемых в запросах.

Для создания объектов данных в области вывода данных сервера:

1. В дереве проекта откройте редактор устройства *Outputs (Выходы)* двойным щелчком левой кнопкой мыши на элементе *Device – Modbus – Server (Сервер Modbus TCP) – Outputs (Выходы)*. Открытая вкладка редактора идентична показанной на рисунке 29.
2. Нажмите кнопку **Добавить**. На экране будет отображено окно **Добавить объекты данных**, показанное на рисунке 32.

3. Выберите *DWORD Output* в выпадающем списке **Тип объекта** и нажмите кнопку **Добавить**. В таблице редактора появится строка с информацией о только что созданном объекте типа *DWORD Output*.
Данный объект впоследствии будет использоваться для передачи счетчика циклов программы *PLC_PRG* клиентам MODBUS TCP.
4. Не закрывая окно **Добавить объекты данных**, выберите тип объекта *2-Bytes Output*, нажмите кнопку **Добавить**, а затем **Заккрыть**. В таблице редактора появится строка с информацией о только что созданном объекте типа *2-Bytes Output*, как показано на рисунке 33. Данный объект впоследствии будет использоваться для передачи текущего состояния выходных каналов модуля DIM713 клиентам MODBUS TCP

Рисунок 32 – Диалоговая панель создания коммуникационного объекта в области вывода сервера

Параметры области MODBUS							
Адрес регистра:	1	Адрес бит:	1	Размер данных (байт):	6	Доступно (байт):	8186
Объекты области							
		Добавить...		Вставить...		Удалить выбор	
		Удалить все					
Тип объекта	Переменная	IEC адрес	Регистровый адрес	К-во регистров	Битовый адрес	К-во бит	Комментарий
DWORD Output		%QD8	1	2	1	32	
2-Bytes Output		%QW12	3	1	33	16	

Рисунок 33 – Объекты типа *DWORD Output* и *WORD Output* в области вывода сервера

Ячейки таблицы содержат информацию о связи коммуникационного объекта с приложением и параметры запроса к объекту по протоколу MODBUS:

Переменная – содержит имя переменной, созданной во выходном образе процесса приложения, или имя существующей переменной приложения, связанной с объектом данных. Переменная в выходном образе процесса будет доступна в приложении для чтения и записи.

IEC адрес – содержит смещение переменной во выходном образе процесса приложения.

Регистровый адрес – начальный адрес входного регистра (Input Register) в сетевых запросах чтения объекта по протоколу MODBUS. Обратите внимание, что значение адреса, отображаемое в данном столбце, на единицу превышает значение адреса регистра, которое должно передаваться серверу удаленными клиентами в сетевых запросах.

К-во регистров – максимальное количество регистров, которое может передаваться в сетевых запросах чтения объекта.

Битовый адрес – начальный адрес битового поля типа Discrete Input в сетевых запросах чтения битовых полей объекта по протоколу MODBUS. Значение адреса в данном столбце на единицу превышает значение адреса битового поля, передаваемое в сетевом запросе серверу.

К-во бит – максимальное количество битовых полей, передаваемое в сетевых запросах чтения всего объекта.

Для чтения значений созданных объектов данных по сети мастером (клиентом) MODBUS TCP могут использоваться запросы MODBUS: (02) Read Input, (04) Read Input Registers. Адреса регистров и битовых полей и количество объектов в запросах должны соответствовать указанным в таблице объектов области с учетом отличия значений регистровых и битовых адресов на 1 от фактически передаваемых в запросах.

5. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш Ctrl-S.

3.6.4. Создание связи приложения с коммуникационными объектами

Информация о доступных способах организации связи приложения с каналами модулей ввода-вывода и коммуникационных объектов внешней сети приведена в п. 3.5.5.1. В рассматриваемом учебном проекте будет использован способ, при котором имеющиеся переменные приложения соотносятся с адресами коммуникационных объектов в образе процесса.

При выполнении указаний п. 3.5.5.2 в программе PLC_PRG был создан именованный экземпляр функционального блока *DIM7DO_SIMPLE_STIN* из библиотеки FastwelFbusIO и установлена связь его входных и выходных переменных с входными и выходными каналами модуля DIM713.

В настоящем подразделе в программе PLC_PRG будут объявлена переменная *dwCycleCounter* типа DWORD, которая будет использоваться для передачи удаленному клиенту MODBUS счетчика циклов программы PLC_PRG.

Для связи объектов данных сервера MODBUS TCP с каналами модуля DIM713 будут использоваться входные и выходные переменные функционального блока *m_do_1* типа *DIM7DO_SIMPLE_STIN*, объявленного в программе PLC_PRG при выполнении указаний п. 3.5.5.2.

1. Откройте редактор программы *PLC_PRG*, для чего дважды щелкните левой кнопкой мыши на элементе *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – PLC_PRG* в дереве проекта.
2. В области деклараций переменных, расположенной в части редактора, в блоке VAR ... END_VAR введите декларацию переменной *dwCycleCounter*:

```
VAR
  // Экземпляр блока управления модулем дискретного вывода
  m_do_1 : FastwelFbusIO.DIM7DO_SIMPLE_STIN;
  // Счетчик циклов PLC_PRG
  dwCycleCounter : DWORD;
END_VAR
```

Внешний вид окна редактора *PLC_PRG* с уже созданными декларациями переменных показан на рисунке 34.

3. Щелкните правой кнопкой мыши на элементе *Device – Modbus – Server (Сервер Modbus TCP)* в дереве проекта и выполните команду **Изменить IO соотношение** в появившемся контекстном меню.

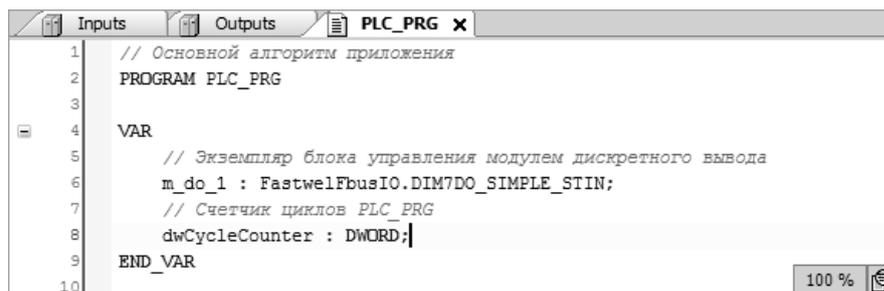


Рисунок 34 – Декларации переменных для обмена по сети

4. На вкладке редактора образа процесса **Изменить I/O-соотнесение** в столбце **Переменная** раскройте элемент *Server*, а затем элементы *Inputs* и *2-Bytes Input*, как показано на рисунке 35.
5. Дважды щелкните левой кнопкой мыши в ячейке *Inputs – 2-Bytes Input – Bit1* в столбце **Переменная**, как показано на рисунке 35, и нажмите кнопку , появившуюся в ячейке справа.

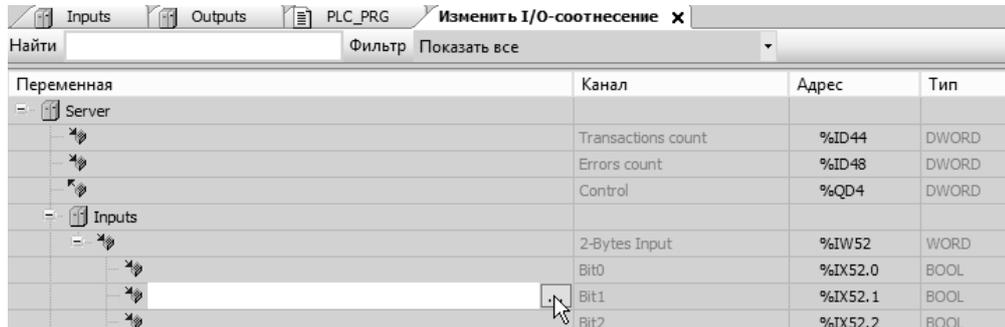


Рисунок 35 – Начало соотнесения с имеющейся переменной *PLC_PRG.wNetRelayControl*

6. В окне **Ассистент ввода** выберите переменную *Application – PLC_PRG – m_do1_1 – Controls*, как показано на рисунке 36, и нажмите кнопку **ОК**.

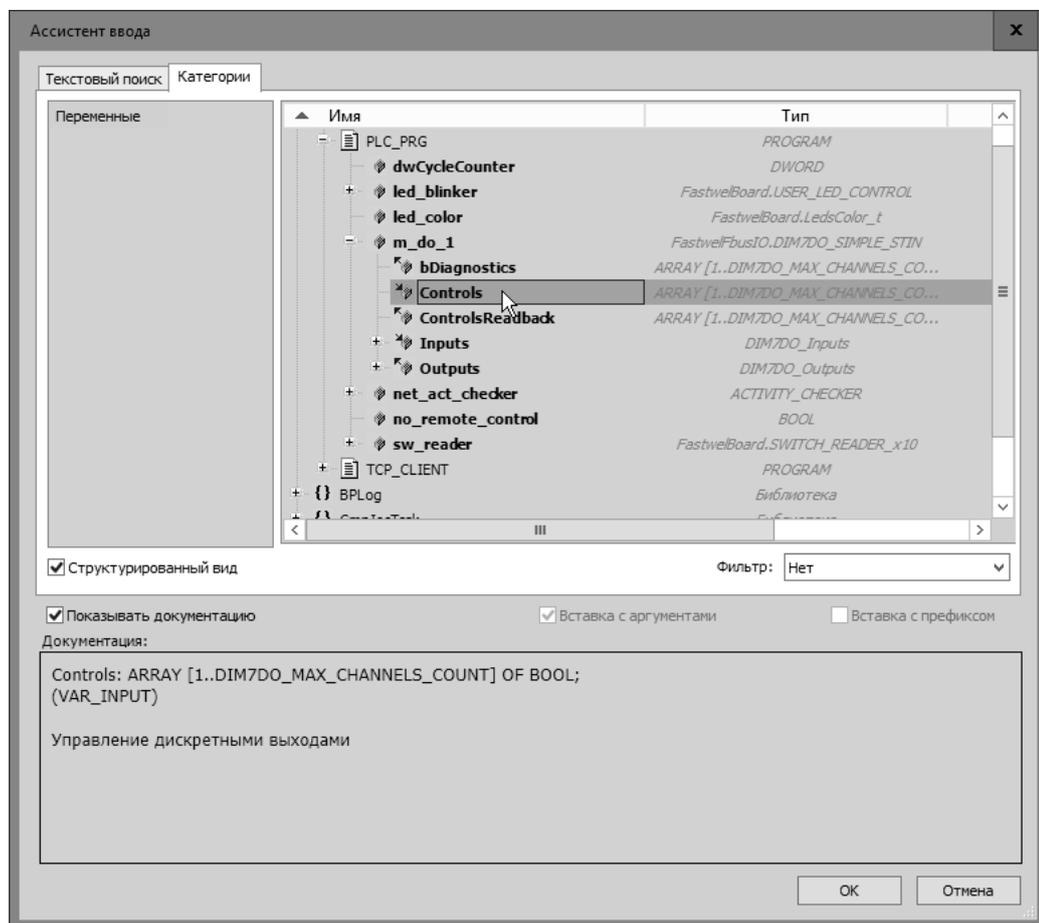


Рисунок 36 – Выбор переменной для соотнесения с областью ввода сервера

7. Для выбора второго элемента во входном массиве *m_do1_1 – Controls* для соотнесения в ячейке таблицы соотнесения введите *Controls[2]*, как показано на рисунке 37.

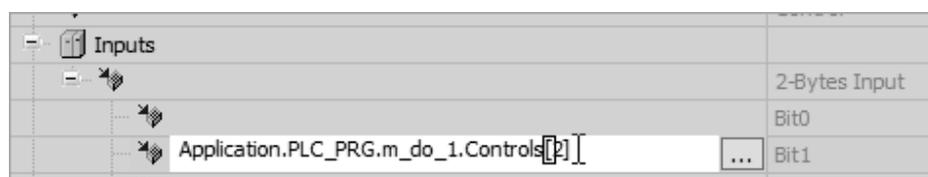


Рисунок 37 – Соотнесение со вторым элементом массива *Controls* во входах функционального блока *m_do1_1*

8. Для завершения соотнесения переменной нажмите клавишу Enter или щелкните левой кнопкой мыши во внутренней области редактора образа процесса.
9. На вкладке редактора образа процесса **Изменить I/O-соотнесение** в столбце **Переменная** раскройте элемент *Server – Outputs* и выполните соотнесение переменной *Application.PLC_PRG.dwCycleCounter* с объектом доступа к данным сервера *Server – Outputs – DWORD Output* в соответствии с указаниями шагов 5–8.
 На вкладке редактора образа процесса **Изменить I/O-соотнесение** в столбце **Переменная** раскройте элемент *Server – Outputs – 2-Bytes Output* и выполните соотнесение переменных *Application.PLC_PRG.m_do_1.ControlsReadback[1]* и *Application.PLC_PRG.m_do_1.ControlsReadback[2]* с объектами доступа к данным сервера *Server – Outputs – 2-Bytes Output – Bit0* и *Bit1* соответственно.
 Результат соотнесения показан на рисунке 38.
10. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш Ctrl-S.

Переменная	Канал	Адрес	Тип
Server			
Transactions count	Transactions count	%ID44	DWORD
Errors count	Errors count	%ID48	DWORD
Control	Control	%QD4	DWORD
Inputs			
2-Bytes Input	2-Bytes Input	%IW52	WORD
Application.PLC_PRG.m_do_1.Controls[2]	Bit0	%IX52.0	BOOL
Bit1	Bit1	%IX52.1	BOOL
Bit2	Bit2	%IX52.2	BOOL
Bit3	Bit3	%IX52.3	BOOL
Bit4	Bit4	%IX52.4	BOOL
Bit5	Bit5	%IX52.5	BOOL
Bit6	Bit6	%IX52.6	BOOL
Bit7	Bit7	%IX52.7	BOOL
Bit8	Bit8	%IX53.0	BOOL
Bit9	Bit9	%IX53.1	BOOL
Bit10	Bit10	%IX53.2	BOOL
Bit11	Bit11	%IX53.3	BOOL
Bit12	Bit12	%IX53.4	BOOL
Bit13	Bit13	%IX53.5	BOOL
Bit14	Bit14	%IX53.6	BOOL
Bit15	Bit15	%IX53.7	BOOL
Outputs			
Application.PLC_PRG.dwCycleCounter	DWORD Output	%QD8	DWORD
2-Bytes Output	2-Bytes Output	%QW12	WORD
Application.PLC_PRG.m_do_1.ControlsReadback[1]	Bit0	%QX12.0	BOOL
Application.PLC_PRG.m_do_1.ControlsReadback[2]	Bit1	%QX12.1	BOOL

Рисунок 38 – Результат соотнесения переменных приложения с объектами данных сервера

3.7. Создание конфигурации мастера MODBUS TCP

3.7.1. Общие сведения

Контроллер CPM810-03 обеспечивает возможность запуска до двух мастеров (клиентов) протокола MODBUS TCP и до 16 мастеров протокола MODBUS RTU или ASCII через встроенные последовательные порты интерфейса RS-232C или RS-485 и через коммуникационные модули NIM741 или NIM742, представленные в конфигурации шины описаниями модулей *NIM741 RS-485 1xUART Stream Module* или *NIM742 RS-232C 1xUART Stream Module*.

В разрабатываемом учебном проекте в конфигурацию приложения будет входить один мастер MODBUS TCP. В конфигурацию мастера будут добавлены идентичные описания двух подчиненных устройств (серверов) с IP-адресами сетевых интерфейсов этого же контроллера. Такая конфигурация будет обеспечивать связь мастера с одним физическим подчиненным устройством через два

независимых соединения, поскольку сервер MODBUS TCP, конфигурация которого создана в п. 3.6, принимает входящие соединения от удаленных мастеров через оба независимых сетевых интерфейса.

В случае отказа связи по одному из соединений алгоритм приложения, связанный с клиентом, должен обнаружить эту ситуацию и использовать для обмена с удаленным сервером данные, соответствующие исправному соединению. Механизм переключения с основного сервера на резервный будет реализован на уровне приложения, взаимодействующего с клиентом.

При использовании динамического назначения IP-адресов сетевым интерфейсам контроллера (переключатели в положении 254) отказы связи могут быть имитированы отсоединением кабелей К3 или К4 от коммутатора или портов контроллера, показанных на рисунке 4.

3.7.2. Создание и настройка объекта мастера MODBUS TCP

Для создания конфигурации клиента (мастера) MODBUS TCP в учебном проекте:

1. Щелкните правой кнопкой мыши над элементом дерева проектной информации *Device – Modbus* и выберите команду **Добавить устройство** в контекстном меню.
2. В окне **Добавить устройство** выберите элемент *Master Modbus TCP* и нажмите кнопку **Добавить устройство**. В дереве проектной информации появится элемент *Device – Modbus – Master (Master Modbus TCP)*.
3. Закройте окно **Добавить устройство** нажатием кнопки **Заккрыть**.
4. Для настройки параметров клиента MODBUS TCP дважды щелкните левой кнопкой мыши над элементом *Device – Modbus – Master (Master Modbus TCP)*. В правой области окна среды разработки будет отображена вкладка редактора устройства, показанная на рисунке 39.

5. Параметр **Идентификатор мастера MODBUS TCP** может принимать значения 0 или 1, и служит для соотнесения экземпляра мастера с IP-адресом сетевого интерфейса контроллера. Данное соотнесение устанавливается при помощи веб-браузера в группе системных параметров **Параметры MODBUS** в случае, если требуется использовать один или два экземпляра мастера и посылать запросы соединения и обмена данными с удаленными серверами через определенный сетевой интерфейс.

Обратите внимание, что при назначении IP-адресов двум сетевым интерфейсам контроллера в одной подсети без явного соотнесения экземпляров мастера MODBUS TCP с сетевыми интерфейсами контроллера все запросы к серверам на других физических узлах этой подсети будут передаваться только через интерфейс LAN1.

В рассматриваемом учебном проекте будет использован единственный экземпляр мастера MODBUS TCP без дополнительных настроек в веб-конфигураторе, поскольку мастер будет передавать запросы серверу, запущенному на том же контроллере, что и он сам. В таком случае сетевой обмен между мастером и сервером будет происходить без передачи пакетов по каналу связи.

6. Параметр **Гранулярность опроса, мс** устанавливает период времени в миллисекундах, по истечении которого мастер планирует выполнение сетевых запросов к подчиненным узлам (серверам).

Сетевые запросы к серверам определяются коммуникационными объектами нескольких типов в конфигурации мастера и, помимо соотнесения с данными приложения, имеют целочисленный параметр **Тип передачи**, который может принимать значения от 0 до 255.

Значение 0 означает, что данный коммуникационный объект исключен из расписания опроса и не будет передаваться в сеть, а значения от 1 до 250 определяют количество интервалов гранулярности опроса, по истечении которых должен передаваться запрос.

Например, если гранулярность опроса составляет 50 мс, а тип передачи коммуникационного объекта установлен равным 1, то мастер будет пытаться передавать соответствующий запрос чтения или записи с периодом $50 \times 1 = 50$ мс.

Для коммуникационных объектов, соответствующих запросам записи (05) Write Single Coil, (06) Write Holding Register, (15) Write Multiple Coils, (16) Write Multiple Registers, (23) Read-Write Multiple Registers) возможно использовать тип передачи 255, который предназначен для передачи соответствующего запроса записи только при изменении данных, которые должны быть переданы подчиненному узлу в этом запросе.

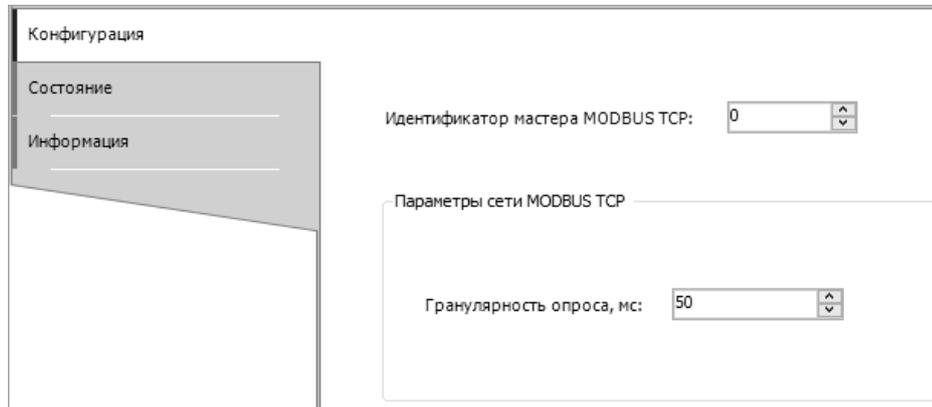


Рисунок 39 – Конфигурация мастера MODBUS TCP

7. Для добавления в конфигурацию мастера описания подчиненного узла щелкните правой кнопкой мыши над элементом *Device – Modbus – Master (Master Modbus TCP)* и выберите команду **Добавить устройство** в контекстном меню.
8. В окне **Добавить устройство** выберите элемент *Подчиненный узел*, в поле **Имя** введите *Slave_1_1* и нажмите кнопку **Добавить устройство**. В дереве проектной информации появится элемент *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1*.
9. Не закрывая окно **Добавить устройство**, выберите элемент *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1* в дереве проекта, а затем в окне **Добавить устройство** выберите элемент *Коммуникационный объект*, в поле **Имя** введите *SrvAppCycles_1* и нажмите кнопку **Добавить устройство**. В дереве проектной информации появится элемент *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1 – SrvAppCycles_1*. Данный коммуникационный объект будет использоваться для получения счетчика циклов приложения на сервере.
10. Не закрывая окно **Добавить устройство**, добавьте в конфигурацию еще два коммуникационных объекта с именами *RelayControl_1* и *RelayStates_1*. Данные коммуникационные объекты будут использоваться для передачи бинарной команды управления вторым каналом модуля DIM713 и для получения текущего состояния каналов этого же модуля.
11. Закройте окно **Добавить устройство** нажатием кнопки **Закреть**. Конфигурация мастера MODBUS TCP в дереве проектной информации будет выглядеть, как показано на рисунке 40.
12. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки.

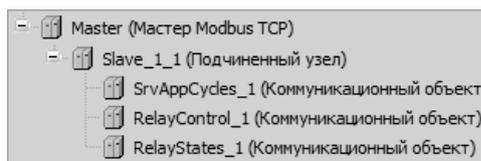


Рисунок 40 – Конфигурация мастера MODBUS TCP с одним подчиненным узлом (сервером)

13. Для настройки параметров подчиненного узла *Slave_1_1* в конфигурации мастера дважды щелкните левой кнопкой мыши на элементе *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1* в дереве проекта. В правой области главного окна среды разработки будет отображена вкладка редактора устройства *Slave_1_1*, показанная на рисунке 41.

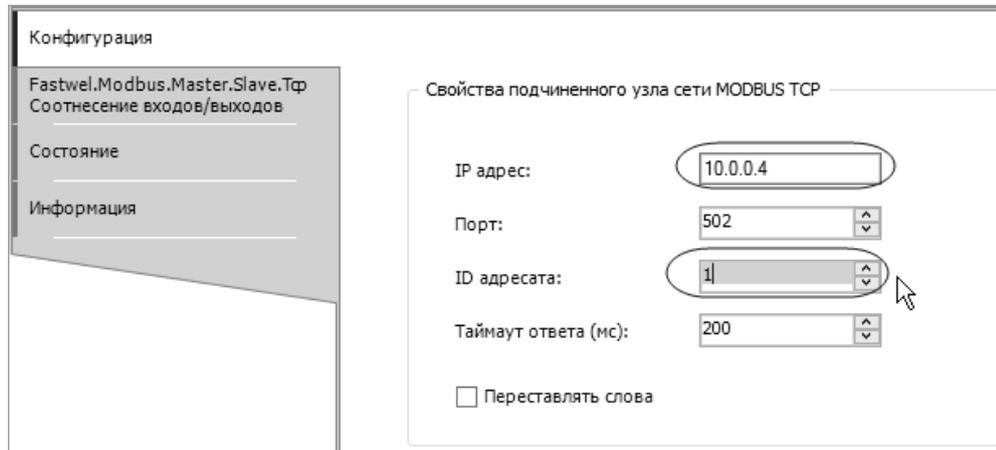


Рисунок 41 – Редактор конфигурации подчиненного узла MODBUS TCP

14. В поле **IP адрес** введите IP-адрес второго сетевого интерфейса контроллера. Если на первых восьми переключателях контроллера установлено значение 3, что соответствует статическому назначению адресов (см. рисунок 5), то введите IP-адрес 10.0.0.4.
Если на первых восьми переключателях установлено значение 254, что соответствует динамическому назначению адресов, то введите IP-адрес второго интерфейса (LAN2), выясненный при выполнении команды оболочки ПЛК (**PLC Shell**) *ipinfo* в соответствии с указаниями п. 3.2.3.
15. Оставьте неизменным значение в поле **Порт**, которое соответствует номеру TCP-порта, по умолчанию используемому протоколом MODBUS TCP.
16. Установите значение параметра **ID адресата** равным 1. Данное значение должно соответствовать заданному для параметра **ID узла (Unit ID)** сервера (см. п. 3.6.2.).
17. Оставьте неизменными значения параметров **Таймаут ответа (мс)** и **Переставлять слова**. Параметр **Таймаут ответа (мс)** определяет интервал времени, в течение мастер будет ожидать ответ сервера на очередной запрос. Поскольку подчиненные узлы MODBUS TCP опрашиваются мастером независимо, длительность таймаута не влияет на время выполнения расписания опроса в отношении других подчиненных узлов.
Параметр **Переставлять слова** предназначен для перестановки 2-байтовых слов данных длиной более 2 байт перед отправкой серверу в запросах записи и после получения ответов сервера на запросы чтения. Поскольку в учебном проекте на сервере и клиенте используется одинаковый порядок упаковки 2-байтовых слов в элементах данных длиной более одного 2-байтового слова, перестановку слов выполнять не требуется.
18. Для настройки параметров сетевого запроса *Slave_1_1 – SrvAppCycles_1* в конфигурации мастера дважды щелкните левой кнопкой мыши на элементе *Device – Modbus – Master (Мастер Modbus TCP) – Slave_1_1 – SrvAppCycles_1* в дереве проекта. В правой области главного окна среды разработки будет отображена вкладка редактора устройства *SrvAppCycles_1*, соответствующая настраиваемому запросу и показанная на рисунке 42.
Данный запрос будет использоваться для чтения счетчика циклов приложения, выполняющегося на сервере и доступного по адресу 1 входного регистра (Input Register). В связи с этим оставьте неизменными значения параметров **Тип доступа** и **Адрес чтения**. Первый параметр определяет тип запроса, который по умолчанию установлен равным *Read Input Registers (04h)*, а второй – начальный адрес набора регистров, который должен быть прочитан на сервере данным запросом. Обратите внимание, что значение **Адрес чтения** на единицу превышает реальное значение адреса регистра, передаваемое в сетевом запросе.
Кроме того, оставьте неизменным значение параметра **Тип передачи**, который определяет период передачи запроса серверу в миллисекундах в виде произведения значения **Гранулярность опроса** на значение данного параметра. В рассматриваемом учебном проекте значение гранулярности опроса установлено равным 50 мс (см. шаг б), т.е. данный запрос будет передаваться серверу с периодом 50 мс.

Параметры запроса

Тип доступа: Тип передачи:

Адрес чтения: Использовано слотов чтения: Доступно слотов чтения:

Адрес записи: Использовано слотов записи: Доступно слотов записи:

Объекты Чтение Запись

Тип объекта	Имя переменной	Адрес IEC	Смещение	Размер	Комментарий
DWORD Input		%ID128	0	2	

Рисунок 42 – Параметры запроса двух регистров типа Input Register

19. Определите данные, которые требуется получать у подчиненного узла, и область памяти во входном образе процесса, в которую будут помещены полученные данные. Для этого нажмите кнопку **Добавить** и в окне **Добавить объекты данных** выберите тип объекта *DWORD Input*, нажмите кнопку **Добавить** и **Заккрыть**. Объект данных будет отображен в таблице **Объекты**, как показано на рисунке 42. Размер данных у типа объекта *DWORD Input* соответствует размеру *DWORD Output*, который определен на сервере для передачи значения счетчика циклов.

Поле **Использовано слотов чтения** содержит количество 2-байтовых слов, запрашиваемых у сервера, начиная с адреса регистра **Адрес чтения**.

Поле **Доступно слотов чтения** содержит количество 2-байтовых слов, которое еще можно добавить в данный запрос. Запрос 04 может использоваться для чтения до 125 регистров, поэтому после добавления объекта *DWORD Input* остаются доступными 123 слота чтения.

20. Для настройки параметров сетевого запроса *Slave_1_1 – RelayControl_1* в конфигурации мастера дважды щелкните левой кнопкой мыши на элементе *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1 – RelayControl_1* в дереве проекта.
21. На вкладке редактора устройства *RelayControl_1* установите следующие значения параметров:

Тип доступа: *Write Single Coil (05h)*

Тип передачи: 255

Адрес записи: 2

Данный запрос будет использоваться для записи бинарной команды включения или выключения второго канала модуля DIM713, который на сервере соотнесен со вторым битовым полем типа *Coil*, поэтому параметр **Адрес записи** должен иметь значение 2.

В качестве типа запроса в поле **Тип доступа** установлен *Write Single Coil (05h)*, поскольку такой запрос вполне достаточен для записи в единственное битовое поле.

Для того, чтобы запрос записи передавался только при изменении переменной, соотнесенной с запросом, параметр **Тип передачи** сделан равным 255.

22. Определите область в выходном образе процесса приложения, из которой будут передаваться данные подчиненному узлу в запросе записи. Для этого нажмите кнопку **Добавить** и в окне **Добавить объекты данных** выберите тип объекта *1-Bit Output*, нажмите кнопку **Добавить** и **Заккрыть**. Объект данных будет отображен в таблице **Объекты**, как показано на рисунке 43.

Поскольку в запросе *Write Single Coil (05h)* подчиненному узлу может быть передано значение единственного битового поля, поле **Доступно слотов записи** содержит нулевое значение, и добавление дополнительных объектов данных в запрос заблокировано.

Параметры запроса

Тип доступа: Write Single Coil (05h) Тип передачи: 255

Адрес чтения: 1 Использовано слотов чтения: 0 Доступно слотов чтения: 0

Адрес записи: 2 Использовано слотов записи: 1 Доступно слотов записи: 0

Объекты Чтение Запись

Тип объекта	Имя переменной	Адрес IEC	Смещение	Размер	Комментарий
1-Bit Output		%QX28.0	0	1	

Рисунок 43 – Параметры запроса записи битового поля типа Coil

23. Для настройки параметров сетевого запроса *Slave_1_1 – RelayStates_1* в конфигурации мастера дважды щелкните левой кнопкой мыши на элементе *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1 – RelayStates_1* в дереве проекта.

24. На вкладке редактора устройства *RelayStates_1* установите следующие значения параметров:

Тип доступа: *Read Discrete Inputs (02h)*

Тип передачи: *1*

Адрес чтения: *33*

Данный запрос будет использоваться для чтения состояния двух каналов модуля DIM713, которые на сервере соотнесены со битовыми полями типа Discrete Input с адресами 33 и 34 (см. рисунок 33), поэтому параметр **Адрес чтения** должен иметь значение 33.

В качестве типа запроса в поле **Тип доступа** установлен *Read Discrete Inputs (02h)*, поскольку такой запрос может использоваться для чтения нескольких битовых полей.

Для того, чтобы запрос чтения передавался с периодом 50 мс, параметр **Тип передачи** сделан равным 1.

25. Определите область памяти во входном образе процесса, в которую будут помещены данные, полученные от подчиненного узла. Для этого нажмите кнопку **Добавить**, в окне **Добавить объекты данных** выберите тип объекта *1-Bit Input*, нажмите кнопку **Добавить** два раза для добавления описаний двух объектов данных, а затем нажмите **Заккрыть**. Два объекта данных типа *1-Bit Input* будут отображены в таблице **Объекты**, как показано на рисунке 44.

Параметры запроса

Тип доступа: Read Discrete Inputs (02h) Тип передачи: 1

Адрес чтения: 33 Использовано слотов чтения: 2 Доступно слотов чтения: 1998

Адрес записи: 1 Использовано слотов записи: 0 Доступно слотов записи: 0

Объекты Чтение Запись

Тип объекта	Имя переменной	Адрес IEC	Смещение	Размер	Комментарий
1-Bit Input		%IX124.0	0	1	
1-Bit Input		%IX124.1	1	1	

Рисунок 44 – Параметры запроса чтения битовых полей типа Discrete Input

26. Для создания конфигурации второго подчиненного узла, полностью идентичной первому, за исключением IP-адреса, выберите элемент *Device – Modbus – Master (Master Modbus TCP) – Slave_1_1* в дереве проекта, нажмите сочетание клавиш Ctrl-C (Ctrl-Insert) или выполните команду меню **Правка – Копировать**, а затем, для вставки копии объекта в дерево проекта, нажмите сочетание клавиш Ctrl-V или выполните команду **Правка – Вставить** в главном меню среды разработки. Копия элемента *Slave_1_1* будет вставлена в конфигурацию мастера MODBUS TCP над оригиналом, при этом имена всех объектов копии автоматически получают суффикс *_1*, чтобы не было совпадения имен копий объектов с именами оригиналов.

27. При необходимости переименуйте объекты копии, как показано на рисунке 45.

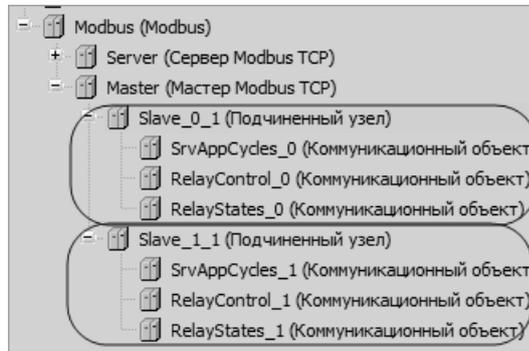


Рисунок 45 – Описание двух подчиненных узлов в конфигурации мастера MODBUS TCP

Для переименования можно дважды с паузой щелкнуть левой кнопкой мыши над элементом в дереве проекта или в контекстном меню над элементом выбрать команду **Свойства** и изменить имя в окне свойств.

28. Измените IP-адрес подчиненного узла *Slave_0_1* так, чтобы он совпадал с IP-адресом первого сетевого интерфейса контроллера.

Если на первых восьми переключателях контроллера установлено значение 3, что соответствует статическому назначению адресов (см. рисунок 5), то введите IP-адрес 10.0.0.3, как показано на рисунке 46.

Если на первых восьми переключателях установлено значение 254, что соответствует динамическому назначению адресов, то введите IP-адрес первого интерфейса (LAN1), выясненный при выполнении команды оболочки ПЛИК (**PLC Shell**) *ipinfo* в соответствии с указаниями п. 3.2.3.

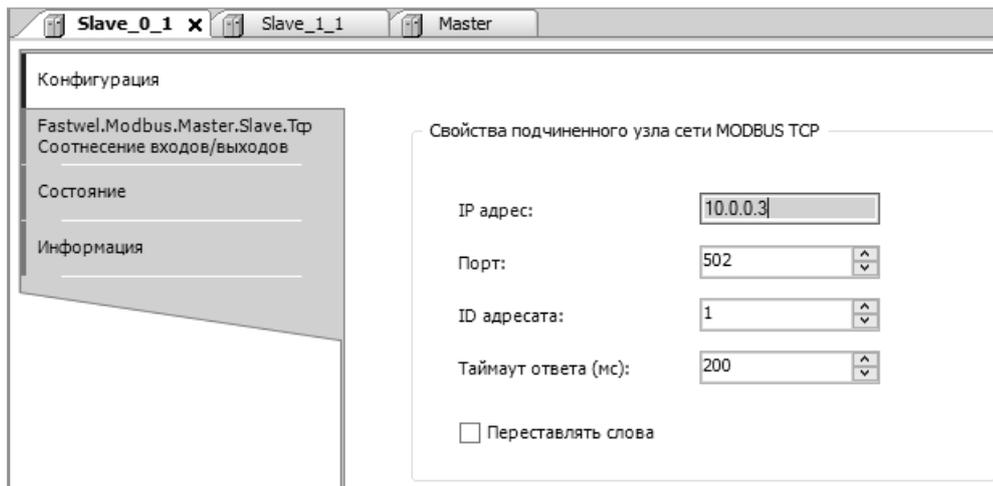


Рисунок 46 – Конфигурация второго подчиненного узла MODBUS TCP

29. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш **Ctrl-S**.

3.7.3. Создание связи приложения с коммуникационными объектами мастера MODBUS TCP

Соотнесение переменных приложения с коммуникационными объектами мастера MODBUS TCP будет произведено после разработки программного кода алгоритма, реализующего функции удаленного управления на мастере. Способ соотнесения аналогичен описанному в п. 3.6.4 для связывания переменных с коммуникационными объектами сервера MODBUS TCP.

3.8. Разработка программы

3.8.1. Общие сведения

Приложение контроллера будет разработано на языке ST (Structured Text) стандарта ГОСТ Р МЭК 61131-3, поскольку данный язык обладает наиболее выразительными средствами для реализации алгоритмов сбора данных и управления, а также системных сервисов и библиотек. Информация об особенностях реализации стандарта в IDE МЭК 61131-3 приведена в интерактивной справочной системе среды разработки в разделе **Руководство по программированию**.

Прикладной алгоритм, реализуемый пользователем в приложении, представляется множеством программ (программных единиц типа *PROGRAM*), которые, в свою очередь, могут содержать множество вызовов других программ, функций (программных единиц типа *FUNCTION*) и экземпляров функциональных блоков (программных единиц типа *FUNCTION BLOCK*), реализованных на языках ST, CFC, SFC, LD, FBD, IL стандарта ГОСТ Р МЭК 61131-3.

Программная единица при вызове получает входные данные и данные окружения, передаваемые в нее через ссылки на входной образ процесса, обрабатывает их, в том числе с учетом своего внутреннего состояния, и передает результаты в виде выходных данных другим программным единицам или в окружение через ссылки на выходного образ процесса.

Входные данные программы, функционального блока и функции представляются так называемыми входными переменными, которые объявляются в секции *VAR_INPUT* области декларации переменных редактора программных единиц.

Выходные данные программы, функционального блока и функции представляются так называемыми выходными переменными, которые объявляются в секции *VAR_OUTPUT* области декларации переменных редактора программных единиц.

Внутреннее состояние программной единицы представляется так называемыми внутренними переменными, которые объявляются в секции *VAR* области декларации переменных редактора программ.

Между программой, функциональным блоком и функцией имеются следующие основные отличия:

1. Программа и экземпляр функционального блока, в отличие от функции, сохраняют свое состояние между вызовами. Иными словами, внутренние, входные и выходные переменные программы и экземпляра функционального блока сохраняют свои значения после очередного вызова и до следующего. Время жизни входных, выходных и внутренних (локальных) переменных функции ограничено интервалом между вызовом и возвратом результата вызова.
2. Программа отличается от функционального блока тем, что в приложении может присутствовать единственный экземпляр программы с некоторым именем, в то время, как функциональный блок с некоторым именем может использоваться для декларации и последующего вызова множества переменных, тип которых совпадает с именем функционального блока.

Таким образом, программная единица типа *PROGRAM* с некоторым именем (например, *PROG_NAME*), созданная пользователем, по сути определяет пользовательский тип данных (*PROG_NAME*) и единственный экземпляр ("переменную") этого типа, и имя экземпляра совпадает с именем типа. Программа может вызываться автоматически из корневой программной единицы некоторой задачи, с которой она явно ассоциирована в редакторе задач **Конфигурация задач**. Кроме того, программу можно вызвать явно по имени из другой программной единицы с передачей входных переменных и получением выходных переменных.

Программная единица типа *FUNCTION_BLOCK* с некоторым именем (например, *FB_NAME*) определяет пользовательский тип данных с этим именем (*FB_NAME*), а экземпляры данного типа создаются пользователем в виде внутренних переменных программ или других блоков. Вызов экземпляра функционального блока осуществляется по имени соответствующей переменной в теле содержащей его программы или блока, причем при вызове можно передавать входные и получать выходные переменные.

Информация о программной модели приложения приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Приложение контроллера в учебном проекте будет состоять из двух независимых алгоритмов, выполняющихся под управлением отдельных циклических задач:

1. Алгоритм сервера MODBUS TCP: считывает состояние блока переключателей, включает и выключает первый канал модуля DIM713 в зависимости от положения переключателя 10, принимает сетевую команду включения и выключения второго канала модуля DIM713, передает в сеть текущее состояние двух каналов модуля DIM713 и управляет индикатором USER.
2. Алгоритм клиента MODBUS TCP: выполняет удаленное управление вторым каналом модуля DIM713 по сети и на уровне приложения обеспечивает механизм дублирования обмена данными с сервером MODBUS TCP.

Алгоритм сервера будет реализован в программной единице типа *PROGRAM* с именем *PLC_PRG*, выполняемой под управлением циклической задачи *MainTask*.

Алгоритм клиента будет реализован в программной единице типа *PROGRAM* с именем *TCP_CLIENT*, выполняемой под управлением циклической задачи *ClientTask*.

Кроме того, в приложении будет создан и использоваться функциональный блок *ACTIVITY_CHECKER*. Данный блок предназначен для диагностики выполнения вычислительного процесса циклической природы, в котором для контроля факта выполнения используется счетчик циклов, а для контроля правильности выполнения – счетчик ошибок. Данный функциональный блок будет использоваться в алгоритмах сервера и клиента.

3.8.2. Функциональный блок *ACTIVITY_CHECKER*

3.8.2.1. Общие сведения

Циклическая природа алгоритмов сбора данных и управления и вычислительной модели ГОСТ Р МЭК 61131-3 позволяет использовать счетную переменную для диагностики активности локального или удаленного алгоритма, сетевого сервиса или сервиса ввода-вывода, которым принадлежит переменная. Для реализации такого способа диагностики достаточно определить интервал времени, на котором должно произойти изменение значения контролируемой счетной переменной, и проверять приращение значения на этом интервале. При необходимости диагностики ошибок достаточно на том же интервале анализа проверять приращение второй счетной переменной, содержащей количество ошибок контролируемой сущности (алгоритма или сервиса).

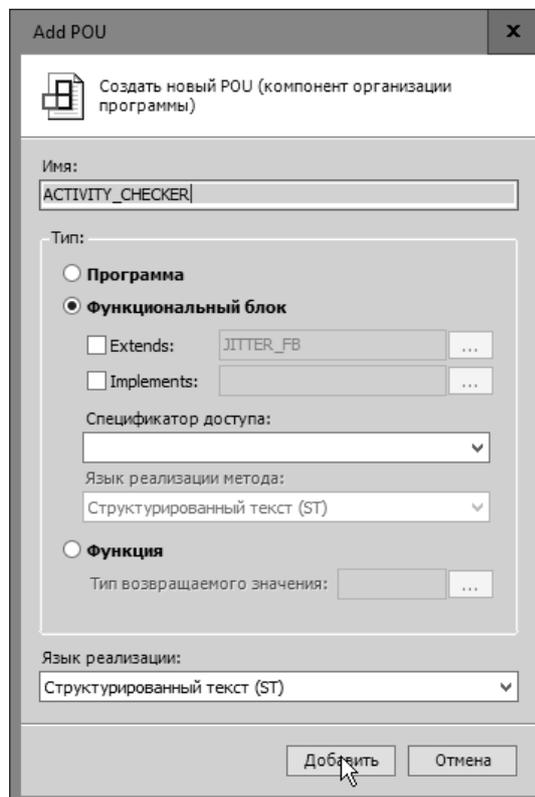


Рисунок 47 – Создание функционального блока *ACTIVITY_CHECKER*

3.8.2.2. Создание функционального блока (FUNCTION_BLOCK)

Для создания функционального блока:

1. Щелкните правой кнопкой мыши над элементом *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application* и выполните команду **Add Object – POU** в контекстном меню.
2. В окне **Add POU** в поле **Тип** выберите тип создаваемой программной единицы **Функциональный блок**, в поле **Имя** введите имя *ACTIVITY_CHECKER*, в выпадающем списке **Язык реализации** выберите тип *Структурированный текст (ST)* и нажмите кнопку **Добавить**, как показано на рисунке 47.

В правой области главного окна среды разработки будет открыта вкладка редактора программных единиц с заготовками деклараций входных, выходных и внутренних переменных блока *ACTIVITY_CHECKER*.

3. В области декларации переменных введите декларации следующих переменных и комментарии:

```
(*
  Блок проверки активности по счетчику операций ActivityCounter.
  Active устанавливается в TRUE, если за время CheckPeriod счетчик
  ActivityCounter изменился хотя бы на 1.
  Error устанавливается в TRUE, если за время CheckPeriod счетчик
  ErrorsCounter изменился на ту же величину, что и ActivityCounter.
*)
```

```
FUNCTION_BLOCK ACTIVITY_CHECKER
```

```
VAR_INPUT
```

```
(* Счетчик активности *)
ActivityCounter : DWORD;
```

```
(* Счетчик ошибок *)
ErrorsCounter : DWORD;
```

```
(* Интервал проверки активности *)
CheckPeriod : TIME := T#500MS;
```

```
(* Сброс состояния блока (по переднему фронту) *)
Reset : BOOL;
```

```
END_VAR
```

```
VAR_OUTPUT
```

```
(* Количество единиц, на которое изменяется ActivityCounter за секунду *)
Performance : REAL;
```

```
(* Признак наличия активности *)
Active : BOOL;
```

```
(* Признак ошибки *)
Error : BOOL;
```

```
END_VAR
```

```
VAR
```

```
act_count_prev : DWORD;
err_count_prev : DWORD;
reset_prev : BOOL;
next_time_to_check : TIME;
```

```
END_VAR
```

```
VAR_TEMP
```

```
current_time : TIME;
time_delta : TIME;
err_delta : DWORD;
act_delta : DWORD;
```

```
END_VAR
```

Переменные *VAR_TEMP* являются временными, т.е. их значения не сохраняются между вызовами экземпляра блока. Переменные *VAR* относятся к алгоритму, реализуемому блоком, и представляют внутреннее состояние блока, сохраняемое между вызовами. Формально внутренние переменные блока недоступны прочим программным единицам приложения.

4. В области реализации (исходного текста алгоритма, реализуемого блоком) введите следующий исходный текст:

```
(* Берем текущее время (счетчик миллисекунд от запуска контроллера) *)
current_time := TIME();

IF (NOT reset_prev AND Reset) OR (next_time_to_check = T#0MS) THEN
  (* Обработка команды сброса по переднему фронту Reset или первого цикла блока *)
  Performance := 0.0;
  Active := FALSE;
  Error := FALSE;
  act_count_prev := ActivityCounter;
  err_count_prev := ErrorsCounter;
  next_time_to_check := current_time + CheckPeriod;
  reset_prev := Reset;
  RETURN;
END_IF
reset_prev := Reset;

(* Определяем кол-во мс между текущим моментом и запланированным *)
(* моментом проверки активности *)
time_delta := current_time - next_time_to_check;

IF TIME_TO_DINT(time_delta) >= 0 THEN
  (* Пришло время проверки наличия активности *)

  (* Вычисляем приращения счетчиков ActivityCounter и ErrorsCounter *)
  err_delta := ErrorsCounter - err_count_prev;
  act_delta := ActivityCounter - act_count_prev;

  (* Устанавливаем признак ошибки, если есть ненулевое приращение счетчика *)
  (* ошибок, и это приращение совпадает с приращением ActivityCounter. *)
  Error := err_delta <> 0 AND err_delta = act_delta;

  (* Устанавливаем признак наличия активности, если есть ненулевое приращение*)
  (* счетчика ActivityCounter. *)
  Active := act_delta <> 0;

  (* Вычисляем производительность, если нет признака ошибки. *)
  Performance := SEL (Error OR CheckPeriod = T#0MS,
    DWORD_TO_REAL(1000 * act_delta) / TIME_TO_REAL(CheckPeriod + time_delta), 0.0);

  (* Запоминаем значения счетчиков для следующего момента проверки *)
  act_count_prev := ActivityCounter;
  err_count_prev := ErrorsCounter;

  (* Вычисляем момент времени для следующей проверки *)
  next_time_to_check := current_time + CheckPeriod;
END_IF
```

Таким образом, при первом вызове экземпляра функционального блока или при обнаружении переднего фронта на входе *Reset* выходные и внутренние переменные блока принимают исходные значения, а также определяется момент времени *next_time_to_check*, в который будет принято решение о наличии активности по счетчику *ActivityCounter* и наличии ошибок по счетчику *ErrorsCounter*.

Затем, при последующих вызовах экземпляра блока, вычисляется разность между текущим временем в миллисекундах от момента запуска контроллера *current_time* и временем принятия решения *next_time_to_check*.

Если значение разности, приведенное к знаковому 32-разрядному типу DINT, больше либо равно 0, то наступил очередной момент принятия решения о наличии активности по счетчику *ActivityCounter* и ошибок по счетчику *ErrorsCounter*.

Если за время *CheckPeriod* с момента последнего вызова экземпляра блока счетчик ошибок *ErrorsCounter* имеет ненулевое приращение, и это приращение по величине совпадает с приращением счетчика *ActivityCounter*, то устанавливается признак ошибки *Error*.

Если за время *CheckPeriod* с момента последнего вызова экземпляра блока счетчик активности *ActivityCounter* имеет ненулевое приращение, то устанавливается признак наличия активности *Active*.

Условная производительность контролируемой активности *Performance*, выраженная в приращении *ActivityCounter* за секунду, вычисляется при сброшенном признаке ошибки *Error* и при ненулевом интервале анализа *CheckPeriod*. Обратите внимание, что при вычислении *Performance* учитывается, возможно ненулевая, разница во времени между определенным ранее моментом времени принятия решения и текущим моментом времени *time_delta*.

На последнем шаге алгоритма, относящемся к моменту принятия решения о наличии активности и ошибки, определяется новый момент времени принятия решения путем сложения *CheckPeriod* со значением *current_time*. Формально, для обеспечения стабильности выдержки интервала анализа при вычислении *next_time_to_check* также должна учитываться величина *time_delta* путем вычитания из результата сложения *CheckPeriod* со значением *current_time*:

```
(* Вычисляем момент времени для следующей проверки. *)
(* time_delta не нужно приводить к DINT, т.к. ее значение в этом месте *)
(* равно 0 или почти равно 0. *)
next_time_to_check := current_time + CheckPeriod - time_delta;
```

3.8.2.3. Использование функционального блока

Функциональный блок *ACTIVITY_CHECKER* может использоваться в других программных единицах следующими способами:

1. Объявление переменной типа *ACTIVITY_CHECKER* в области деклараций других функциональных блоков или программ и вызов именованных экземпляров блоков:

```
PROGRAM PLC_PRG
VAR
  dwIoTxCounter AT %ID44 : DWORD;
  dwIoErrorCounter AT %ID48 : DWORD;
  act_1 : ACTIVITY_CHECKER;
END_VAR

  act_1( ActivityCounter := dwIoTxCounter,
        ErrorsCounter := dwIoErrorCounter,
        CheckPeriod := T#1000MS,
        Reset :=,
        Performance=>,
        Active=>,
        Error=>);
END_PROGRAM
```

2. Передача указателя на экземпляр функционального блока в качестве параметра функции и вызов блока путем "разыменования" указателя:

```
FUNCTION CallChecker : BOOL
VAR_INPUT
  dwCounter : DWORD;
  pAct : POINTER TO ACTIVITY_CHECKER;
END_VAR
  pAct^(ActivityCounter := dwCounter, CheckPeriod := T#200ms, Active=>CallChecker);
END_FUNCTION
```

```
PROGRAM PLC_PRG
VAR
  dwIoTxCounter AT %ID44 : DWORD;
  bActive : BOOL;
  act_1 : ACTIVITY_CHECKER;
END_VAR
  bActive := CallChecker(dwCounter:= dwIoTxCounter, pAct:= ADR(act_1));
END_PROGRAM
```

3. Наследование интерфейса и реализации функционального блока:

```
(* Бессмысленное применение наследования *)
FUNCTION_BLOCK SPECIFIC_CHECKER EXTENDS ACTIVITY_CHECKER
VAR_OUTPUT
  wStateMask : WORD;
END_VAR
  SUPER^(CheckPeriod := T#1000ms, Active=> wStateMask.0, Error=> wStateMask.1);
END_FUNCTION_BLOCK
```

```

PROGRAM PLC_PRG
VAR
  dwIoTxCounter AT %IB1004 : DWORD;
  dwIoErrorCounter AT %IB1008 : DWORD;
  wMask : WORD;
  act_1 : SPECIFIC_CHECKER;
END_VAR

  act_1 ( ActivityCounter := dwIoTxCounter,
          ErrorsCounter := dwIoErrorCounter,
          CheckPeriod := T#1000MS,
          wStateMask=> wMask);
END_PROGRAM

```

Пример более приемлемого использования механизма наследования имеется в проекте TasksExchange.project, поставляемого в составе Fastwel PLC Application Toolkit.

3.8.3. Алгоритм сервера MODBUS TCP

3.8.3.1. Общие сведения

Согласно п. 3.2.1, приложение должно обеспечивать выполнение следующих функций:

1. Прерывистое свечение индикатора USER контроллера красным или зеленым цветом. Красный цвет будет использоваться при отсутствии запросов к серверу MODBUS TCP от удаленных клиентов, а зеленый – при периодическом поступлении запросов к серверу.
2. Включение и выключение первого канала модуля реле DIM713 при включении и выключении 10-го переключателя.
3. Включение и выключение второго канала модуля реле DIM713 по команде, поступающей от удаленного клиента протокола MODBUS TCP.
4. Включение и выключение второго канала модуля реле DIM713 по сети MODBUS TCP из другой циклической задачи этого же приложения.

Первые три функциональных требования будут выполняться программой *PLC_PRG*, реализующей прикладной алгоритм на подчиненном узле (сервере) MODBUS TCP.

Для чтения состояния блока переключателей контроллера будет использоваться экземпляр функционального блока *SWITCH_READER_x10* из библиотеки FastwelBoard.

Для управления индикатором USER будет применен функциональный блок *USER_LED_CONTROL* из библиотеки FastwelBoard.

Для контроля наличия обмена с клиентами MODBUS TCP будет использоваться экземпляр функционального блока *ACTIVITY_CHECKER*, разработанного при выполнении указаний п. 3.8.2 настоящего документа. Соотнесение входных переменных функционального блока со счетчиками транзакций и ошибок сервера MODBUS TCP будет выполнено непосредственно в редакторе образа процесса.

Управление каналами модуля DIM713 и получение текущего состояния каналов модуля выполняется при помощи экземпляра функционального блока *DIM7DO_SIMPLE_STIN* из библиотеки FastwelFbusIO, который был объявлен в программе *PLC_PRG* при выполнении указаний п. 3.5.5 настоящего документа.

3.8.3.2. Декларация переменных PLC_PRG

Откройте программу *PLC_PRG* в редакторе программных единиц и введите следующие декларации переменных и констант в области деклараций:

```

// Основной алгоритм приложения
PROGRAM PLC_PRG
VAR CONSTANT
  // Период контроля состояния переключателей
  SWITCHES_UPDATE_PERIOD : TIME := T#100MS;
  // Период "мигания" USER
  USER_BLINK_PERIOD : TIME := T#1000MS;
  // Период проверки наличия сетевого обмена с клиентом MODBUS TCP
  NET_CHECK_PERIOD : TIME := T#100MS;
END_VAR

```

```

VAR
  // Экземпляр блока управления модулем дискретного вывода
  m_do_1 : FastwelfbusIO.DIM7DO_SIMPLE_STIN;
  // Счетчик циклов PLC_PRG
  dwCycleCounter : DWORD;
  // Блок проверки наличия сетевого обмена с клиентом MODBUS TCP
  net_act_checker : ACTIVITY_CHECKER;
  // Блок чтения состояния переключателей
  sw_reader : FastwelfBoard.SWITCH_READER_x10;
  // Цвет индикатора USER
  led_color : FastwelfBoard.LedsColor_t := FastwelfBoard.LED_COLOR_RED;
  // Блок "мигания" индикатором USER
  led_blinker : FastwelfBoard.USER_LED_CONTROL;
END_VAR

```

Обратите внимание на возможность ввода однострочных комментариев с использованием префикса строки //.

Кроме того, обратите внимание на синтаксис декларации переменных, определенных в библиотеках: перед именем типа указывается имя библиотеки и точка. Такой синтаксис позволяет избежать ошибок компиляции при совпадении имен типов, определенных в разных библиотеках и программных единицах.

3.8.3.3. Исходный текст алгоритма PLC_PRG

В области реализации программы *PLC_PRG* введите следующий исходный текст:

```

// Увеличиваем счетчик циклов алгоритма
dwCycleCounter := dwCycleCounter + 1;

// Вызываем блок чтения положения переключателей
sw_reader(Period := SWITCHES_UPDATE_PERIOD);

// Вызываем блок проверки наличия сетевого обмена с клиентом MODBUS TCP
net_act_checker(CheckPeriod := NET_CHECK_PERIOD);

// Если нет обмена или все транзакции закончились ошибкой, используем
// красный цвет индикатора USER.
// При наличии обмена с клиентами MODBUS TCP используем зеленый цвет.
led_color := SEL(net_act_checker.Error OR NOT net_act_checker.Active,
  FastwelfBoard.LED_COLOR_GREEN, FastwelfBoard.LED_COLOR_RED);

// Если успешно прочитали положение переключателей, то состояние первого
// канала DIM713 определяется состоянием переключателя 10
m_do_1.Controls[1] := NOT sw_reader.Failure AND sw_reader.SwitchStates[10];

// Вызываем блок управления модулем DIM713
m_do_1();

// Вызываем блок прерывистого свечения индикатора USER
led_blinker(Color := led_color, BlinkPeriod := USER_BLINK_PERIOD);

```

Обратите внимание, что комментарии в исходном тексте являются наилучшим способом документирования замысла автора программы.

Полный исходный текст программы *PLC_PRG* в редакторе программных единиц представлен на рисунке 48. Обратите внимание на возможность "сворачивания" и "разворачивания" отдельных фрагментов исходного текста щелчком мышью на элементах управления  и  в области отображения номеров строк.

```

1 // Основной алгоритм приложения
2 PROGRAM PLC_PRG
3
4 VAR CONSTANT
5 // Период контроля состояния переключателей
6 SWITCHES_UPDATE_PERIOD : TIME := T#100MS;
7 // Период "мигания" USER
8 USER_BLINK_PERIOD : TIME := T#1000MS;
9 // Период проверки наличия сетевого обмена с клиентом MODBUS TCP
10 NET_CHECK_PERIOD : TIME := T#100MS;
11 END_VAR
12
13 VAR
14 // Экземпляр блока управления модулем дискретного вывода
15 m_do_1 : FastwelFbusIO.DIM7DO_SIMPLE_STIN;
16 // Счетчик циклов PLC_PRG
17 dwCycleCounter : DWORD;
18 // Блок проверки наличия сетевого обмена с клиентом MODBUS TCP
19 net_act_checker : ACTIVITY_CHECKER;
20 // Блок чтения состояния переключателей
21 sw_reader : FastwelBoard.SWITCH_READER_x10;
22 // Цвет индикатора USER
23 led_color : FastwelBoard.LedsColor_t := FastwelBoard.LED_COLOR_RED;
24 // Блок "мигания" индикатором USER
25 led_blinker : FastwelBoard.USER_LED_CONTROL;
26 END_VAR

```

```

1 // Увеличиваем счетчик циклов алгоритма
2 dwCycleCounter := dwCycleCounter + 1;
3
4 // Вызываем блок чтения положения переключателей
5 sw_reader(Period := SWITCHES_UPDATE_PERIOD);
6
7 // Вызываем блок проверки наличия сетевого обмена с клиентом MODBUS TCP
8 net_act_checker(CheckPeriod := NET_CHECK_PERIOD);
9
10 // Если нет обмена или все транзакции закончились ошибкой, используем
11 // красный цвет индикатора USER.
12 // При наличии обмена с клиентами MODBUS TCP используем зеленый цвет.
13 led_color := SEL(net_act_checker.Error OR NOT net_act_checker.Active,
14                 FastwelBoard.LED_COLOR_GREEN, FastwelBoard.LED_COLOR_RED);
15
16 // Если успешно прочитали положение переключателей, то состояние первого
17 // канала DIM713 определяется состоянием переключателя 10
18 m_do_1.Controls[1] := NOT sw_reader.Failure AND sw_reader.SwitchStates[10];
19
20 // Вызываем блок управления модулем DIM713
21 m_do_1();
22
23 // Вызываем блок прерывистого свечения индикатора USER
24 led_blinker(Color := led_color, BlinkPeriod := USER_BLINK_PERIOD);
25

```

Рисунок 48 – Исходный текст PLC_PRG. Элементы управления для скрытия отдельных фрагментов исходного текста

3.8.3.4. Соотнесение счетчиков транзакций и ошибок сервера MODBUS TCP

Для соотнесения входных переменных *ActivityCounter* и *ErrorsCounter* функционального блока *PLC_PRG.net_act_checker* со счетчиками транзакций и ошибок сервера MODBUS TCP:

- Щелкните правой кнопкой мыши над элементом *Device – Modbus – Server* в дереве проекта и выберите команду **Изменить I/O-соотнесение**.
- На вкладке редактора образа процесса дважды щелкните левой кнопкой мыши в ячейке *Server – Transaction count*, а затем нажмите кнопку , как показано на рисунке 49.

Переменная	Канал	Адрес	Тип
Server			
	Transactions count	%ID44	DWORD
	Errors count	%ID48	DWORD
	Control	%QD4	DWORD

Рисунок 49 – Соотнесение со счетчиком сетевых транзакций MODBUS TCP

- В окне **Ассистент ввода** выберите переменную *Application.PLC_PRG.net_act_checker.ActivityCounter*, как показано на рисунке 50, и нажмите **ОК**, а затем, после закрытия окна **Ассистент ввода**, нажмите клавишу **Enter**.

Имя выбранной переменной будет отображено в ячейке *Server – Transactions count* на вкладке редактора образа процесса.

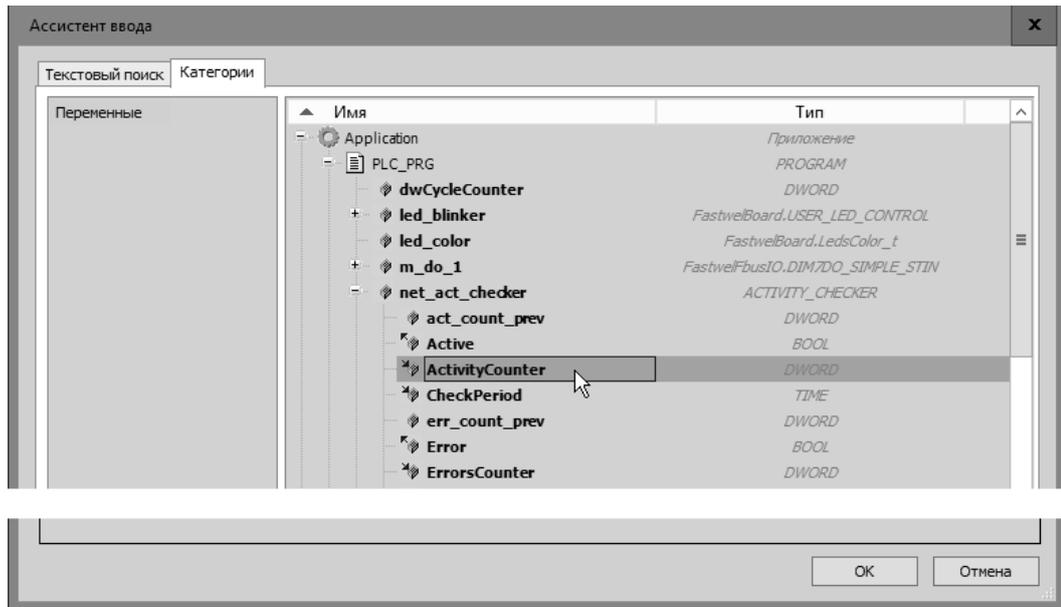


Рисунок 50 – Соотнесение переменной *Application.PLC_PRG.net_act_checker.ActivityCounter*

4. Таким же образом, как указано выше в описании шагов 1–3, выполните соотнесение канала *Server – Errors count* с переменной *Application.PLC_PRG.net_act_checker.ErrorsCounter*. Содержимое вкладки редактора образа процесса после выполнения соотнесения показано на рисунке 51.

Переменная	Канал	Адрес	Тип
Application.PLC_PRG.net_act_checker.ActivityCounter	Transactions count	%ID44	DWORD
Application.PLC_PRG.net_act_checker.ErrorsCounter	Errors count	%ID48	DWORD
	Control	%QD4	DWORD
Inputs			
Outputs			

Рисунок 51 – Входные переменные блока *Application.PLC_PRG.net_act_checker*, отображенные на счетчики транзакций и ошибок сервера MODBUS TCP

5. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш **Ctrl-S**.

3.8.4. Алгоритм мастера MODBUS TCP

3.8.4.1. Общие сведения

Включение и выключение второго канала модуля реле DIM713 по сети MODBUS TCP будет выполняться программой *TCP_CLIENT*, вызываемой из циклической задачи *ClientTask*. При этом программа *TCP_CLIENT* должна реализовывать механизм дублирования связи с подчиненным узлом MODBUS TCP на прикладном уровне, как указано в п. 3.7.1.

3.8.4.2. Создание программы TCP_CLIENT

Для добавления программной единицы типа *PROGRAM* в приложение:

1. Щелкните правой кнопкой мыши над элементом *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application* и выполните команду **Add Object – POU** в контекстном меню.
2. В поле **Тип** окна **Add POU** выберите тип создаваемой программной единицы **Программа**, в поле **Имя** введите имя *TCP_CLIENT*, в выпадающем списке **Язык реализации** выберите *Структурированный текст (ST)* и нажмите кнопку **Добавить**, как показано на рисунке 52.

В дереве проекта появится элемент *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – TCP_CLIENT (PRG)*, а в правой области главного

окна среды разработки будет открыта вкладка редактора программных единиц с пустой реализацией программы *TCP_CLIENT*.

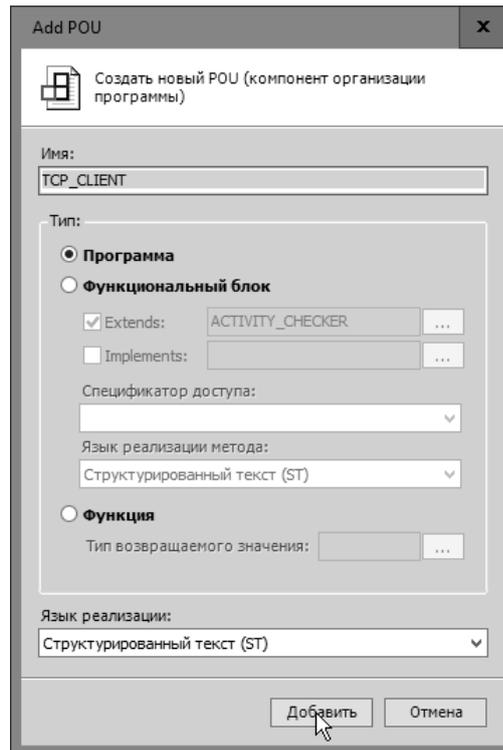


Рисунок 52 – Создание программной единицы типа PROGRAM

3. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки.

3.8.4.3. Создание задачи *ClientTask* и установка вызова *TCP_CLIENT*

Для создания описания задачи *ClientTask* и добавления программы *TCP_CLIENT* в ее список исполнения:

1. Щелкните правой кнопкой мыши над элементом *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – Конфигурация задач* и выполните команду **Add Object – Task** в контекстном меню.
2. В появившемся окне **Add Task** в поле **Имя** введите имя задачи *ClientTask* и нажмите **Добавить**, как показано на рисунке 53.

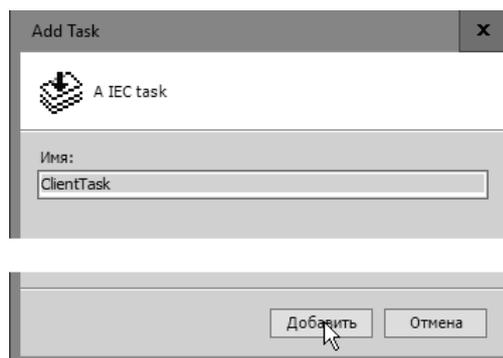


Рисунок 53 – Создание задачи *ClientTask*

В дереве проекта появится элемент *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – Конфигурация задач – ClientTask*, а в правой области главного окна среды разработки будет отображена вкладка редактора конфигурации задачи, показанная на рисунке 54.

3. На вкладке конфигурации задачи нажмите кнопку **Добавить вызов** и в окне **Ассистент ввода** выберите программу *Application – TCP_CLIENT*, после чего нажмите **ОК**. Имя программы *TCP_CLIENT* будет отображено в списке вызываемых программ задачи *ClientTask*.
4. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки.

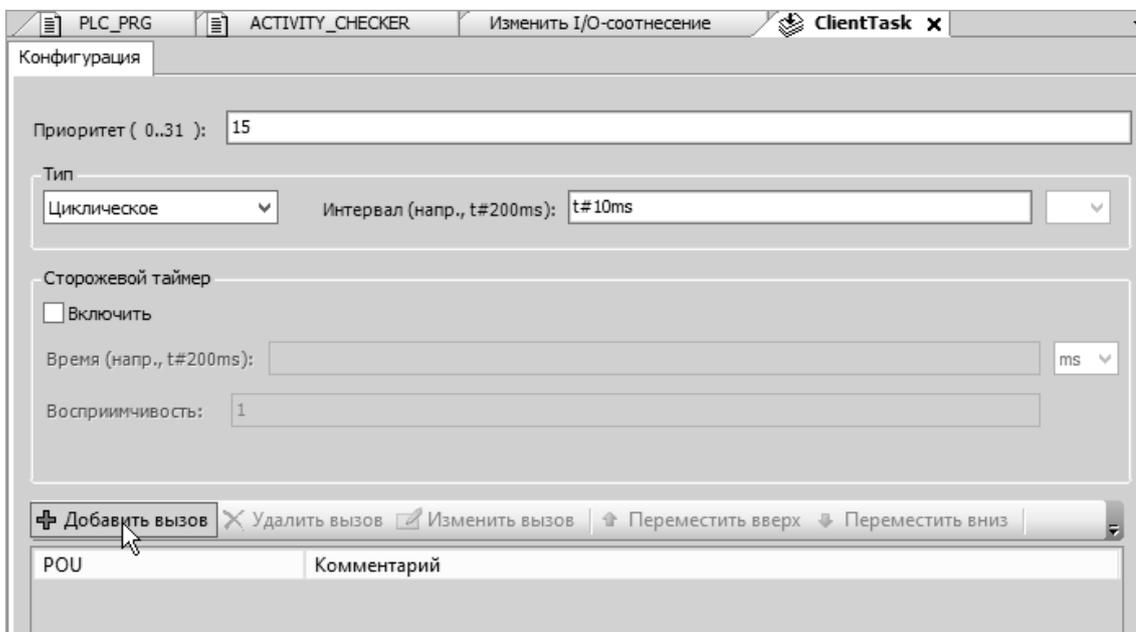


Рисунок 54 – Добавление программы в список вызовов задачи ClientTask

Редактор конфигурации задачи позволяет выбрать тип задачи, установить приоритет и период исполнения задачи, включить и настроить сторожевой таймер выполнения задачи и определить список программ, выполняющихся под управлением задачи.

Обратите внимание, что меньшему абсолютному значению приоритета задачи, задаваемому в поле **Приоритет**, соответствует более высокий приоритет задачи у планировщика. Более подробная информация о задачах приведена в руководстве пользователя Fastwel PLC Application Toolkit.

В учебном проекте используется исходная конфигурация задачи *ClientTask*.

3.8.4.4. Декларация переменных TCP_CLIENT

Откройте программу *TCP_CLIENT* в редакторе программных единиц и объявите следующие переменные и константы в области деклараций:

```
// Алгоритм клиента MODBUS TCP, реализующий дублирование канала
// управления через переменную app_control_cmd на прикладном уровне.
PROGRAM TCP_CLIENT
VAR CONSTANT
  // Период проверки наличия отказа связи на уровне протокола
  NET_CHECK_PERIOD : TIME := T#500MS;
  // Период проверки наличия отказа канала на прикладном уровне
  APP_CHECK_PERIOD : TIME := T#100MS;
END_VAR

VAR
  // Переменные прикладной части алгоритма:
  // - команда управления 2-м каналом модуля реле
  app_control_cmd : BOOL;
  // - текущее состояние 1-го канала модуля реле
  app_relay1_state : BOOL;
  // - текущее состояние 2-го канала модуля реле
  app_relay2_state : BOOL;
  // Переменные диагностики:
  // - признак отказа связи по основному каналу
  app_slave1_failure : BOOL;
  // - признак отказа связи по резервному каналу
  app_slave2_failure : BOOL;
  // - признак отказа связи по двум каналам
  common_cause_failure : BOOL;

  // Переменные обеспечения дублированного канала:
  // 1. Переменные основного канала:
  // - блок проверки наличия связи на уровне протокола
  slave1_comm_act : ACTIVITY_CHECKER;
  // - блок проверки наличия связи на уровне приложения
  slave1_app_act : ACTIVITY_CHECKER;
```

```

// - команда управления по основному каналу
slave1_app_control_cmd : BOOL;
// - состояние 1-го канала модуля реле по основному каналу
slave1_app_relay1_state : BOOL;
// - состояние 2-го канала модуля реле по основному каналу
slave1_app_relay2_state : BOOL;

// 2. Переменные резервного канала:
// - блок проверки наличия связи на уровне протокола
slave2_comm_act : ACTIVITY_CHECKER;
// - блок проверки наличия связи на уровне приложения
slave2_app_act : ACTIVITY_CHECKER;
// - команда управления по резервному каналу
slave2_app_control_cmd : BOOL;
// - состояние 1-го канала модуля реле по резервному каналу
slave2_app_relay1_state : BOOL;
// - состояние 2-го канала модуля реле по резервному каналу
slave2_app_relay2_state : BOOL;
END_VAR

```

Переменные *app_control_cmd*, *app_relay1_state* и *app_relay2_state* определяют состояние прикладного алгоритма, реализуемого программой *TCP_CLIENT*, а переменные с префиксом имени *slave1_* и *slave2_* представляют механизм дублирования связи программы с подчиненным узлом MODBUS TCP на прикладном уровне.

3.8.4.5. Исходные текст TCP_CLIENT

В области реализации программы *TCP_CLIENT* введите следующий исходный текст:

```

// Вызываем блоки проверки связи на уровне протокола
slave1_comm_act(CheckPeriod := NET_CHECK_PERIOD);
slave2_comm_act(CheckPeriod := NET_CHECK_PERIOD);
// Вызываем блоки проверки связи на уровне приложения
slave1_app_act(CheckPeriod := APP_CHECK_PERIOD);
slave2_app_act(CheckPeriod := APP_CHECK_PERIOD);

// Устанавливаем признаки отказа основного и резервного каналов.
// Видим, что можно не использовать проверку связи на уровне протокола
app_slave1_failure := NOT slave1_app_act.Active AND slave1_comm_act.Error;
app_slave2_failure := NOT slave2_app_act.Active AND slave2_comm_act.Error;

// Устанавливаем признак отсутствия связи по основному и резервному каналам
common_cause_failure := app_slave1_failure AND app_slave2_failure;

// Команда управления передается в оба канала
slave1_app_control_cmd := app_control_cmd;
slave2_app_control_cmd := app_control_cmd;

// Определяем состояние каналов модуля реле:
// - используем основной канал, если нет отказа по нему;
// - используем резервный канал, если есть отказ по основному
IF NOT app_slave1_failure THEN
  app_relay1_state := slave1_app_relay1_state;
  app_relay2_state := slave1_app_relay2_state;
ELSIF NOT app_slave2_failure THEN
  app_relay1_state := slave2_app_relay1_state;
  app_relay2_state := slave2_app_relay2_state;
END_IF

```

Блоки *slave1_comm_act* и *slave2_comm_act* предназначены для проверки связи с подчиненным узлом на уровне сетевого протокола, а блоки *slave1_app_act* и *slave2_app_act* позволяют проверить связь с алгоритмом на удаленном узле сети на прикладном уровне за счет контроля увеличения значения счетчика циклов, получаемого по сети от двух подчиненных узлов сервера MODBUS TCP.

Из кода *TCP_CLIENT* видно, что контроль связи с подчиненным узлом может выполняться только на прикладном уровне, поскольку при разрыве связи по любому из каналов программа *TCP_CLIENT* по истечении *APP_CHECK_PERIOD* обнаружит отсутствие увеличения счетчика циклов соответствующего подчиненного узла.

3.9. Трансляция приложения и загрузка в контроллер

3.9.1. Общие сведения

IDE МЭК 61131-3 загружает в контроллер исполняемый бинарный код процессора, на базе которого реализовано главное вычислительное устройство контроллера, а также бинарную конфигурацию устройств и сетевых протоколов, описания которых созданы пользователем в проекте.

До загрузки приложения в контроллер возможны три варианта трансляции приложения, доступные в главном меню **Компиляция (Build)** среды разработки:

1. **Компиляция (Build)** – данная команда не генерирует исполняемый код, а выполняет синтаксический разбор, анализ и проверку корректности всех элементов проекта.

Если с момента последней компиляции в приложение не внесено изменений, последующее выполнение команды **Компиляция** не приводит к фактическому выполнению операции, а в панели сообщений среды разработки выводится сообщение "Приложение актуально".

Если в приложение внесены изменения с момента последней компиляции, то при выполнении команды **Компиляция** транслятор предпринимает попытку учета только изменений в проектной информации.

2. **Перекомпиляция (Rebuild)** – данная команда также не генерирует исполняемый код приложения, а выполняет повторный синтаксический разбор, анализ и проверку корректности всех элементов проекта, независимо от наличия изменений в проекте с момента последней компиляции.

3. **Генерировать код (Generate code)** – данная команда выполняет компиляцию, а затем генерирует исполняемый код целевого процессора и другие бинарные данные приложения, готовые к загрузке в контроллер.

Данная операция выполняется автоматически в самом начале загрузки приложения в контроллер при выполнении команды **Онлайн (Online) – Логин (Login)**. При этом, после успешной загрузки приложения в контроллер, в каталоге размещения файла проекта на ПК генерируются служебные файлы с информацией о трансляции, имеющие расширения *compileinfo* и *bootinfo*.

После внесения изменений в исходный текст программных единиц проекта транслятор среды разработки предпримет попытку сгенерировать и загрузить в контроллер бинарный код только измененных программных единиц без останова выполнения приложения (**Online Change** или **Логин с онлайн-изменением**).

В IDE МЭК 61131-3 Astra.IDE и CODESYS V3, начиная с версии 3.5 SP17, доступна только команда **Генерировать код (Generate code)**.

Если при фактическом отсутствии каких-либо ошибок во время работы на проекте в панели сообщений среды разработки появились сообщения об ошибках, можно выполнить команду **Компиляция (Build) – Очистить все (Clean All)**, а затем повторить команду **Генерировать код**. Команда очистки служит для удаления ранее сформированной средой разработки информации о компиляции и о последней загрузке приложения в контроллер, чтобы иметь возможность сгенерировать код приложения "с нуля".

Обратите внимание, что после выполнения команды **Очистить** загрузка изменений в контроллер (**Логин с онлайн-изменением**) становится невозможной, и приложение может быть передано в контроллер только методом полной загрузки.

Отличие команды **Очистить все** от **Очистить** состоит в том, что **Очистить все** удаляет ранее сформированную информацию о компиляции для всех приложений в данном проекте, а команда **Очистить** – только для *активного* приложения.

Большинство операций в главном меню среды разработки выполняются в отношении текущего *активного* приложения. Проект IDE МЭК 61131-3 может содержать приложения для нескольких контроллеров, и в этом случае для трансляции необходимо выбрать требуемое приложение в качестве *активного*, как показано на рисунке 5б.

При работе над проектом, который содержит приложения для нескольких контроллеров, все приложения могут быть транслированы и загружены во все контроллеры одной командой **Онлайн – Множественная загрузка**. Перед выполнением данной команды на вкладке **Установка соединения**

каждого устройства, соответствующего контроллеру, необходимо задать сетевой путь в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.

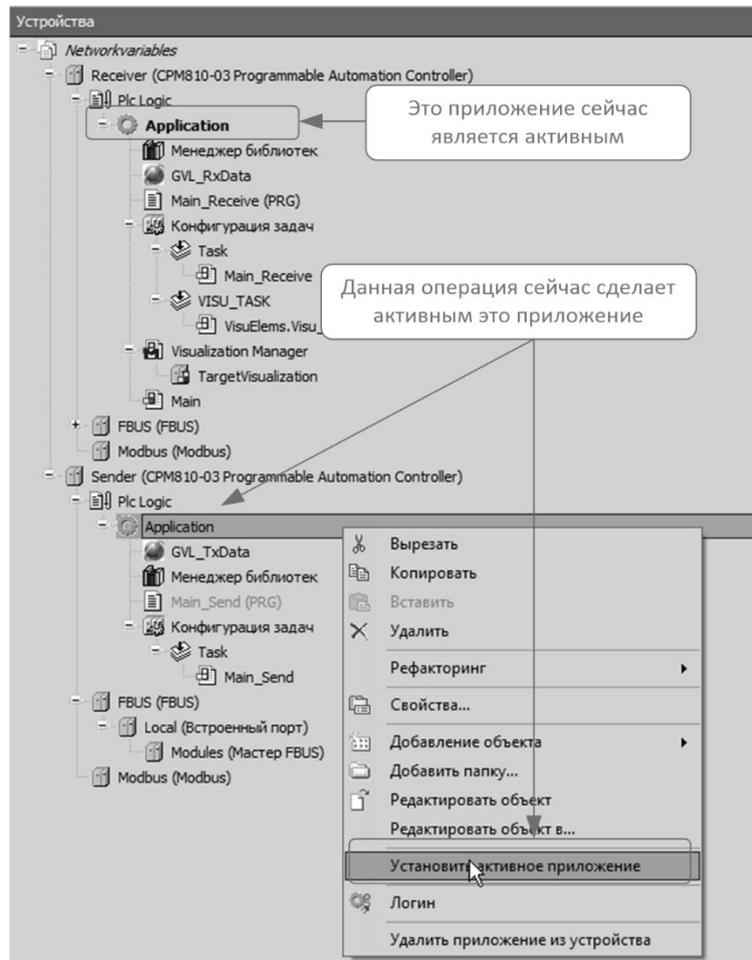


Рисунок 56 – Выбор активного приложения в проекте с приложениями для двух контроллеров

3.9.2. Загрузка приложения учебного проекта в контроллер

1. В главном меню среды разработки выполните команду **Компиляция – Генерировать код** и убедитесь, что в панели сообщений отсутствует информация о зафиксированных ошибках, как показано на рисунке 57.

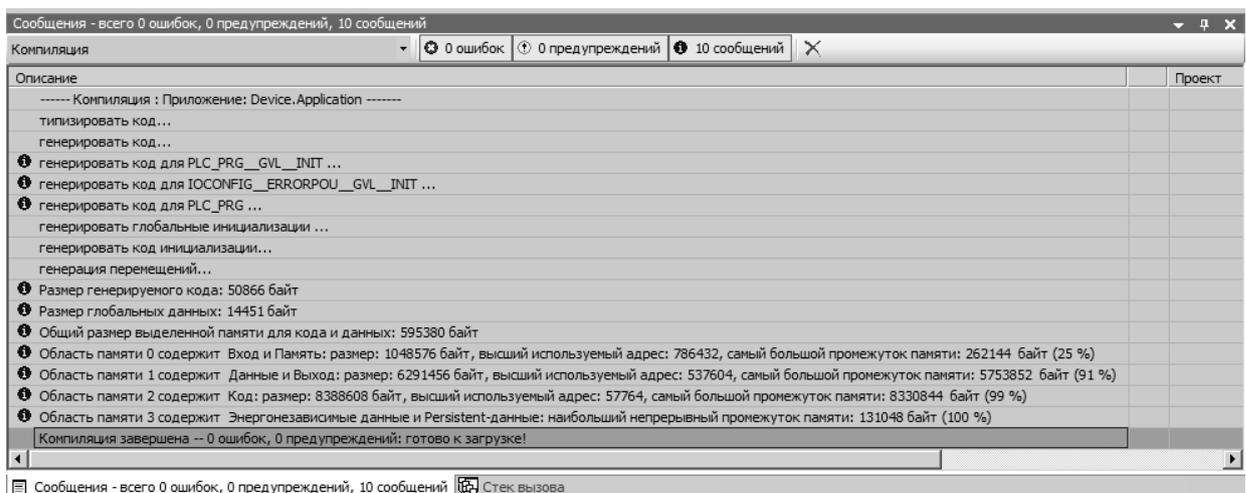


Рисунок 57 – Сообщения о безошибочной генерации кода приложения учебного проекта

2. Включите питание контроллера, если это не было сделано ранее.
3. Настройте сетевой путь доступа среды разработки к контроллеру по протоколу TCP или UDP в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.

4. В главном меню среды разработки выполните команду **Онлайн – Логин** или нажмите сочетание клавиш Alt-A8. Если с контроллером есть связь по настроенному сетевому пути, и ранее в контроллер не было загружено это же или другое приложение, на экран монитора будет выведено сообщение, показанное на рисунке 58.

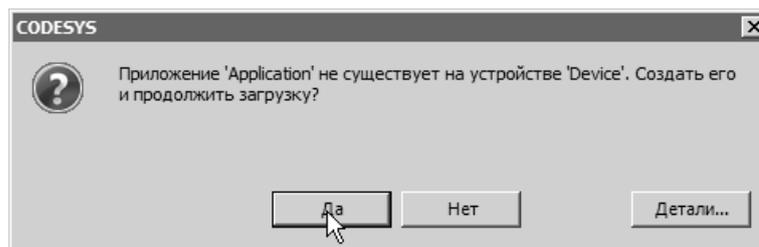


Рисунок 58 – Сообщение о первоначальной загрузке приложения Application

Для загрузки нажмите **Да**, и в строке состояния среды разработки будет отображен прогресс выполнения загрузки.

5. При успешном завершении загрузки приложения из среды разработки система исполнения контроллера оставит приложение в остановленном состоянии, о чем свидетельствует поле статуса **СТОП**, отображаемое в строке состояния среды разработки, как показано на рисунке 59, и прерывистое свечение индикатора RUN.

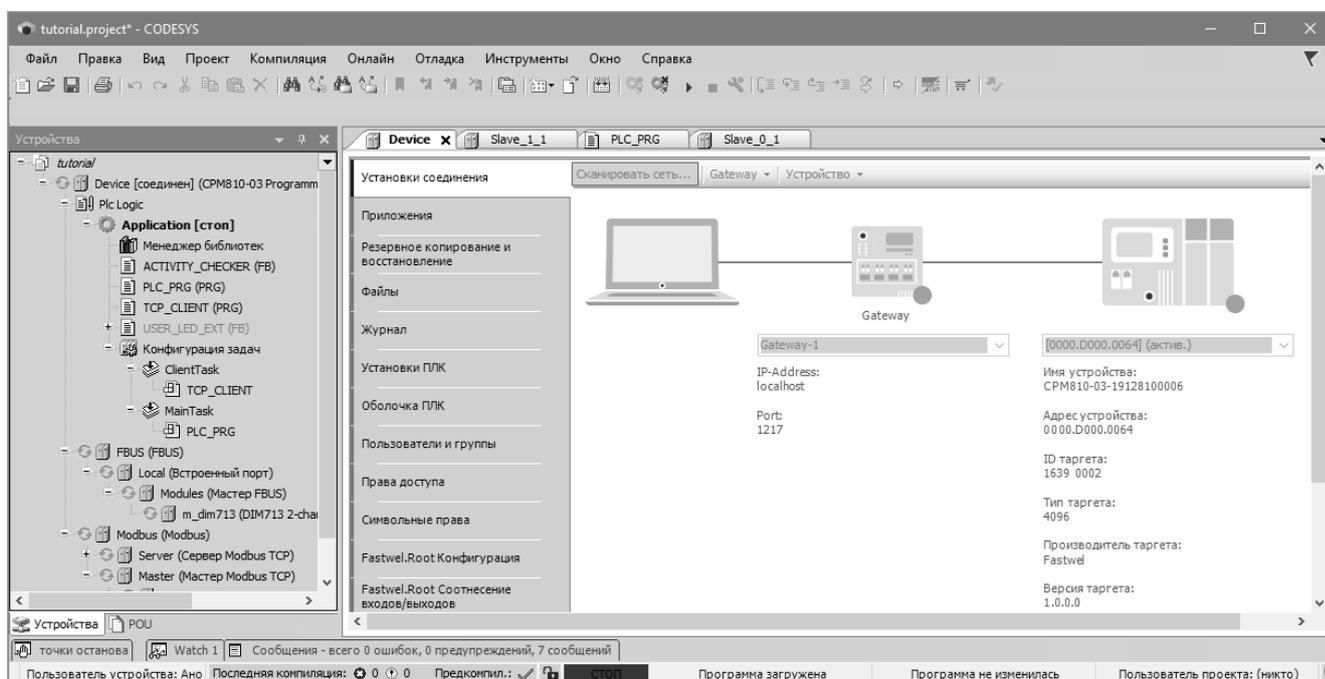


Рисунок 59 – Главное окно среды разработки после загрузки приложения из учебного проекта

6. Для запуска приложения нажмите клавишу F5 или выполните команду **Отладка – Старт** в главном меню среды разработки. При успешном запуске приложения поле статуса изменится на **ЗАПУСК**.
Если схема соединений, показанная на рисунке 4, собрана правильно, и в приложении не допущено логических ошибок и неточностей, то индикаторы LAN1 и LAN2 контроллера будут светиться прерывисто или непрерывно зеленым цветом, индикаторы RUN, APP и I/O будут светиться непрерывно зеленым цветом, а индикатор USR1 будет светиться прерывисто зеленым цветом с периодом около 1 секунды.
7. Сохраните проект командой **Файл – Сохранить** в главном меню среды разработки или нажатием сочетания клавиш Ctrl-S.
8. Если после выключения и повторного включения питания или сброса контроллера требуется запустить ранее загруженное в него приложение, выполните команду **Онлайн – Создать загрузочное приложение**.
9. Для прекращения связи между контроллером и средой разработки выполните команду **Онлайн – Отключение** или нажмите сочетание клавиш Ctrl-F8.
10. Выключите, а затем вновь включите питание контроллера или нажмите кнопку сброса, расположенную под пластиковой крышкой сервисного USB-порта. Убедитесь, что через

некоторое время индикатор USER светится прерывисто зеленым цветом, а индикаторы RUN, APP и I/O светятся непрерывно.

Если индикатор USR1 светится прерывисто красным цветом, то, согласно алгоритму загруженного приложения, это свидетельствует об отсутствии связи между клиентом и сервером MODBUS TCP по основному и резервному каналу. Перед выяснением причины отсутствия связи убедитесь, что порты LAN1 и LAN2 контроллера физически подключены к сетевому концентратору, о чем свидетельствует свечение индикаторов LAN1 и LAN2 зеленым цветом.

Причина отсутствия связи между мастером и подчиненными узлами может заключаться в установке значений IP-адресов подчиненных узлов в конфигурации мастера MODBUS TCP, отличающихся от реальных IP-адресов интерфейсов LAN1 и LAN2 (см. п. 3.7.2). Для выяснения IP-адресов интерфейсов контроллера подключитесь к нему средой разработки через последовательный порт и выполните команду оболочки ПЛИК (PLC Shell) ipinfo в соответствии с указаниями п. 3.2.3.

Если сетевые интерфейсы контроллера настроены на работу в режиме динамического назначения IP-адресов (на переключателях 1—8 установлено значение 254, см. рисунок 5), то причина отсутствия связи мастера с подчиненными узлами может быть в отсутствии в сети DHCP-сервера.

Если сетевые интерфейсы контроллера настроены на работу в режиме статического назначения адресов, на переключателях 1—8 установлено значение 3, а в конфигурации мастера заданы IP-адреса подчиненных узлов 10.0.0.3 и 10.0.0.4, то связь между мастером и подчиненными узлами, запущенными на одном контроллере, будет присутствовать, даже если порты LAN1 и LAN2 физически не подключены к сети. Это объясняется тем, что стек TCP/IP, функционирующий на некотором вычислительном устройстве (контроллере), отправляет пакет в сеть только в том случае, если физический (MAC-адрес) получателя пакета не назначен какому-либо сетевому интерфейсу данного вычислительного устройства.

3.10. Отладка приложения

3.10.1. Общие сведения

Для отладки приложения в контроллере можно воспользоваться функциями мониторинга переменных и пошаговой отладки.

Мониторинг переменных позволяет просматривать текущие значения переменных приложения, записывать или форсировать новые значения требуемых переменных во время работы приложения в контроллере. Кроме того, имеется возможность просмотра статистики выполнения задач приложения, в том числе значения счетчиков циклов задач, минимальное, максимальное и среднее значения времени выполнения цикла каждой задачи.

Функция пошаговой отладки позволяет устанавливать безусловные и условные точки останова в исходном тексте программных единиц приложения, просматривать стек вызовов и перемещаться по нему, продолжать выполнение в пошаговом режиме, при котором выполнение отлаживаемого алгоритма продолжается от текущей точки останова до высокоуровневой или элементарной конструкции языка программирования.

Более подробная информация о функциях отладки и мониторинга переменных приведена в руководстве пользователя Fastwel PLC Application Toolkit.

3.10.2. Мониторинг переменных

Для проверки правильности функционирования алгоритмов приложения:

1. Если это не было сделано ранее, загрузите приложение в контроллер в соответствии с указаниями п. 3.9.2 настоящего документа.
2. Оставаясь в состоянии соединения среды разработки с контроллером, откройте в ST-редакторе программу *PLC_PRG*, для чего дважды щелкните на элементе *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – PLC_PRG* в дереве проекта.
3. В области деклараций *PLC_PRG* раскройте переменные *m_do_1.ControlsReadback* и *sw_reader.SwitchsesState*, как показано на рисунке 60. Переменная *m_do_1* является экземпляром функционального блока *DIM7DO_SIMPLE_STIN* из библиотеки *FastwelFbusIO*, который управляет каналами модуля DIM713 и считывает текущее

состояние каналов модуля, а *sw_reader* – экземпляром функционального блока *SW_READER_x10* из библиотеки *FastwelBoard*, возвращающим текущее состояние 10 переключателей контроллера.

Выражение	Тип	Значение	Подготовленное з...	Ar
m_do_1	FastwelFbusIO.DIM7DO_SIMPLE_STIN			
Controls	ARRAY [1..DIM7DO_MAX_CHANNELS_COUNT] O...			
Inputs	DIM7DO_Inputs			
ControlsReadback	ARRAY [1..DIM7DO_MAX_CHANNELS_COUNT] O...			
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
ControlsReadback...	BOOL	FALSE		
bDiagnostics	ARRAY [1..DIM7DO_MAX_CHANNELS_COUNT] O...			
Outputs	DIM7DO_Outputs			
idx	INT	<Установите точку ос...		
b_tmp	BYTE	<Установите точку ос...		
value_mask	BYTE	<Установите точку ос...		
dwCycleCounter	DWORD	989045		
net_act_checker	ACTIVITY_CHECKER			
sw_reader	FastwelBoard.SWITCH_READER_x10			
SWITCHES_NUMBER	INT	10		
Period	TIME	T#100ms		
SwitchStates	ARRAY [1..SWITCHES_NUMBER] OF BOOL			
SwitchStates[1]	BOOL	TRUE		
SwitchStates[2]	BOOL	TRUE		
SwitchStates[3]	BOOL	FALSE		
SwitchStates[4]	BOOL	FALSE		
SwitchStates[5]	BOOL	FALSE		
SwitchStates[6]	BOOL	FALSE		
SwitchStates[7]	BOOL	FALSE		
SwitchStates[8]	BOOL	FALSE		
SwitchStates[9]	BOOL	FALSE		
SwitchStates[10]	BOOL	FALSE		
Failure	BOOL	FALSE		
sw_mask_prev	DWORD	0		
next_time_to_run	TIME	T#2h47m52s138ms		

```

1 // Увеличиваем счетчик циклов алгоритма
2 dwCycleCounter := dwCycleCounter + 1;
3
4 // Вызываем блок чтения положения переключателей
5 sw_reader(Period := SWITCHES_UPDATE_PERIOD);
6
7 // Вызываем блок проверки наличия сетевого обмена с клиентом MODBUS TCP

```

Рисунок 60 – Мониторинг переменных PLC_PRG

4. Включите переключатель 10 блока переключателей контроллера. Убедитесь, что включился первый канал модуля DIM713 (индикатор А светится).
5. Убедитесь, что в ячейках **Значение** области деклараций *PLC_PRG* для переменных *m_do_1.ControlsReadback[1]* и *sw_reader.SwitchStates[10]* отображается значение *TRUE*, как показано на рисунке 61.
6. Если индикатор *USR1* светится прерывисто зеленым цветом, откройте в *ST*-редакторе программу *TCP_CLIENT*, для чего дважды щелкните на элементе *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – TCP_CLIENT* в дереве проекта.
7. Убедитесь, что переменная *app_relay1_state* и хотя бы одна из переменных *slave1_app_relay1_state* и *slave2_app_relay1_state* программы *TCP_CLIENT* имеют значение *TRUE*.
8. Выключите переключатель 10 блока переключателей контроллера. Убедитесь, что выключился первый канал модуля DIM713 (индикатор А не светится).

9. Убедитесь, что переменная *app_relay1_state* программы *TCP_CLIENT* имеет значение *FALSE*.

Выражение	Тип	Значение	Подготовленное з...
m_do_1	FastwelFbusIO.DIM7DO_SIMPLE_STIN		
Controls	ARRAY [1..DIM7DO_MAX_CHANNELS_COUNT] O...		
Inputs	DIM7DO_Inputs		
ControlsReadback	ARRAY [1..DIM7DO_MAX_CHANNELS_COUNT] O...		
ControlsReadback[1]	BOOL	TRUE	
ControlsReadback[2]	BOOL	FALSE	
ControlsReadback[3]	BOOL	FALSE	
ControlsReadback[4]	BOOL	FALSE	
ControlsReadback[5]	BOOL	FALSE	
ControlsReadback[6]	BOOL	FALSE	
ControlsReadback[7]	BOOL	FALSE	
ControlsReadback[8]	BOOL	FALSE	
bDiagnostics	ARRAY [1..DIM7DO_MAX_CHANNELS_COUNT] O...		
Outputs	DIM7DO_Outputs		
idx	INT	<Установите точку ос...	
b_tmp	BYTE	<Установите точку ос...	
value_mask	BYTE	<Установите точку ос...	
dwCycleCounter	DWORD	1025493	
net_act_checker	ACTIVITY_CHECKER		
sw_reader	FastwelBoard.SWITCH_READER_x10		
SWITCHES_NUMBER	INT	10	
Period	TIME	T#100ms	
SwitchStates	ARRAY [1..SWITCHES_NUMBER] OF BOOL		
SwitchStates[1]	BOOL	TRUE	
SwitchStates[2]	BOOL	TRUE	
SwitchStates[3]	BOOL	FALSE	
SwitchStates[4]	BOOL	FALSE	
SwitchStates[5]	BOOL	FALSE	
SwitchStates[6]	BOOL	FALSE	
SwitchStates[7]	BOOL	FALSE	
SwitchStates[8]	BOOL	FALSE	
SwitchStates[9]	BOOL	FALSE	
SwitchStates[10]	BOOL	TRUE	
Failure	BOOL	FALSE	
sw_mask_prev	DWORD	0	
next_time_to_run	TIME	T#2h53m56s638ms	

Рисунок 61 – Контроль включения первого канала DIM713 по включению переключателя 10

10. Однократно щелкните левой кнопкой мыши над ячейкой **Подготовленное значение** для переменной *app_control_cmd*. В ячейке **Подготовленное значение** появится значение *TRUE*, как показано на рисунке 62.

Выражение	Тип	Значение	Подготовленное значение
NET_CHECK_PERIOD	TIME	T#500ms	
APP_CHECK_PERIOD	TIME	T#100ms	
app_control_cmd	BOOL	FALSE	TRUE
app_relay1_state	BOOL	FALSE	
app_relay2_state	BOOL	FALSE	

Рисунок 62 – Подготовка значения для записи в переменную

- Для записи значения в переменную *app_control_cmd* выполните команду меню **Отладка – Записать значение** или нажмите сочетание клавиш **Ctrl-F7**.
- Убедитесь, что включился второй канал модуля DIM713 (индикатор **C** светится), а переменная *app_relay2_state* программы *TCP_CLIENT* имеет значение *TRUE*.
- Запишите значение *FALSE* в переменную *app_control_cmd* программы *TCP_CLIENT* и убедитесь, что второй канал модуля DIM713 выключился (индикатор **C** не светится), а переменная *app_relay2_state* программы *TCP_CLIENT* имеет значение *FALSE*.
- Для прекращения связи между средой разработки и контроллером выполните команду **Онлайн – Отключение** или нажмите сочетание клавиш **Ctrl-F8**.

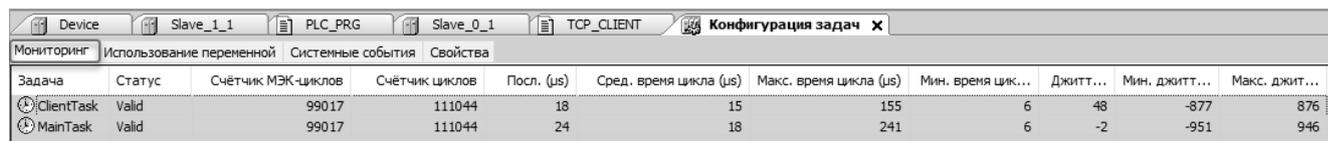
3.10.3. Просмотр статистики выполнения задач

В учебном проекте используются две задачи, *ClientTask* и *MainTask*, с одинаковыми значениями приоритета (15) и периода цикла (10 мс).

Приоритет 15, устанавливаемый по умолчанию при добавлении задачи в приложение, на контроллере СРМ810-03 означает, что задача с таким приоритетом вытесняется сервисом ввода-вывода, обслуживающим обмен с по межмодульной шине с модулями ввода-вывода, а также другими задачами с приоритетом от 0 до 14 в конфигурации приложения. Кроме того, задача с приоритетом 15 вытесняет все задачи с приоритетом от 16 до 31, а также коммуникационные сервисы для связи со средой разработки и по протоколу MODBUS / MODBUS TCP. При этом две задачи одинакового приоритета от 0 до 15 не вытесняют друг друга, а получают процессор до тех пор, пока не станет готовой к выполнению задача более высокого (меньшего по значению) приоритета или пока не закончат выполнение своих очередных циклов.

Таким образом, в приложении учебного проекта используются совершенно равноправные задачи, имеющие преимущество у планировщика перед сервисами протокола MODBUS TCP и коммуникационными сервисами взаимодействия со средой разработки, но не имеющие преимущества перед сервисом ввода-вывода. Более подробная информация о распределении приоритетов задач в системе исполнения приложений МЭК 61131-3 приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Для просмотра статистики выполнения задач приложения, оставаясь в состоянии соединения среды разработки с контроллером, дважды щелкните левой кнопкой мыши на элементе *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – Конфигурация задач* и на вкладке **Конфигурация задач** щелкните на вложенной вкладке **Мониторинг**, как показано на рисунке 63



Задача	Статус	Счётчик МЭК-циклов	Счётчик циклов	Посл. (us)	Сред. время цикла (us)	Макс. время цикла (us)	Мин. время цик...	Джитт...	Мин. джитт...	Макс. джитт...
ClientTask	Valid	99017	111044	18	15	155	6	48	-877	876
MainTask	Valid	99017	111044	24	18	241	6	-2	-951	946

Рисунок 63 – Отображение статистики выполнения задач приложения

Столбец **Счетчик МЭК-циклов** содержит количество циклов задачи, в которых выполнялся код программных единиц из списка вызовов данной задачи, а столбец **Счетчик циклов** – общее количество циклов задачи. Если приложение останавливалось или подвергалось отладке, данные счетчики будут иметь разные значения.

Столбцы **Посл.(us)**, **Сред. время цикла (us)**, **Макс. время цикла (us)** и **Мин. время цикла (us)** содержат текущее, среднее, максимальное и минимальное значения времени выполнения кода программных единиц из списка вызовов задачи в микросекундах за цикл.

Столбцы **Джиттер (us)**, **Мин. джиттер (us)** и **Макс. джиттер (us)** содержат текущее, минимальное и максимальное значения времени (в микросекундах) запаздывания или опережения начала очередного цикла выполнения кода программных единиц из списка выполнения задачи относительно момента времени, запланированного на предыдущем цикле. Положительное значение джиттера соответствует запаздыванию, а отрицательное – опережению. Опережение возникает в ситуации, когда требуется выровнять временную сетку циклов задачи, если на предыдущем цикле произошло запаздывание.

Для сброса статистики для какой-либо задачи выберите задачу в списке мониторинга, щелкните правой кнопкой мыши и выберите команду **Reset** в контекстном меню. Сбор статистики для выбранной задачи начнется заново.

3.10.4. Пошаговая отладка

После первоначальной загрузки приложения в контроллер до запуска командой **Отладка – Старт** (или клавишей F5) возможно начать отладку однократным выполнением цикла по команде **Отладка – Одиночный цикл** (сочетание клавиш Ctrl-F5). В таком случае все циклические задачи выполняют код программных единиц в своих списках вызовов один раз.

После запуска приложения возможно остановить его выполнение командой **Отладка – Стоп** или нажатием сочетания клавиш Shift-F8. Останов приложения означает, что все циклические задачи, входящие в приложение, после завершения выполнения кода программных единиц из списка вызовов, начиная со следующего цикла и до очередной команды **Отладка – Старт**, перестают выполнять код программных единиц.

Для установки безусловной точки останова:

1. Выполните команду **Онлайн – Логин**, если это не было сделано ранее.
2. Откройте программную единицу, подлежащую отладке, в соответствующем редакторе исходного текста.
3. Щелкните на строке исходного текста, где требуется установить точку останова. Строки, в которых потенциально возможна установка точек останова, отмечены символом ● в вертикальной области справа от области отображения номеров строк.
4. Нажмите клавишу F9 или выполните команду меню **Отладка – Переключить точку останова**.
5. Если произойдет останов приложения на точке останова, строка с точкой останова примет вид, показанный на рисунке 64, в строке состояния среды разработки будет отображен статус **HALT ON BP**.
6. Для пошагового исполнения с переходом к следующей инструкции, минуя вызов в текущей строке, нажмите F10 ("шаг с обходом" или "шаг поверху").
Для пошагового исполнения с переходом к коду вызываемой программной единицы в текущей строке, нажмите F8 ("шаг с заходом" или "шаг детальный").
Для продолжения выполнения нажмите F5.

```

1 // Увеличиваем счетчик циклов алгоритма
2 dwCycleCounter := dwCycleCounter + 1;
3
4 // Вызываем блок чтения положения переключателей
5 sw_reader(Period T#100ms := SWITCHES_UPDATE_PERIOD);
6
7 // Вызываем блок проверки наличия сетевого обмена с клиентом MODBUS TCP
8 net_act_checker(CheckPeriod T#100ms := NET_CHECK_PERIOD);
9
10 // Если нет обмена или все транзакции закончились ошибкой,
11 // используем красный цвет индикатора USER.
12 // Если есть обмен, то цвет индикатора зеленый.
13 led_color := SEL(net_act_checker.Error FALSE OR NOT net_act_checker.Active TRUE,
14               FastwelBoard.LED_COLOR_GREEN 1, FastwelBoard.LED_COLOR_RED 2);
15

```

Рисунок 64 – Останов приложения на точке останова

7. Для снятия точки останова щелкните на строке с точкой останова и нажмите клавишу F9. Кроме того, точки останова могут быть отображены, заблокированы, изменены, удалены и добавлены в отдельной панели **Точки останова**, управляемой командой **Вид – Точки останова**.

Для установки точки останова, останов на которой произойдет только при выполнении заданного условия:

1. Выполните команду **Онлайн – Логин**, если это не было сделано ранее.
2. Откройте программную единицу, подлежащую отладке, в соответствующем редакторе исходного текста.
3. Щелкните правой кнопкой мыши на строке исходного текста, где требуется установить условную точку останова, и в контекстном меню выберите команду **Новая точка останова**. Строки, в которых потенциально возможна установка точек останова, отмечены символом ● в вертикальной области справа от области отображения номеров строк.
4. Если точку останова предполагается установить в коде функционального блока, в диалоговой панели **Свойства точки останова** на вкладке **Расположение** отметьте экземпляр функционального блока, в котором требуется установить точку останова.
5. Щелкните на вкладке **Условие**, при необходимости отметьте опцию **Останов, если точка останова достигнута в одной из следующих задач** и отметьте задачу, под управлением которой должна выполняться отлаживаемая программная единица. Дополнительно в поле **Условие** можно отметить флажок **Останов, если true** и ввести условие по данным, при котором должна сработать точка останова.
6. Если требуется активизировать точку останова немедленно, отметьте флажок **Включить точку останова сразу**, и нажмите **ОК** для закрытия диалоговой панели **Свойства точки останова**.

При выполнении условий, заданных для точки останова, произойдет останов, и строка будет отмечена слева символом .

Возможная конфигурация условной точки останова в экземпляре функционального блока *PLC_PRG.net_act_checker* учебного проекта показана на рисунке 65.

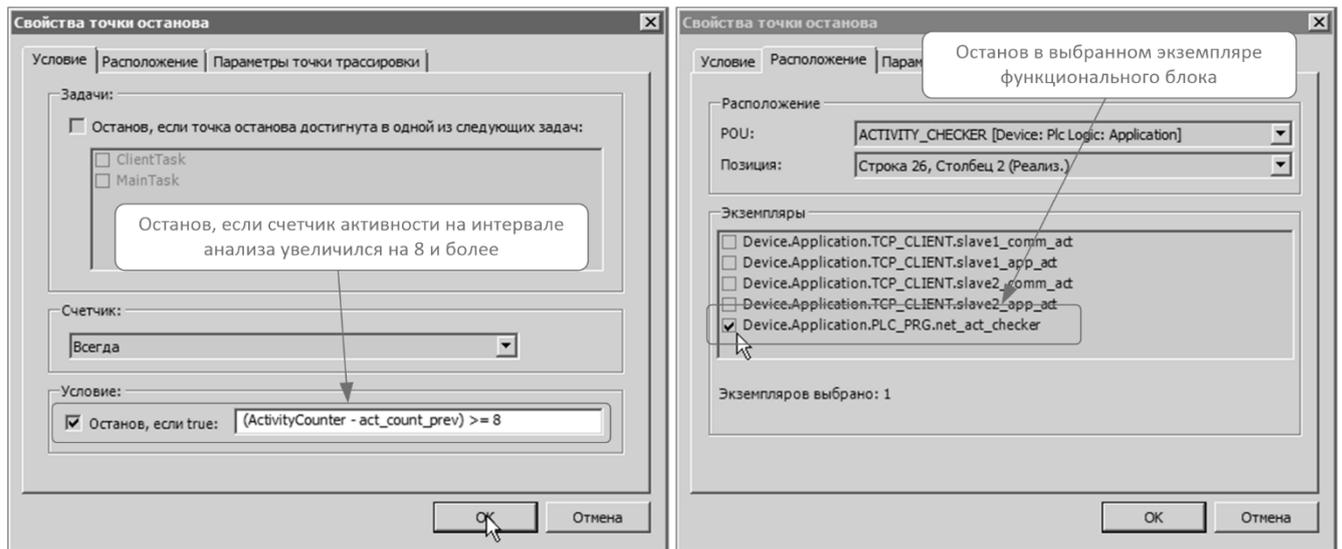


Рисунок 65 – Параметры условной точки останова

После остановки приложения командой **Отладка – Стоп** или при достижении точки останова индикатор RUN контроллера светится прерывисто зеленым цветом.

3.11. Модификация приложения в контроллере

3.11.1. Общие сведения

В процессе разработки, внедрения и обслуживания системы требуется вносить изменения в исполняемый код или конфигурацию приложения, функционирующего на контроллерах. Для контроллера СРМ810-03 определены следующие виды таких изменений:

1. Изменение конфигурации системных сервисов – модификация режима работы и IP-параметров и таблиц маршрутизации сетевых интерфейсов, управление маршрутизацией сетевого протокола взаимодействия среды разработки с контроллером, включение, отключение и настройка параметров сетевых сервисов синхронизации времени NTP и PTP, а также соотнесение отдельных экземпляров сервисов мастера и подчиненного узла MODBUS TCP с сетевыми интерфейсами контроллера и т.д.

Конфигурирование системных сервисов выполняется в веб-конфигураторе и при помощи первых восьми переключателей контроллера.

После внесения изменений в конфигурацию системных сервисов в веб-конфигураторе контроллер должен быть перезапущен командой **Применить конфигурацию** на странице **Система** веб-конфигуратора.

После внесения изменений в конфигурацию системных сервисов переключателями контроллер должен быть перезапущен любым способом: командой *reboot* **Оболочки ПЛК**, нажатием кнопки сброса, нажатием кнопки **Перезапустить контроллер** на странице **Система** веб-конфигуратора или выключением и повторным включением питания.

2. Изменение конфигурации приложения – модификация параметров системы исполнения контроллера, параметров задач, а также параметров сервиса ввода-вывода и модулей ввода-вывода и сервисов мастера и подчиненного узла MODBUS TCP, входящих в приложение IDE МЭК 61131-3, загруженное в контроллер.

Изменение конфигурации приложения выполняется в проекте IDE МЭК 61131-3 в редакторах устройств и соответствующих элементов дерева проектной информации.

После модификации конфигурации приложения в проекте, открытом в среде разработки, приложение должно быть загружено в контроллер методом полной

загрузки. После полной загрузки и команды **Онлайн – Старт** измененное приложение начинает свое исполнение с самого начала, т.е. все переменные приложения, за исключением объявленных в списке сохраняемых энергонезависимых (RETAIN PERSISTENT), получают начальные значения.

3. Изменение кода приложения – модификация имеющихся и добавление новых программных единиц в приложение, а также изменение соотношения переменных с каналами модулей ввода-вывода и коммуникационных объектов внешней сети. Модификация кода приложения выполняется в среде разработки, при этом измененное приложение может быть передано в контроллер методом полной загрузки или так называемым онлайн-изменением (Online Change), при котором в контроллер передаются только те программные единицы, которые были добавлены или изменены по сравнению с ранее загруженным приложением. При онлайн-изменении система исполнения контроллера заменяет модифицированные и добавляет новые программные единицы без останова выполнения приложения. Обратите внимание, что онлайн-изменение возможно только при наличии в каталоге размещения файла проекта дополнительного файла с расширением `compileinfo`, содержащего информацию о предыдущей компиляции приложения.

В настоящем подразделе в исходный текст приложения учебного проекта будут внесены изменения, требующие полной загрузки, а также позволяющие обновить приложение в контроллере онлайн-изменением.

3.11.2. Модификация приложения методом полной загрузки

3.11.2.1. Общие сведения

Алгоритм сервера MODBUS TCP, реализованный в программе *PLC_PRG*, осуществляет прерывистое свечение индикатора *USR1*.

В текущем варианте алгоритма при выполнении команд сброса (**Онлайн – Сброс, Сброс холодный и Сброс заводской**), а также при загрузке другого приложения в контроллер индикатор *USR1* скорее всего продолжит светиться тем цветом, который был выбран алгоритмом перед сбросом или загрузкой другого приложения.

Такое поведение может оказаться нежелательным, и в данном подразделе будет выполнена модификация алгоритма, позволяющая выключить индикатор *USR1* при любом виде сбросе и перед запуском другого приложения, загруженного в контроллер. Для реализации выключения индикатора в код приложения будет добавлен новый функциональный блок *USER_LED_EXT*, расширяющий библиотечный блок *FastwelBoard.USER_LED_CONTROL*, а в программе *PLC_PRG* тип экземпляра блока управления индикатором *USR1 led_blinker* будет заменен на вновь созданный *USER_LED_EXT*. Термин "расширяющий" в данном случае имеет смысл, соответствующий понятию объектно-ориентированного программирования "наследование интерфейса и реализации" для имеющегося типа функционального блока.

Обратите внимание, что в системном программном обеспечении контроллера CPM810-03 индикаторы *USR1* и *USR2* всегда выключаются при выполнении команд сброса и при загрузке другого приложения в контроллер.

3.11.2.2. Создание функционального блока с наследованием интерфейса и реализации

Для добавления типа функционального блока, наследующего интерфейс и реализацию библиотечного блока *FastwelBoard.USER_LED_CONTROL*:

1. В дереве проектной информации учебного проекта щелкните правой кнопкой мыши над элементом *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application* и выполните команду **Add Object – POU** в контекстном меню.
2. В окне **Add POU** в поле **Имя** введите имя *USER_LED_EXT*, в выпадающем списке **Язык реализации** выберите тип *Структурированный текст (ST)*, в поле **Тип** выберите тип создаваемой программной единицы **Функциональный блок**, а затем отметьте флажок **Extends** и нажмите кнопку  справа от поля ввода имени типа функционального блока, от которого должен наследоваться функциональный блок, как показано на рисунке 6б.

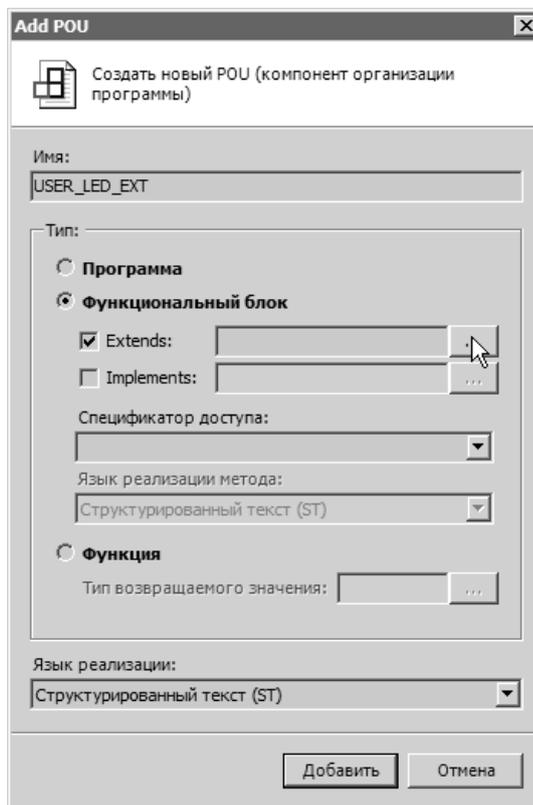


Рисунок 66 – Создание типа функционального блока с наследованием интерфейса и реализации существующего блока

3. В выведенном на экран окне **Ассистент ввода** выберите тип наследуемого функционального блока *FastwelBoard – POUs – USER_LED_CONTROL* и нажмите **ОК**. В поле **Extends** окна **Add POU** появится имя наследуемого типа функционального блока *FastwelBoard.USER_LED_CONTROL*.
4. Нажмите кнопку **Добавить** в окне **Add POU**. В дереве проекта появится вновь созданный элемент *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – USER_LED_EXT*, а в правой области окна среды разработки будет отображена вкладка ST-редактора с пустой реализацией блока *USER_LED_EXT* и следующим объявлением типа в области деклараций:


```
FUNCTION_BLOCK USER_LED_EXT EXTENDS FastwelBoard.USER_LED_CONTROL
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR
```
5. В области реализации ST-редактора введите следующий вызов:


```
SUPER^();
```

 как показано на рисунке 67.

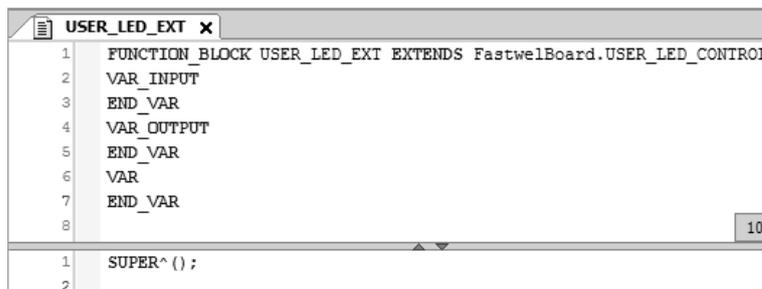


Рисунок 67 – Вызов алгоритма базового блока из реализации наследованного блока

Таким образом, при вызове любого экземпляра вновь созданного функционального блока будет выполняться алгоритм блока, от которого он унаследован.

6. Сохраните проект командой **Файл – Сохранить** или нажатием сочетания клавиш **Ctrl-S**.

3.11.2.3. Создание метода `FB_Exit` функционального блока

Для реализации функции выключения индикатора `USR1` при любом виде сброса и при полной загрузке этого же или другого приложения следует в блок `USER_LED_EXT` добавить метод `FB_Exit` со следующей сигнатурой:

```
METHOD FB_Exit : BOOL
VAR_INPUT
    bInCopyCode : BOOL;
END_VAR
```

Данный метод будет автоматически вызываться системой исполнения контроллера при завершении работы приложения по причине сброса или полной загрузки этого же или другого приложения, а также при модификации текущего приложения онлайн-изменением. Параметр метода `bInCopyCode` устанавливается в `TRUE` системой исполнения, если происходит изменение текущего приложения во время онлайн-изменения. В остальных случаях `bInCopyCode` равен `FALSE`.

Для добавления метода в блок `USER_LED_EXT`:

1. Щелкните правой кнопкой мыши над элементом *Device (CPM810-03 Programmable Automation Controller) – Plc Logic – Application – USER_LED_EXT* в дереве проекта и выполните команду **Add Object – Метод** в контекстном меню.
2. В окне **Add Method** в поле **Имя** введите `FB_Exit`, в поле **Возвращаемый тип** введите `BOOL`.
3. Нажмите кнопку **Добавить**.

В дереве проекта у блока `USER_LED_EXT` появится вложенный элемент  `FB_Exit`, а в правой области окна среды разработки будет отображена вкладка ST-редактора с пустой реализацией метода `FB_Exit`.

4. В области деклараций и области реализации введите следующий исходный текст:

```
METHOD FB_Exit : BOOL
VAR_INPUT
    bInCopyCode : BOOL;
END_VAR
```

```
IF NOT bInCopyCode THEN
    FastwelBoard.SysBoardSetUserLed( FastwelBoard.LED_COLOR_NONE);
END_IF
```

```
FB_Exit := TRUE;
```

Таким образом, во всех случаях автоматического вызова `FB_Exit`, кроме выполняемого системой исполнения контроллера при обновлении приложения онлайн-изменением, будет вызываться функция `SysBoardSetUserLed` с параметром `LED_COLOR_NONE`. Данная функция определена в библиотеке `FastwelBoard`, и, при вызове с параметром `LED_COLOR_NONE`, выключает индикатор `USR1`.

Полный исходный текст метода в ST-редакторе показан на рисунке 68.

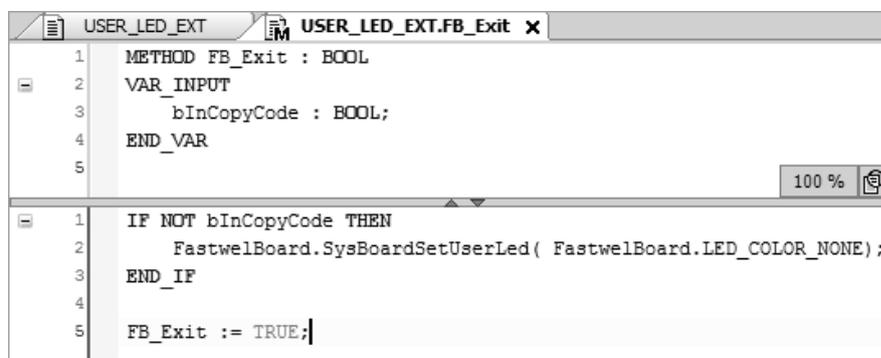


Рисунок 68 – Исходный текст метода `FB_Exit`

5. Сохраните проект командой **Файл – Сохранить** или нажатием сочетания клавиш `Ctrl-S`.

3.11.2.4. Использование USER_LED_EXT и загрузка приложения в контроллер

Для того, чтобы программа PLC_PRG приложения контроллера использовала блок USER_LED_EXT вместо библиотечного USER_LED_CONTROL, выполните следующие действия:

1. Откройте исходный текст программы PLC_PRG в ST-редакторе.
2. В области деклараций измените тип переменной led_blinker с FastwelBoard.USER_LED_CONTROL на USER_LED_EXT:

```
// Блок "мигания" индикатором USR1
led_blinker : USER_LED_EXT; //FastwelBoard.USER_LED_CONTROL;
```
3. Выполните команду **Онлайн – Логин**. Если контроллер доступен по заданному ранее сетевому пути, на экран монитора будет выведена диалоговая панель, показанная на рисунке 69.

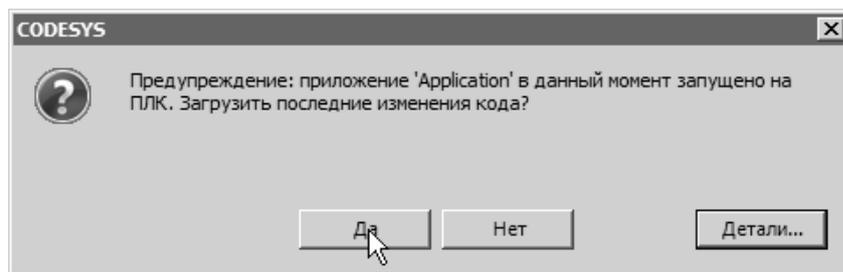


Рисунок 69 – Сообщение о существенном изменении приложения в контроллере

Кнопка **Детали** позволяет просмотреть перечень изменений в приложении.

4. Нажмите кнопку **Да**, после чего приложение будет загружено в контроллер и окажется в состоянии останова. Обратите внимание, что индикатор USR1 с, высокой вероятностью, будет светиться зеленым или красным цветом, поскольку в предыдущем приложении не было предусмотрено выключение индикатора в случае замены, полного обновления или сброса приложения.
5. Для запуска приложения нажмите клавишу F5 или выполните команду **Отладка – Старт**. Индикатор USR1 начнет светиться прерывисто зеленым или красным цветом.
6. Выполните команду **Онлайн – Сброс**. Переменная dwCycleCounter программы PLC_PRG станет равной 0, приложение перейдет в состояние останова, а индикатор USR1 будет выключен.
7. Для повторного запуска приложения нажмите клавишу F5.
8. Выполните команду **Онлайн – Отключение** и сохраните проект командой **Файл – Сохранить**.

В данном случае изменения в приложении оказались настолько существенными, что среда разработки не смогла выполнить обновление приложения методом онлайн-изменения.

При необходимости автоматического запуска приложения после полной загрузки воспользуйтесь механизмом обработки системного события *LegacyOnInit* и программного управления состоянием приложения при помощи функций системной библиотеки *StrApp*, как это сделано в примере программирования PlatformControl.project, поставляемом в составе Fastwel PLC Application Toolkit.

3.11.3. Модификация приложения онлайн-изменением (Online Change)

3.11.3.1. Общие сведения

Большинство изменений в исходном тексте приложения могут быть переданы в контроллер методом онлайн-изменения (Online Change). Данный способ модификации приложения в контроллере возможен только в том случае, если в каталоге размещения файла проекта также размещена актуальная информация о компиляции и загрузке приложения. Данная информация формируется средой разработки во время последней загрузки приложения в контроллер и сохраняется в файле с именем:

<имя проекта>. <имя устройства в приложении>. <имя приложения>. <ID приложения>. compileinfo

При онлайн-изменении приложения в контроллер передаются код и данные изменившихся и новых программных единиц, в том числе код инициализации и присвоения начальных значений имеющихся и новых переменных. Перед переключением задач приложения на исполнение кода обновленного приложения, код программных единиц, которые не изменились, копируется в новую область сегмента

кода без изменений, а код изменившихся и новых программных единиц – в последующее свободное пространство памяти в сегменте кода. Изменившиеся данные программных единиц и глобальные переменные размещаются в свободном пространстве памяти в сегменте данных. Затем система исполнения, при необходимости, разрешает ссылки на код и данные в сегменте кода измененного приложения, после чего, по завершению очередных циклов задач, переключает текущий активный сегмент кода на область памяти, в которой размещены неизменные, измененные и новые программные единицы.

В данном подразделе для иллюстрации онлайн-изменения приложения в контроллере будет расширен интерфейс и алгоритм функционального блока *USER_LED_EXT*, разработанного при выполнении указаний п. 3.11.2.

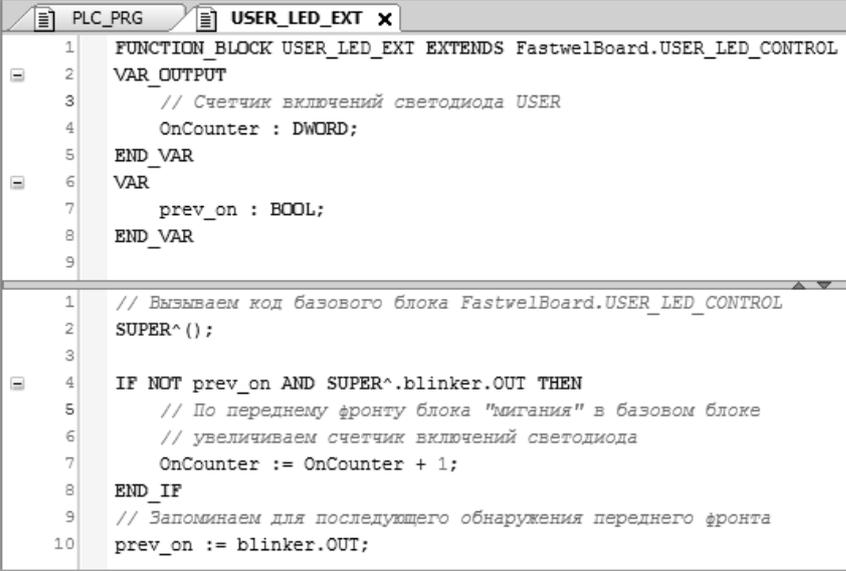
3.11.3.2. Изменение функционального блока *USER_LED_EXT* и онлайн-изменение приложения

1. Откройте код функционального блока *USER_LED_EXT*, разработанного при выполнении указаний п. 3.11.2.
2. Внесите следующие изменения в области деклараций и реализации:

```
FUNCTION_BLOCK USER_LED_EXT EXTENDS FastwelBoard.USER_LED_CONTROL
VAR_OUTPUT
    // Счетчик включений светодиода USR1
    OnCounter : DWORD;
END_VAR
VAR
    prev_on : BOOL;
END_VAR
// Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
SUPER^();

IF NOT prev_on AND SUPER^.blinker.OUT THEN
    // По переднему фронту блока "мигания" в базовом блоке
    // увеличиваем счетчик включений светодиода
    OnCounter := OnCounter + 1;
END_IF
// Запоминаем для последующего обнаружения переднего фронта
prev_on := blinker.OUT;
```

Исходный текст измененного блока показан на рисунке 70.



```
1 FUNCTION_BLOCK USER_LED_EXT EXTENDS FastwelBoard.USER_LED_CONTROL
2 VAR_OUTPUT
3     // Счетчик включений светодиода USER
4     OnCounter : DWORD;
5 END_VAR
6 VAR
7     prev_on : BOOL;
8 END_VAR
9
10 // Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
11 SUPER^();
12
13 IF NOT prev_on AND SUPER^.blinker.OUT THEN
14     // По переднему фронту блока "мигания" в базовом блоке
15     // увеличиваем счетчик включений светодиода
16     OnCounter := OnCounter + 1;
17 END_IF
18 // Запоминаем для последующего обнаружения переднего фронта
19 prev_on := blinker.OUT;
```

Рисунок 70 – Модификация блока *USER_LED_EXT* для загрузки онлайн-изменений

3. Выполните команду **Онлайн – Логин**. Если контроллер доступен по ранее установленному сетевому пути, на экран монитора будет выведена диалоговая панель, показанная на рисунке 71.

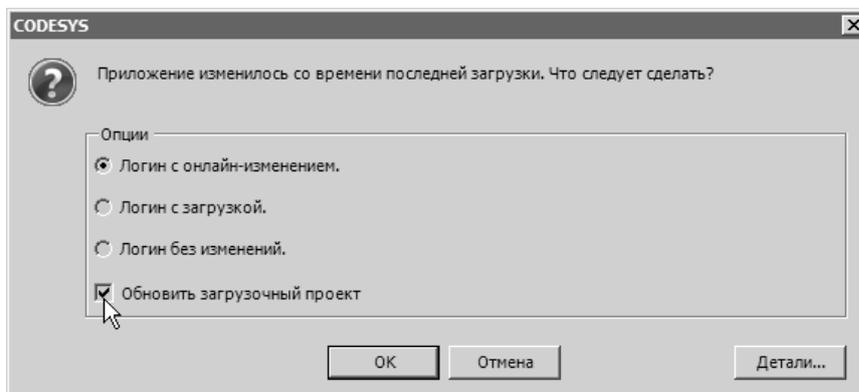


Рисунок 71 – Диалоговая панель выбора варианта загрузки модифицированного приложения

4. Для онлайн-изменения приложения в контроллере оставьте выбранной опцию **Логин с онлайн-изменением**.
5. Для обновления загрузочного приложения в контроллере отметьте флажок **Обновить загрузочный проект**, как показано на рисунке 71. Обновление загрузочного приложения необходимо в случае, если требуется, чтобы после перезапуска контроллер начал выполнять модифицированное приложение.

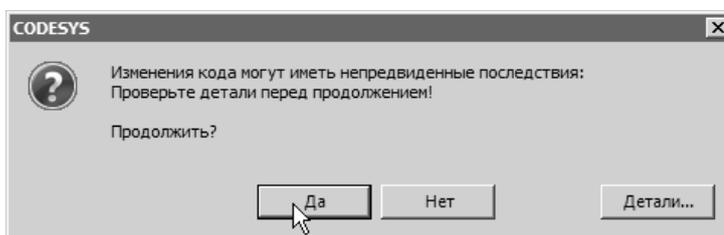


Рисунок 72 – Предупреждение о существенных изменениях в модифицированном приложении

6. Если при генерации кода перед началом онлайн-изменения среда разработки обнаружила, что модификация носит существенный характер, на экран монитора выводится предупреждение, показанное на рисунке 72.
В некоторых случаях онлайн-изменение приложения в контроллере может привести к неправильной работе приложения. К таким случаям прежде всего относится изменение состава и типов входных, выходных и внутренних переменных программ и функциональных блоков, определяющих их состояние между вызовами, при которых алгоритм может оказаться в неожиданном участке кода или вычислить неправильные выходные данные.
Нажмите кнопку **Да** в диалоговой панели предупреждения, т.к. произведенная модификация функционального блока *USER_LED_EXT* не нарушает логику работы ранее загруженного приложения, а правильность функционирования измененного алгоритма не зависит от состояния алгоритма до изменения.
7. После успешной онлайн-изменения приложения в контроллере, оставаясь в состоянии соединения с контроллером, откройте исходный текст функционального блока *USER_LED_EXT* и убедитесь, что значение переменной *OnCounter* увеличивается на 1 примерно один раз в секунду.
8. Выполните команду **Онлайн – Отключение** и сохраните проект командой **Файл – Сохранить**.

3.11.4. Идентификационная информация проекта

Для поддержки формального процесса конфигурационного управления разработкой прикладного программного обеспечения среда разработки обеспечивает возможность задания идентификационной информации о проекте приложения, загружаемого в контроллер. Идентификационная информация включает в себя название предприятия, имя проекта, номер версии и ряд других свойств и может получена как программным способом в коде приложения, так и удаленно при помощи команды оболочки ПЛК *pinf*.

Для ввода идентификационной информации для приложения:

1. Выполните команду **Проект – Информация проекта**.
2. В окне **Информация проекта** введите название компании в поле **Компания**, название приложения – в поле **Заголовок**, номер версии – в поле **Версия**, и отметьте флажок **Автоматически генерировать POU 'Информация о проекте'**, как показано на рисунке 73.

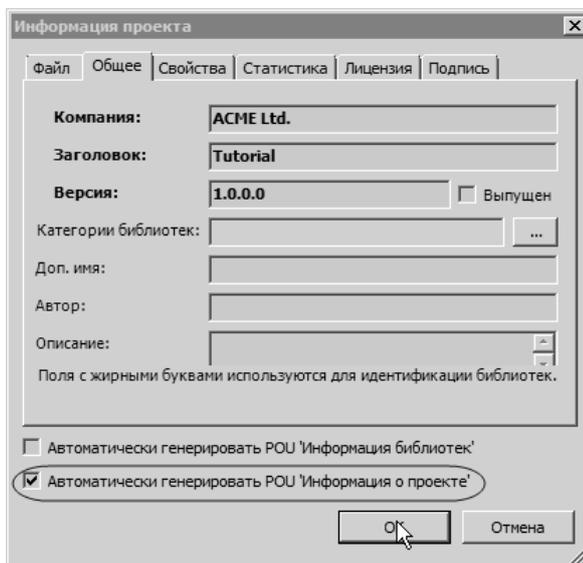


Рисунок 73 – Ввод идентификационной информации о проекте

Установка флажка **Выпущен** позволяет запретить последующие изменения проекта после того, как приложение отлажено и выпущено.

3. Закройте окно **Информация о проекте**, нажав **ОК**, а затем выполните команду меню **Компиляция – Генерировать код**.
4. При отсутствии ошибок компиляции загрузите приложение в контроллер. С высокой вероятностью, среда разработки предложит произвести онлайн-изменение.
5. Перейдите на вкладку редактора устройства для элемента проектной информации *Device (CPM810-03 Programmable Automation Controller)*, а затем выберите вложенную вкладку **Оболочка ПЛК**.
6. В поле ввода оболочки ПЛК введите команду *pinf*. В области результата выполнения команды будет отображена информация, подобная приведенной на рисунке 74.

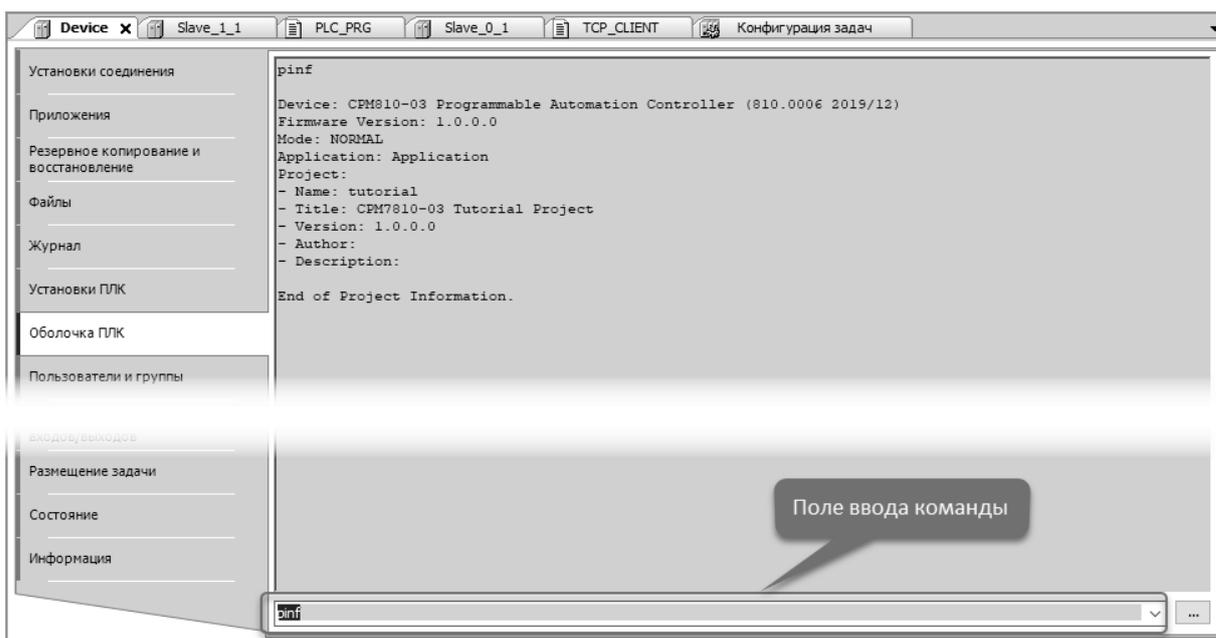


Рисунок 74 – Получение идентификационной информации о контроллере

Информация о типе устройства, версии системного программного обеспечения и режиме работы системы исполнения контроллера передается в полях *Device*, *Firmware Version* и *Mode*.

Информация о загруженном и выполняющемся приложении и проекте отображается в полях *Application*, *Project.Name*, *Project.Title* и *Project.Version*.

Идентификационная информация проекта доступна в коде приложения путем вызова функции *AppGetProjectInformation* из библиотеки *StrApp* или функций, автоматически сгенерированных средой разработки, в дереве общих элементов проектной информации **POU**, как показано на рисунке 75.

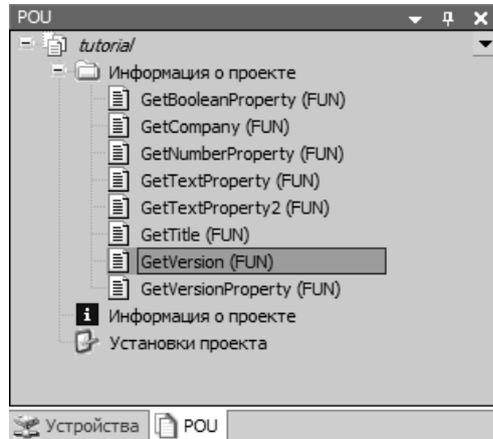


Рисунок 75 – Автоматически сгенерированные функции получения идентификационной информации

Для получение информации о версии проекта в коде приложения:

1. В области деклараций одной из программных единиц или в списке глобальных переменных объявите переменную структурного типа *VERSION*:
ProjectVersion: VERSION;
2. В коде одной из программных единиц вызовите автоматически сгенерированную функцию получения версии следующим образом:
ProjectVersion := GetVersion();

Обратите внимание, что при использовании автоматически сгенерированных функций получения идентификационной информации в коде приложения, как правило, требуется использовать версию компилятора, соответствующую используемой версии IDE МЭК 61131-3.

3.12. Диагностика критических отказов приложения

3.12.1. Общие сведения

Во время работы контроллера при возникновении критических отказов в приложении система исполнения переводит контроллер в безопасный режим. В безопасном режиме обмен данными по межмодульной шине с модулями ввода-вывода не выполняется, и их выходы могут быть переведены в безопасное состояние, заданное в конфигурации приложения. Таким образом исключается неконтролируемое воздействие на объект управления.

При работе контроллера в безопасном режиме индикатор RUN светится прерывисто, меняя цвет с зеленого на красный и погасая. Если запуск или перезапуск в безопасном режиме обусловлен критическим отказом, область нахождения причины отказа обозначается индикатором APP при помощи цветовой сигнализации.

Существуют следующие основные классы причин перехода контроллера в безопасный режим:

1. Исключение – причины данного класса связаны с проявлением дефектов в пользовательском коде приложения или в системном программном обеспечении, при наличии которых система исполнения не может обеспечить предсказуемое выполнение пользовательских алгоритмов и функционирование периферийных устройств, подключенных к контроллеру.
2. Зацикливание в пользовательском коде – причины данного класса связаны с наличием бесконечных (или почти бесконечных) циклов в пользовательском коде.

3. Дефицит системных ресурсов – причины данного класса обусловлены нехваткой памяти или других ресурсов, возникающей после загрузки приложения или в процессе его работы.

Настоящий подраздел содержит указания по определению причины перехода контроллера в безопасный режим и нахождению местоположения дефекта в коде приложения на примере учебного проекта.

3.12.2. Поиск местоположения бесконечного цикла в коде приложении

3.12.2.1. Общие сведения

Заикливание в коде приложения обычно происходит в циклах FOR, WHILE и REPEAT из-за ошибок в определении условия окончания выполнения цикла, в том числе в операциях сравнения и/или преобразования знаковых типовых в беззнаковые и наоборот. В ряде случаев заикливание может произойти по истечении очень длительного времени работы приложения, например, если некорректная операция определения окончания выполнения цикла использует в качестве параметра 32-разрядную переменную.

Для своевременного обнаружения заикливания системой исполнения все циклические задачи приложения должны быть снабжены сторожевым таймером, настраиваемым в свойствах каждой задачи, как показано на рисунке 76.

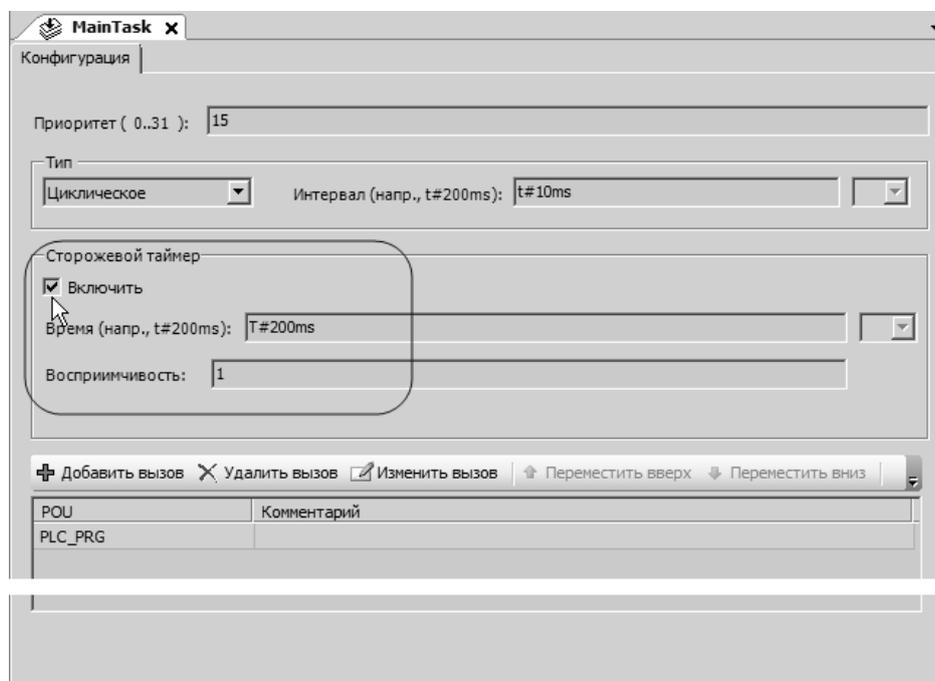


Рисунок 76 – Включение и настройка сторожевого таймера задачи

Установка переключателя **Сторожевой таймер – Включить** активизирует сторожевой таймер задачи после загрузки приложения в контроллер.

Параметр **Время** определяет интервал времени, в течение которого должен гарантированно завершиться очередной цикл контролируемой задачи после своего начала.

Параметр **Восприимчивость** определяет верхний предел количества циклов задачи, следующих друг за другом, в которых может быть превышен заданный интервал сторожевого таймера задачи до перевода контроллера в безопасный режим по заикливанию. Если для параметра **Восприимчивость** установлено значение 0, 1 или 2, то для цикла задачи допускается однократное превышение интервала сторожевого таймера, а при втором подряд превышении контроллер будет переведен в безопасный режим. Для значений более 2 допустимое количество циклов с превышением на 1 меньше заданного значения (для 3 допускается 2 цикла с превышением, для n – n-1 цикл).

Если сторожевой таймер задачи не включен, то для всех задач система исполнения включает его автоматически и устанавливает интервал сторожевого таймера 30 секунд и восприимчивость 1.

3.12.2.2. Ошибка зацикливания в учебном проекте

Для имитации ошибки зацикливания в учебном проекте включите сторожевой таймер для задачи *MainTask*, как показано на рисунке 76, а затем откройте исходный текст функционального блока *USER_LED_EXT*, разработанного при выполнении указаний п. 3.11.2, в ST-редакторе и измените его реализацию следующим образом:

```
// Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
SUPER^();

IF NOT prev_on AND SUPER^.blinker.OUT THEN
  // По переднему фронту блока "мигания" в базовом блоке
  // увеличиваем счетчик включений светодиода
  OnCounter := OnCounter + 1;
END_IF

(* ДОБАВЛЯЕМ КОД ДЛЯ ЗАЦИКЛИВАНИЯ *)
IF OnCounter >= 20 THEN
  WHILE OnCounter >= 0 DO
    OnCounter := OnCounter - 1;
  END_WHILE
END_IF

// Запоминаем для последующего обнаружения переднего фронта
prev_on := blinker.OUT;
```

Таким образом, как только индикатор USR1 "мигнет" 20 (или более) раз, задача, под управлением которой вызывается экземпляр функционального блока *USER_LED_EXT*, начнет выполнение цикла WHILE. Поскольку переменная *OnCounter* имеет тип *DWORD*, условие повторения цикла всегда будет выполнено, и задача, из которой вызывается экземпляр блока *USER_LED_EXT*, зависнет. Данная ошибка иногда встречается в реальных приложениях при выполнении повторяющихся операций, когда не учитывается тот факт, что переменная цикла определена как беззнаковая. Если бы переменная *OnCounter* имела тип *SINT*, *INT* или *DINT*, то проблема бы не возникла.

Выполните команду **Компиляция – Генерировать код** и загрузите приложение в контроллер методом полной загрузки, как указано в п. 3.11.2, нажмите клавишу F5 для запуска приложения, а затем выполните команду **Онлайн – Отключение**. Обратите внимание, что если среда разработки будет подключена к контроллеру во время перехода в безопасный режим, то по прошествии заметного времени произойдет разрыв связи между средой разработки и контроллером. При этом до разрыва связи какие-либо манипуляции со средой разработки будут невозможными.

Примерно через 20-30 секунд контроллер перейдет в безопасный режим: индикатор RUN будет светиться прерывисто, меняя цвет с красного на зеленый и погасая, а индикатор APP будет светиться прерывисто зеленым цветом с частотой около 1 Гц.

3.12.2.3. Получение информации о местоположении зацикливания

Если контроллер функционирует в безопасном режиме по ошибке в приложении, для получения информации о местоположении ошибки в исходном тексте:

1. Если это не было сделано ранее, откройте проект, из которого было загружено приложение с ошибкой.
2. В дереве проекта дважды щелкните на элементе, соответствующем используемому контроллеру. В случае CPM810-03 откройте редактор устройства для элемента типа (*CPM810-03 Programmable Automation Controller*).
3. На вложенной вкладке **Установки соединения** редактора устройства нажмите **Сканировать сеть** и выберите сетевой путь к требуемому контроллеру, который перешел в безопасный режим, после чего нажмите **ОК**.
4. Перейдите на вложенную вкладку **Журнал** и нажмите кнопку  в его панели инструментов. Если контроллер доступен по выбранному сетевому пути, из него будут выгружены записи системного журнала, как показано на рисунке 77.

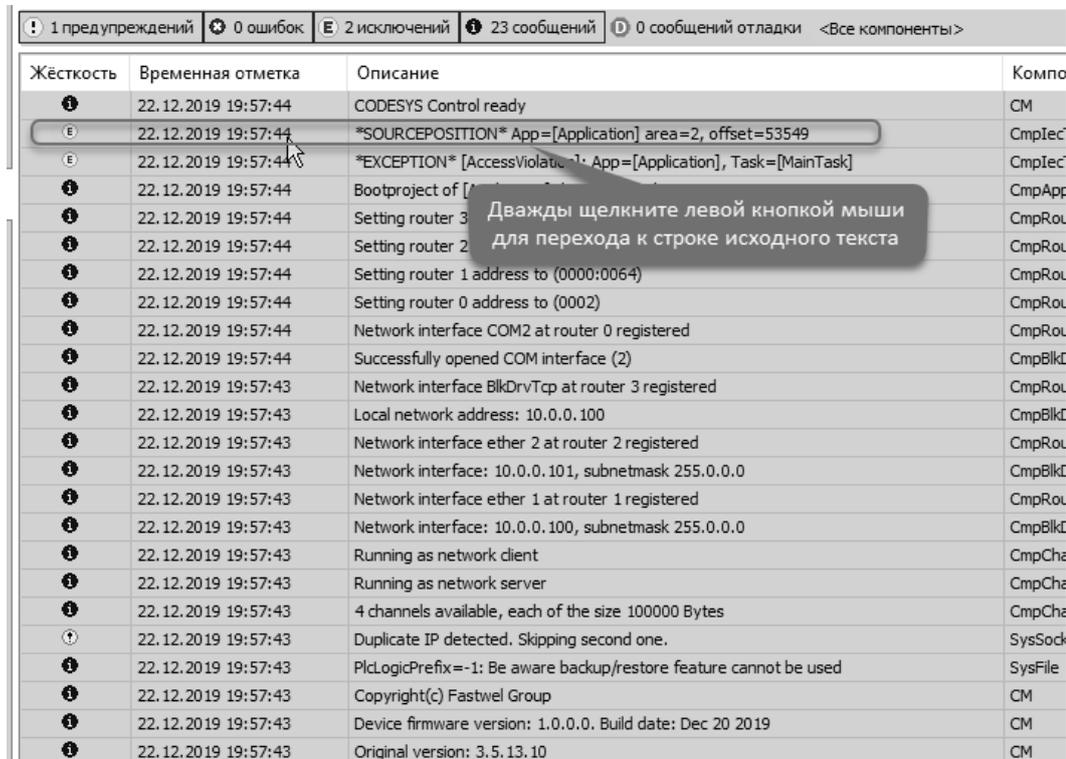
Информация об исключении по зацикливанию содержится строке, обозначенной символом , следующего содержания:

```
*EXCEPTION* [Watchdog]: App=[имя приложения], Task=[имя задачи]
```

Информация о местоположении исключения в коде приложения содержится в соседней (снизу вверх) строке, также обозначенной символом , следующего содержания:

- *SOURCEPOSITION* App=[имя приложения] area=<номер области памяти>, offset=<смещение>
5. Для перехода к строке исходного текста, при выполнении которой произошло исключение, дважды щелкните левой кнопкой мыши на строке  *SOURCEPOSITION*, как показано на рисунке 77.

Среда разработки откроет программную единицу, в которой произошло исключение, и установит курсор на начало выражения, в котором было зафиксировано зависание, и произошло исключение с переходом в безопасный режим, как показано на рисунке 78.



Жёсткость	Временная отметка	Описание	Компо
	22.12.2019 19:57:44	CODESYS Control ready	CM
	22.12.2019 19:57:44	*SOURCEPOSITION* App=[Application] area=2, offset=53549	CmpIec
	22.12.2019 19:57:44	*EXCEPTION* [AccessViolation]: App=[Application], Task=[MainTask]	CmpIec
	22.12.2019 19:57:44	Bootproject of [...]	CmpApp
	22.12.2019 19:57:44	Setting router 3	CmpRou
	22.12.2019 19:57:44	Setting router 2	CmpRou
	22.12.2019 19:57:44	Setting router 1 address to (0000:0064)	CmpRou
	22.12.2019 19:57:44	Setting router 0 address to (0002)	CmpRou
	22.12.2019 19:57:44	Network interface COM2 at router 0 registered	CmpRou
	22.12.2019 19:57:44	Successfully opened COM interface (2)	CmpBlkI
	22.12.2019 19:57:43	Network interface BlkDrvTcp at router 3 registered	CmpRou
	22.12.2019 19:57:43	Local network address: 10.0.0.100	CmpBlkI
	22.12.2019 19:57:43	Network interface ether 2 at router 2 registered	CmpRou
	22.12.2019 19:57:43	Network interface: 10.0.0.101, subnetmask 255.0.0.0	CmpBlkI
	22.12.2019 19:57:43	Network interface ether 1 at router 1 registered	CmpRou
	22.12.2019 19:57:43	Network interface: 10.0.0.100, subnetmask 255.0.0.0	CmpBlkI
	22.12.2019 19:57:43	Running as network client	CmpCha
	22.12.2019 19:57:43	Running as network server	CmpCha
	22.12.2019 19:57:43	4 channels available, each of the size 100000 Bytes	CmpCha
	22.12.2019 19:57:43	Duplicate IP detected. Skipping second one.	SysSock
	22.12.2019 19:57:43	PlcLogicPrefix=-1: Be aware backup/restore feature cannot be used	SysFile
	22.12.2019 19:57:43	Copyright(c) Fastwel Group	CM
	22.12.2019 19:57:43	Device firmware version: 1.0.0.0. Build date: Dec 20 2019	CM
	22.12.2019 19:57:43	Original version: 3.5.13.10	CM

Рисунок 77 – Информация об исключении по срабатыванию сторожевого таймера задачи MainTask

```

1 // Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
2 SUPER^();
3
4 IF NOT prev_on AND SUPER^.blinker.OUT THEN
5     // По переднему фронту блока "мигания" в базовом блоке
6     // увеличиваем счетчик включений светодиода
7     OnCounter := OnCounter + 1;
8 END_IF
9
10 (**)
11 IF OnCounter >= 20 THEN
12     WHILE OnCounter >= 0 DO
13         OnCounter := OnCounter - 1;
14     END_WHILE
15 END_IF
16 (**)
17 // Запоминаем для последующего обнаружения переднего фронта
18 prev_on := blinker.OUT;

```

Рисунок 78 – Переход и отображение позиции в коде, где произошло исключение по зацикливанию

6. Для прекращения связи с системным журналом контроллера нажмите кнопку  в панели инструментов на вкладке **Журнал**, чтобы кнопка перестала отображаться в нажатом состоянии.
7. Закомментируйте фрагмент кода, в котором допущена ошибка, приводящая к зависанию, и повторно загрузите приложение в контроллер. После этого контроллер будет функционировать в нормальном режиме.
8. Выполните команду **Онлайн – Отключение** и сохраните проект командой **Файл – Сохранить**.

3.12.3. Поиск местоположения ошибки доступа к памяти в коде приложения

3.12.3.1. Общие сведения

Реализация языков программирования ГОСТ Р МЭК 61131-3 в среде разработки поддерживает тип данных "указатель" (POINTER TO) или "адрес памяти", что обеспечивает возможность разработки сложных программ и системных сервисов, к каковым относятся сетевые протоколы прикладного уровня.

Использование указателей в коде приложения требует довольно высокой квалификации программиста, которая, однако, не гарантирует от наличия ошибок в приложениях, поскольку переменные-указатели содержат адреса в памяти контроллера, и запись или чтение по неправильному адресу приводит к исключению и переходу контроллера в безопасный режим.

В данном подразделе в исходный текст учебного проекта будет намеренно внесена ошибка доступа к памяти через переменную-указатель, а затем даны указания по определению признаков и местоположения ошибки.

3.12.3.2. Ошибка доступа к памяти в учебном проекте

Для внесения ошибки доступа к памяти в код приложения учебного проекта откройте исходный текст функционального блока *USER_LED_EXT*, разработанного при выполнении указаний п. 3.11.2, в ST-редакторе и измените его реализацию следующим образом:

```
// Функциональный блок, расширяющий интерфейс и реализацию наследуемого
// библиотечного блока USER_LED_CONTROL
FUNCTION_BLOCK USER_LED_EXT EXTENDS FastwelBoard.USER_LED_CONTROL
VAR_OUTPUT
    // Счетчик включений светодиода USR1
    OnCounter : DWORD;
END_VAR
VAR
    prev_on : BOOL;
    // переменная-указатель для иллюстрации ошибки доступа к памяти
    ptr : POINTER TO DWORD;
END_VAR

// Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
SUPER^();

IF NOT prev_on AND SUPER^.blinker.OUT THEN
    // По переднему фронту блока "мигания" в базовом блоке
    // увеличиваем счетчик включений светодиода
    OnCounter := OnCounter + 1;
END_IF

(* ДОБАВЛЯЕМ КОД НЕПРАВИЛЬНОГО ДОСТУПА К ПАМЯТИ *)
IF OnCounter >= 20 THEN
    // Неявно инициализированный указатель содержит 0
    // или "мусор"
    ptr^ := OnCounter;
END_IF

// Запоминаем для последующего обнаружения переднего фронта
prev_on := blinker.OUT;
```

Таким образом, как только индикатор USR1 "мигнет" 20 (или более) раз, произойдет запись значения переменной *OnCounter* в память, адресуемую переменной *ptr*. Поскольку переменная *ptr* нигде явно не инициализирована, и ее значение по умолчанию равно 0, при записи по адресу 0 произойдет исключение, и контроллер перейдет в безопасный режим.

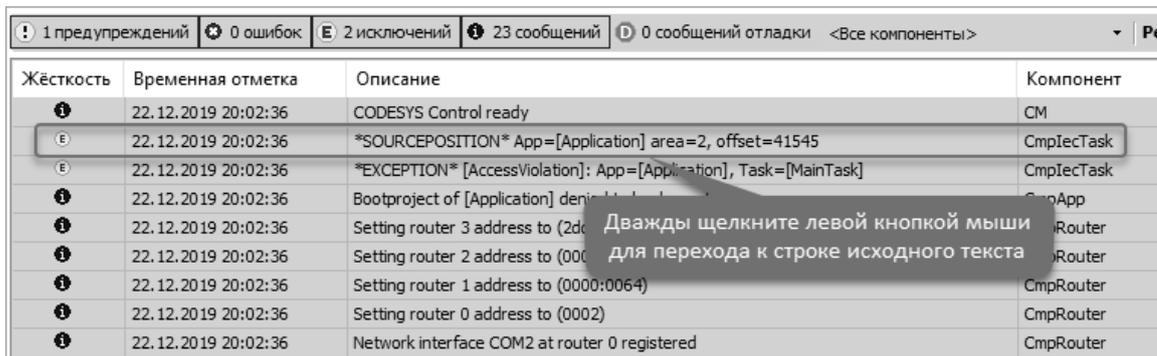
Выполните команду **Компиляция – Генерировать код** и загрузите приложение в контроллер методом полной загрузки, как указано в п. 3.11.2, нажмите клавишу F5 для запуска приложения, а затем выполните команду **Онлайн – Отключение**. Обратите внимание, что если среда разработки будет подключена к контроллеру во время перехода в безопасный режим, то по истечении заметного времени произойдет разрыв связи между средой разработки и контроллером. При этом до разрыва связи какие-либо манипуляции со средой разработки будут невозможными.

Примерно через 20 секунд контроллер перейдет в безопасный режим: индикатор RUN будет светиться прерывисто, меняя цвет с красного на зеленый и погасая, а индикатор APP будет светиться прерывисто зеленым цветом с частотой около 1 Гц.

3.12.3.3. Получение информации о местоположении неправильного доступа к памяти

Если контроллер функционирует в безопасном режиме по ошибке в приложении, для получения информации о местоположении ошибки в исходном тексте:

1. Если это не было сделано ранее, откройте проект, из которого было загружено приложение с ошибкой.
2. В дереве проекта дважды щелкните на элементе, соответствующем используемому контроллеру. В случае CPM810-03 откройте редактор устройства для элемента типа (*CPM810-03 Programmable Automation Controller*).
3. На вложенной вкладке **Установки соединения** редактора устройства нажмите **Сканировать сеть** и выберите сетевой путь к требуемому контроллеру, который перешел в безопасный режим, после чего нажмите **ОК**.
4. Перейдите на вложенную вкладку **Журнал** и нажмите кнопку  в его панели инструментов. Если контроллер доступен по выбранному сетевому пути, из него будут выгружены записи системного журнала, как показано на рисунке 79.



Жёсткость	Временная отметка	Описание	Компонент
	22.12.2019 20:02:36	CODESYS Control ready	CM
	22.12.2019 20:02:36	*SOURCEPOSITION* App=[Application] area=2, offset=41545	CmpIecTask
	22.12.2019 20:02:36	*EXCEPTION* [AccessViolation]: App=[Application], Task=[MainTask]	CmpIecTask
	22.12.2019 20:02:36	Bootproject of [Application] denied	CmpApp
	22.12.2019 20:02:36	Setting router 3 address to (2d...)	Router
	22.12.2019 20:02:36	Setting router 2 address to (00...)	Router
	22.12.2019 20:02:36	Setting router 1 address to (0000:0064)	CmpRouter
	22.12.2019 20:02:36	Setting router 0 address to (0002)	CmpRouter
	22.12.2019 20:02:36	Network interface COM2 at router 0 registered	CmpRouter

Рисунок 79 – Информация об исключении неправильного доступа к памяти

Информация об исключении неправильного доступа к памяти содержится строке, обозначенной символом , следующего содержания:

***EXCEPTION* [AccessViolation]: App=[имя приложения], Task=[имя задачи]**

Информация о местоположении исключения в коде приложения содержится в соседней (снизу вверх) строке, также обозначенной символом , следующего содержания:

***SOURCEPOSITION* App=[имя приложения] area=<номер области памяти>, offset=<смещение>**

5. Для перехода к строке исходного текста, при выполнении которой произошло исключение, дважды щелкните левой кнопкой мыши на строке  ***SOURCEPOSITION***, как показано на рисунке 79.

Среда разработки откроет программную единицу, в которой произошло исключение, и установит курсор на начало выражения, в котором произошел неправильный доступ к памяти, как показано на рисунке 80.

6. Для прекращения связи с системным журналом контроллера нажмите кнопку  в панели инструментов на вкладке **Журнал**, чтобы кнопка перестала отображаться в нажатом состоянии.

```
// Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
SUPER^ ();

IF NOT prev_on AND SUPER^.blinker.OUT THEN
  // По переднему фронту блока "мигания" в базовом блоке
  // увеличиваем счетчик включений светодиода
  OnCounter := OnCounter + 1;
END_IF

// Раскомментируйте для имитации зависания
IF OnCounter >= 20 THEN
  // Неявно инициализированный указатель содержит 0
  // или "мусор"
  ptr^ := OnCounter;
  //OnCounter := 1 / ( OnCounter - OnCounter);
END_IF
```

Рисунок 80 – Переход и отображение позиции в коде, где произошло исключение

7. Закомментируйте фрагмент кода, в котором допущена ошибка, приводящая к неправильному доступу к памяти, и повторно загрузите приложение в контроллер. После этого контроллер будет функционировать в нормальном режиме.
8. Выполните команду **Онлайн – Отключение** и сохраните проект командой **Файл – Сохранить**.

3.12.4. Поиск местоположения ошибки деления на 0 в коде приложения

3.12.4.1. Общие сведения

Система исполнения приложений генерирует исключение при выполнении операций деления '/' и получения остатка от деления MOD, если в качестве операндов используются целочисленные переменные, а в качестве делителя передается нулевое значение.

Деление на 0 в коде приложения обычно возникает непреднамеренно при использовании неинициализированных целочисленных переменных в качестве делителя или после преобразования значений переменных типа REAL или LREAL, близких к 0, в целочисленные и последующего использования их в качестве делителя в операциях деления.

Обратите внимание, что при делении на 0.0 переменных типа REAL или LREAL исключение не генерируется, а результат операции принимает значение NaN ("не число").

В данном подразделе в исходный текст учебном проекта будет намеренно внесена ошибка целочисленного деления на 0, а затем даны указания по определению признаков и местоположения ошибки.

3.12.4.2. Ошибка деления на 0 в учебном проекте

Для внесения ошибки доступа к памяти в код приложения учебного проекта откройте исходный текст функционального блока *USER_LED_EXT*, разработанного при выполнении указаний п. 3.11.2, в ST-редакторе и измените его реализацию следующим образом:

```
// Функциональный блок, расширяющий интерфейс и реализацию наследуемого
// библиотечного блока USER_LED_CONTROL
FUNCTION_BLOCK USER_LED_EXT EXTENDS FastwelBoard.USER_LED_CONTROL
VAR_OUTPUT
    // Счетчик включений светодиода USR1
    OnCounter : DWORD;
END_VAR
VAR
    prev_on : BOOL;
END_VAR

// Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
SUPER^();

IF NOT prev_on AND SUPER^.blinker.OUT THEN
    // По переднему фронту блока "мигания" в базовом блоке
    // увеличиваем счетчик включений светодиода
    OnCounter := OnCounter + 1;
END_IF

(* ДОБАВЛЯЕМ ДЕЛЕНИЕ НА 0 *)
IF OnCounter >= 20 THEN
    OnCounter := OnCounter / (OnCounter - OnCounter);
END_IF

// Запоминаем для последующего обнаружения переднего фронта
prev_on := blinker.OUT;
```

Таким образом, как только индикатор USR1 "мигнет" 20 (или более) раз, произойдет деление на 0, и контроллер перейдет в безопасный режим.

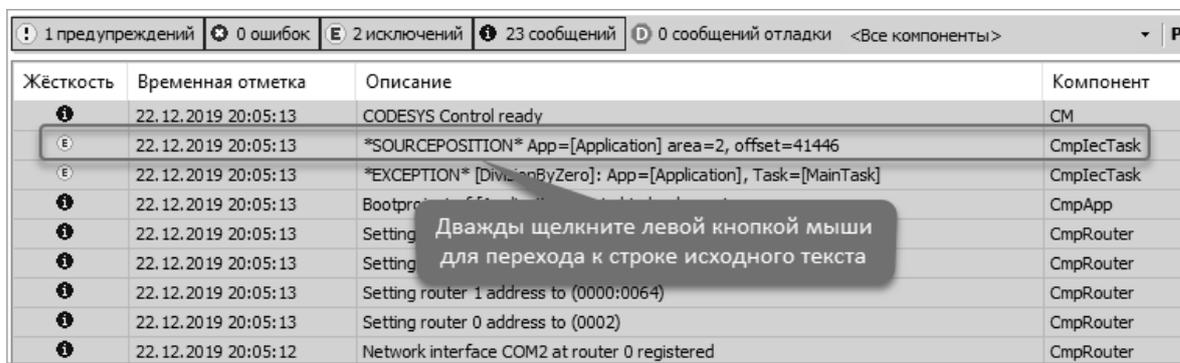
Выполните команду **Компиляция – Генерировать код** и загрузите приложение в контроллер методом полной загрузки, как указано в п. 3.11.2, нажмите клавишу F5 для запуска приложения, а затем выполните команду **Онлайн – Отключение**. Обратите внимание, что если среда разработки будет подключена к контроллеру во время перехода в безопасный режим, то по истечении заметного времени произойдет разрыв связи между средой разработки и контроллером. При этом до разрыва связи какие-либо манипуляции со средой разработки будут невозможными.

Примерно через 20 секунд контроллер перейдет в безопасный режим: индикатор RUN будет светиться прерывисто, меняя цвет с красного на зеленый и погасая, а индикатор APP будет светиться прерывисто зеленым цветом с частотой около 1 Гц.

3.12.4.3. Получение информации о местоположении ошибки деления на 0

Если контроллер функционирует в безопасном режиме по ошибке в приложении, для получения информации о местоположении ошибки в исходном тексте:

1. Если это не было сделано ранее, откройте проект, из которого было загружено приложение с ошибкой.
2. В дереве проекта дважды щелкните на элементе, соответствующем используемому контроллеру. В случае CPM810-03 откройте редактор устройства для элемента типа (*CPM810-03 Programmable Automation Controller*).
3. На вложенной вкладке **Установки соединения** редактора устройства нажмите **Сканировать сеть** и выберите сетевой путь к требуемому контроллеру, который перешел в безопасный режим, после чего нажмите ОК.
4. Перейдите на вложенную вкладку **Журнал** и нажмите кнопку  в его панели инструментов. Если контроллер доступен по выбранному сетевому пути, из него будут выгружены записи системного журнала, как показано на рисунке 81.



Жёсткость	Временная отметка	Описание	Компонент
	22.12.2019 20:05:13	CODESYS Control ready	CM
	22.12.2019 20:05:13	*SOURCEPOSITION* App=[Application] area=2, offset=41446	CmpIecTask
	22.12.2019 20:05:13	*EXCEPTION* [DivisionByZero]: App=[Application], Task=[MainTask]	CmpIecTask
	22.12.2019 20:05:13	Bootpro...	CmpApp
	22.12.2019 20:05:13	Setting ...	CmpRouter
	22.12.2019 20:05:13	Setting router 1 address to (0000:0064)	CmpRouter
	22.12.2019 20:05:13	Setting router 0 address to (0002)	CmpRouter
	22.12.2019 20:05:12	Network interface COM2 at router 0 registered	CmpRouter

Рисунок 81 – Информация об исключении при делении на 0

Информация об исключении при делении на 0 содержится строке, обозначенной символом , следующего содержания:

***EXCEPTION* [DivisionByZero]: App=[имя приложения], Task=[имя задачи]**

Информация о местоположении исключения в коде приложения содержится в соседней (снизу вверх) строке, также обозначенной символом , следующего содержания:

***SOURCEPOSITION* App=[имя приложения] area=<номер области памяти>, offset=<смещение>**

5. Для перехода к строке исходного текста, при выполнении которой произошло исключение, дважды щелкните левой кнопкой мыши на строке  ***SOURCEPOSITION***, как показано на рисунке 81.

Среда разработки откроет программную единицу, в которой произошло исключение, и установит курсор на начало выражения, в котором произведено деление на 0, как показано на рисунке 82.

```
// Вызываем код базового блока FastwelBoard.USER_LED_CONTROL
SUPER^ ();

IF NOT prev_on AND SUPER^.blinker.OUT THEN
  // По переднему фронту блока "мигания" в базовом блоке
  // увеличиваем счетчик включений светодиода
  OnCounter := OnCounter + 1;
END_IF

// Раскомментируйте для генерации ошибки деления на 0
(**)
IF OnCounter >= 20 THEN
  OnCounter := OnCounter / (OnCounter - OnCounter);
END_IF
(**)
```

Рисунок 82 – Переход и отображение позиции в коде, где произошло исключение

6. Для прекращения связи с системным журналом контроллера нажмите кнопку  в панели инструментов на вкладке **Журнал**, чтобы кнопка перестала отображаться в нажатом состоянии.
7. Закомментируйте фрагмент кода, в котором допущена ошибка, приводящая к делению на 0, и повторно загрузите приложение в контроллер.
После этого контроллер будет функционировать в нормальном режиме.
8. Выполните команду **Онлайн – Отключение** и сохраните проект командой **Файл – Сохранить**.

4. Принцип работы программного обеспечения контроллера

4.1. Общие сведения

СРМ810-03 является программируемым контроллером с переменным составом модулей и содержит установленное при производстве системное программное обеспечение, состоящее из операционной системы реального времени, драйверов устройств, системных сервисов и системы исполнения приложений (прикладных программ).

Приложение должно разрабатываться конечным пользователем контроллера в IDE МЭК 61131-3 и содержать реализацию прикладных алгоритмов на языках стандарта ГОСТ Р МЭК 61131-3, конфигурацию модулей ввода-вывода Fastwel I/O-2 и/или Fastwel I/O, подключенных к локальной и удаленным межмодульным шинам контроллера, а также сервисов сетевых протоколов прикладного уровня MODBUS TCP, MODBUS и сервисов сетевых протоколов прикладного уровня.

Настоящий раздел содержит информацию о принципах функционирования системного программного обеспечения контроллера.

4.2. Режимы работы контроллера

4.2.1. Режим с исходными (заводскими) настройками

При поставке контроллер не содержит приложения пользователя и при включении питания запускается в так называемом безопасном режиме с заводскими настройками (далее *заводском режиме*), о чем свидетельствует попеременное свечение индикатора RUN зеленым и красным цветами с периодическим погасанием при отсутствии свечения индикаторов APP, I/O, USR1 и USR2.

В *заводском режиме* системное программное обеспечение контроллера ожидает загрузки приложения пользователя и настройки параметров сетевых интерфейсов и сервисов. Сетевые интерфейсы контроллера в заводском режиме имеют следующие исходные параметры:

IP-адрес интерфейса LAN1: 10.0.0.100/8

IP-адрес интерфейса LAN2: 10.0.0.101/8

Если неизвестны параметры сетевых интерфейсов контроллера, или контроллер не обнаруживается коммуникационными средствами IDE МЭК 61131-3 ни по сети Ethernet, ни через последовательный порт COM2, возможно принудительно перевести контроллер в заводской режим путем включения всех десяти переключателей, как показано на рисунке 83, и последующим перезапуском нажатием кнопки сброса (см. рисунок 1) или выключением и повторным включением питания.

Заводской режим



LAN1: 10.0.0.100/8

LAN2: 10.0.0.101/8

Рисунок 83 – Положение переключателей для принудительного запуска контроллера в заводском режиме

Если контроллер принудительно запущен в заводском режиме, имеется возможность загрузить в него новое приложение и/или настроить параметры системных сервисов в веб-конфигураторе, после чего вернуть переключатели в положение, которое было установлено до перехода в заводской режим, и перезапустить контроллер выключением/включением питания, нажатием кнопки сброса или командой *reboot* в оболочке ПЛК IDE МЭК 61131-3.

В меню **Онлайн** IDE МЭК 61131-3 имеется команда **Сброс заводской**. Данная команда удаляет ранее загруженное приложение, но не изменяет параметры системных сервисов контроллера, настраиваемые в веб-конфигураторе, за исключением паролей учетных записей *Administrator* и *Everyone* – учетным записям возвращаются исходные пароли (*Administrator* и пустой соответственно).

Обратите внимание, что при принудительном запуске контроллера в заводском режиме (все переключатели включены) ранее установленные пароли учетных записей *Administrator* и *Everyone* не изменяются на исходные, т.е. нет возможности получить несанкционированный доступ к проектной информации, загруженной в контроллер. Более подробная информация об учетных записях *Administrator* и *Everyone* приведена в руководстве пользователя Fastwel PLC Application Toolkit.

4.2.2. Нормальный режим

Если в контроллере имеется приложение пользователя, загруженное из IDE МЭК 61131-3, переключатели 1–8 контроллера установлены в любое положение, кроме 255 (все восемь включены), то при включении питания или перезапуске контроллер будет функционировать в нормальном режиме.

В нормальном режиме выполняются основные функции контроллера, включая исполнение задач и обработчиков системных событий, входящих в приложение, обмен данными с модулями ввода-вывода, обслуживание запросов протоколов прикладного уровня, поступающих сети, формирование сетевых запросов и т.д. При этом индикаторы контроллера будут светиться следующим образом:

RUN:

зеленый цвет (непрерывно)

1. если в приложении имеется единственная циклическая задача, и она хотя бы иногда успевает укладываться в заданный период;
2. если в приложении имеется более одной циклической задачи, и хотя бы одна из них хотя бы иногда успевает укладываться в заданный период.

Укладываться в заданный период означает, что все программы, исполняемые под управлением данной задачи, заканчивают свою работу на очередном цикле до наступления времени начала следующего цикла и запускаются повторно строго в запланированный момент начала следующего очередного цикла.

зеленый цвет (прерывисто)

приложение остановлено после загрузки из среды разработки, командой **Отладка – Стоп** или на точке останова (в системном программном обеспечении контроллера версии 2.0.3.0 и выше)

красный цвет – если в приложении имеется более одной циклической задачи, и ни одна из них никогда не успевает укладываться в заданный период.

APP:

зеленый цвет – все циклические задачи всегда успевают укладываться в заданный период;

красный цвет (непрерывно) – все циклические задачи никогда не успевают укладываться в заданный период;

красный цвет (прерывисто) – хотя бы одна циклическая задача иногда успевает укладываться в заданный период;

зеленый цвет (прерывисто) – одна циклическая задача из нескольких иногда не успевает укладываться в заданный период.

I/O:

зеленый цвет (непрерывно) – все межмодульные шины (локальная и удаленные), определенные в конфигурации загруженного приложения, функционируют в *полностью исправном состоянии*, т.е. все модули ввода-вывода, определенные в конфигурации проекта для элементов *Modules (Master FBUS)*, относящихся к локальному и удаленному портам межмодульной шины FBUS, обнаружены и успешно

сконфигурированы. Обмен с модулями ввода-вывода, подключенными к контроллеру, успеваает завершиться за время, установленное в качестве параметра **Период опроса, мс** в конфигурации сервиса ввода-вывода контроллера, при загрузке шины не более 50%;

зеленый цвет (прерывисто) – все межмодульные шины, определенные в конфигурации загруженного приложения, функционируют в *полностью исправном состоянии*, но обмен с модулями ввода-вывода, подключенными к контроллеру, не может быть выполнен за время, заданное в конфигурации контроллера для элементов *Modules (Master FBUS)*, относящихся к локальному и удаленному портам межмодульной шины FBUS, в связи с чем используется расчетное минимально достижимое время обмена данными со всеми модулями при загрузке внутренней шины на 50%, отображаемое в поле **Рекомендуемый период опроса, мс**;

зеленый цвет периодически меняется на красный – хотя бы одна из межмодульных шин, определенных в конфигурации приложения, функционирует в *частично исправном состоянии*, т.е. состав и конфигурация обнаруженных модулей ввода-вывода соответствуют заданным в конфигурации приложения, загруженного в контроллер, за исключением некоторых модулей, которых нет среди обнаруженных хотя бы на одной из шин;

красный цвет – конфигурация модулей ввода-вывода, определенная в загруженном приложении, ни на одной из межмодульных шин (локальной и удаленных) не совпадает с аппаратной конфигурацией модулей, подключенных к контроллеру, что является условием работы шины в *неисправном состоянии*. Кроме того, свечение красным сразу после загрузки нового приложения свидетельствует о выполнении конфигурирования сервиса ввода-вывода;

отсутствие свечения – в конфигурации загруженного приложения отсутствуют описания модулей ввода-вывода.

Информация об индикации контроллера приведена в документе *ИМЕС.421459.252РЭ1. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 2. Составные части комплекса*.

Информация об индикации контроллера и возможных состояниях шины FBUS приведена в документе *ИМЕС.421459.252РЭ. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 1. Общие сведения*.

4.2.3. Безопасный режим

Контроллер переходит в безопасный режим, если при запуске или во время функционирования приложения произошел отказ в исполняемом коде загруженного приложения, или отказ в системе исполнения, связанный с исполнением текущего загруженного приложения.

В безопасном режиме индикатор RUN циклически меняет свой цвет с зеленого на красный и погасает, а индикатор APP светится прерывисто красным или зеленым цветом в зависимости от вида отказа:

1. Прерывистое свечение зеленым цветом с частотой около 1 Гц – исключение в коде приложения, загруженного в контроллер из среды разработки, для которого определено местоположение ошибки в исходном тексте приложения.
2. Прерывистое свечение красным цветом с частотой около 1 Гц – исключение в системе исполнения или в коде приложения, загруженного в контроллер из среды разработки, для которого отсутствует информация о местоположении ошибки в исходном тексте приложения.
3. Прерывистое свечение зеленым цветом с частотой около 2 Гц – контроллер переведен в форсированный безопасный режим из приложения путем вызова функции `FwPlatformReset` из библиотеки `FastwelCore` с параметром `F_RESET_SAFE`. Информация о библиотеке `FastwelCore` приведена в руководстве пользователя `Fastwel PLC Application Toolkit`.

При изложении последующих указаний по определению причины перехода контроллера в безопасный режим предполагается, что среда разработки IDE МЭК 61131-3 настроена для взаимодействия с контроллером через порт COM2 или по сети Ethernet в соответствии с указаниями руководства пользователя `Fastwel PLC Application Toolkit`.

Для определения причины перехода в безопасный режим:

1. Откройте проект, из которого было загружено приложение контроллера, или создайте новый проект для платформы *CPM810-03 Programmable Automation Controller*.
2. В дереве проекта дважды щелкните на элементе, соответствующем используемому контроллеру. В случае CPM810-03 откройте редактор устройства для элемента типа (*CPM810-03 Programmable Automation Controller*).
3. На вложенной вкладке **Установки соединения** редактора устройства нажмите **Сканировать сеть** и выберите сетевой путь к контроллеру, который перешел в безопасный режим, после чего нажмите **ОК**. Идентификация контроллера может быть произведена по серийному (заводскому) номеру контроллера или нажатием кнопки **Помогать** в окне **Выбор устройства** в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.
4. Перейдите на вложенную вкладку **Оболочка ПЛК (PLC Shell)** и в поле ввода, расположенном в нижней части вкладки, введите команду *pinf*. Если контроллер функционирует в безопасном режиме, результат выполнения команды *pinf* будет содержать информацию, подобную следующей:

Device: CPM810-03 Programmable Automation Controller (810.0006 2019/12)

Firmware Version: 1.0.0.0

Mode: SAFE

Reason:

EXCEPTION [<excpt>AccessViolation</excpt>] occurred: App=[<app>Application</app>], Task=[<task>MainTask</task>]

SOURCEPOSITION App=[<app>Application</app>] area=<area>2</area>, offset=<off>21364</off>

End of Project Information.

Поле *Device* содержит информацию о типе устройства, а также серийный (заводской) номер и дату изготовления.

Поле *Firmware Version* содержит информацию о версии системного программного обеспечения контроллера.

В поле *Mode* указан текущий режим работы контроллера: *NORMAL* – для нормального режима, *SAFE* – для безопасного режима.

При работе контроллера в безопасном режиме, поле *Reason* содержит причину запуска контроллера в безопасном режиме.

Если в информации о причине запуска в безопасном режиме имеется строка, начинающаяся с префикса **SOURCEPOSITION**, а в IDE МЭК 61131-3 открыт проект, который был загружен в контроллер до его перехода в безопасный режим, то последующие указания позволяют установить точное место в исходном тексте приложения, в котором произошел отказ.

5. На вкладке редактора устройства перейдите на вложенную вкладку **Журнал** и нажмите кнопку  в его панели инструментов. Если контроллер доступен по выбранному сетевому пути, из него будут выгружены записи системного журнала, как показано на рисунке 77 (рисунке 79 или 81).

Информация об исключении содержится строке, обозначенной символом , следующего содержания:

EXCEPTION [*тип исключения*] occurred: App=[*имя приложения*], Task=[*имя задачи*]

Информация о местоположении исключения в коде приложения содержится в соседней (снизу вверх) строке, также обозначенной символом , следующего содержания:

SOURCEPOSITION App=[*имя приложения*] area=<*номер области памяти*>, offset=<*смещение*>

6. Если в среде разработки открыт проект, содержащий приложение, которое было ранее загружено в контроллер, то для перехода к строке исходного текста, при выполнении которой произошло исключение, дважды щелкните левой кнопкой мыши на строке  ***SOURCEPOSITION***.

Среда разработки откроет исходный текст программной единицы и установит курсор на начало выражения, в котором был зафиксирован отказ.

Примеры определения причин перехода контроллера в безопасный режим из-за отказов, связанных с закливанием, неправильным доступом к памяти и целочисленному делению на 0, приведены в п. 3.12.

Если по имеющейся диагностической информации не удастся определить причину перехода контроллера в безопасный режим, выполните следующие действия:

1. Откройте проект, из которого было загружено приложение контроллера, или создайте новый проект для платформы *CPM810-03 Programmable Automation Controller*.
2. В дереве проекта дважды щелкните на элементе, соответствующем используемому контроллеру. В случае CPM810-03 откройте редактор устройства для элемента типа (*CPM810-03 Programmable Automation Controller*).
3. На вложенной вкладке **Установки соединения** редактора устройства нажмите **Сканировать сеть** и выберите сетевой путь к контроллеру, который перешел в безопасный режим, после чего нажмите **ОК**. Идентификация контроллера может быть произведена по серийному (заводскому) номеру контроллера или нажатием кнопки **Помогать** в окне **Выбор устройства** в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.
4. Перейдите на вложенную вкладку **Файлы** редактора устройства и нажмите кнопку , как показано на рисунке 84.

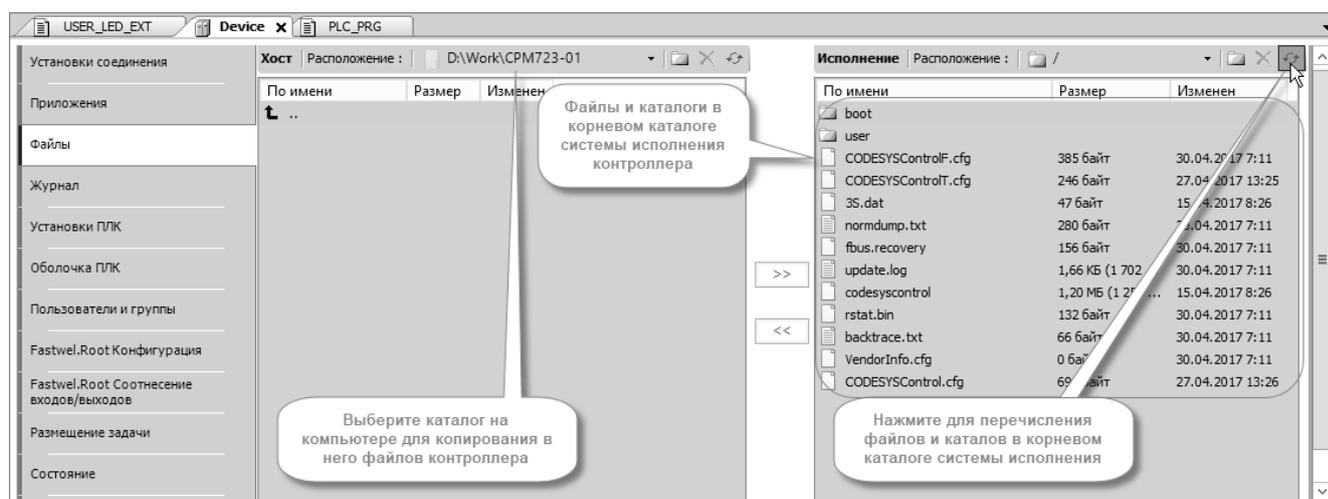


Рисунок 84 – Доступ к файлам в корневом каталоге системы исполнения контроллера

5. Среди файлов в корневом каталоге системы исполнения выберите *normdump.txt* и, при наличии, файл *safedump.txt*, после чего скопируйте их в выбранный каталог ПК, отображаемый в области **Хост**, нажатием кнопки .
- Для выбора нескольких файлов щелкните левой кнопкой мыши над их именами, удерживая нажатой клавишу Ctrl.
6. Сохраните проект, который был загружен в контроллер до перехода в безопасный режим, командой **Файл – Архив проекта – Сохранить/отправить архив...** в главном меню IDE МЭК 61131-3.
7. Отправьте выгруженные из контроллера файлы *normdump.txt*, *safedump.txt* и файл архива с расширением *projectarchive*, сформированный средой разработки, по электронной почте на адрес fo@fastwel.ru.

В сообщении, отправляемом по электронной почте, укажите:

название предприятия,
 фамилию и инициалы,
 фактическую аппаратную конфигурацию контроллера (тип контроллера, серийный номер, версию системного ПО контроллера, номенклатуру модулей ввода-вывода в порядке подключения к контроллеру и тип используемого источника питания),
 обстоятельства, при которых контроллер перешел в безопасный режим.

При наличии связи по сети Ethernet выгрузка файлов из контроллера может быть произведена по протоколу FTP/SFTP/FTPS в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.

4.2.4. Получение информации о режиме работы контроллера по сети

4.2.4.1. Общие сведения

Начиная с версии 3.2.3.0 системного ПО контроллера, имеется возможность получения информации о текущем режиме работы контроллера по сети через встроенный сервер OPC UA, а также через любой из встроенных серверов MODBUS (MODBUS TCP, RTU, ASCII).

4.2.4.2. Получение информации о режиме работы контроллера по MODBUS

Для всех экземпляров сервиса подчиненного узла протокола MODBUS (MODBUS TCP, MODBUS RTU/ASCII) автоматически создается специальный входной регистр (Input Register, запрос чтения 4) с номером 65536 (адрес в запросе чтения 65535), который содержит данные о текущем режиме работы контроллера в следующем формате:

- младший байт регистра содержит признак текущего режима: 0 – нормальный, 1 – безопасный;
- старший байт содержит нулевое значение в безопасном режиме или код статуса системы исполнения в нормальном режиме: 1 – приложения выполняется, 2 – приложение остановлено по команде из среды разработки или из кода приложения, 3 – приложение остановлено на точке останова.

Обратите внимание, что если в конфигурации приложения не определено ни одного экземпляра сервиса подчиненного узла MODBUS, то информация о режиме работы контроллера по сети MODBUS недоступна.

Кроме того, специальный входной регистр с номером 65536 не создается, если входной регистр с таким номером уже определен в конфигурации сервиса подчиненного узла MODBUS приложения.

Информация о сервисе подчиненного узла протокола MODBUS приведена в п. 4.6.3.

4.2.4.3. Получение информации о режиме работы контроллера через OPC UA

Если в контроллере активирован встроенный сервер OPC UA, то информация о текущем режиме работы контроллера может быть получена у сервера OPC UA в элементе *Server – ServerStatus – State*, показанном на рисунке 85.

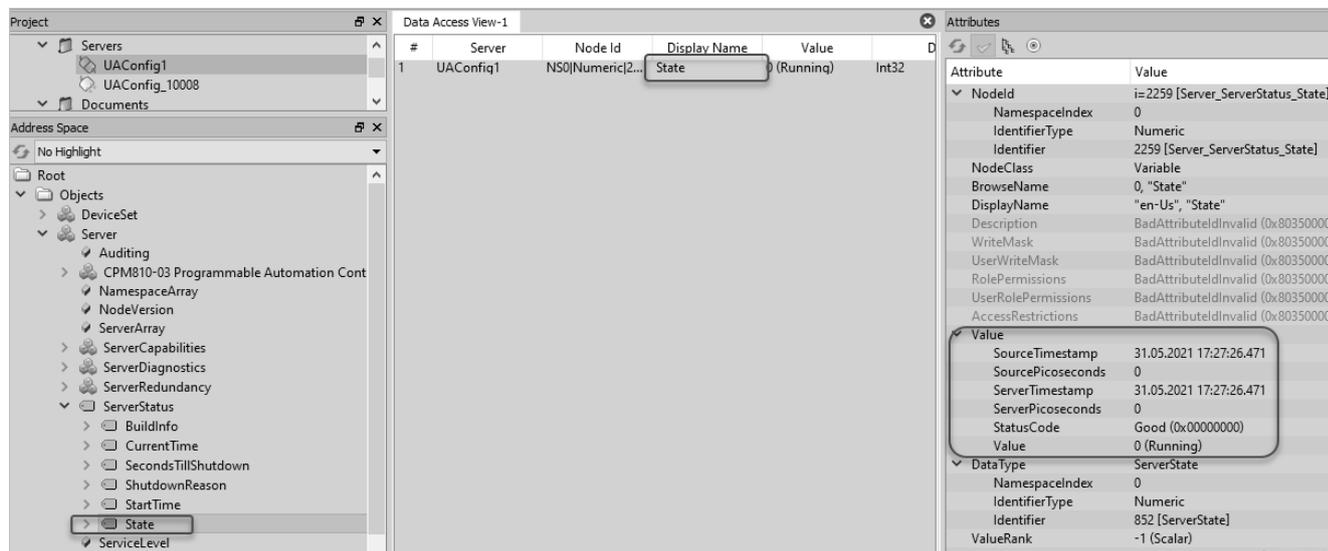


Рисунок 85 – Элемент *Server – ServerStatus – State* в адресном пространстве сервера OPC UA

Данный элемент имеет тип Int32 и может принимать следующие значения:

- 0 – нормальный режим;
- 1 – безопасный режим по ошибке в ранее загруженном приложении;
- 2 – безопасный режим без приложения (после заводского сброса из среды разработки).

Информация о встроенном сервере OPC UA приведена в руководстве пользователя Fastwel PLC Application Toolkit.

4.3. Процесс запуска контроллера

4.3.1. Запуск при первом включении питания

При поставке контроллер не содержит пользовательского приложения и при включении питания запускается в безопасном режиме с исходными (заводскими) настройками, о чем свидетельствует попеременное свечение индикатора RUN зеленым и красным цветами, и прекращение свечения, при отсутствии свечения индикаторов APP, I/O, USR1 и USR2. В безопасном режиме системное программное обеспечение контроллера не обращается к модулям ввода-вывода и ожидает загрузки в контроллер приложения, разработанного в IDE МЭК 61131-3.

Более подробная информация об условиях запуска контроллера в заводском режиме приведена в п. 4.2.1.

4.3.2. Запуск при наличии загруженного приложения

Если перед последним выключением питания в контроллер было загружено приложение из IDE МЭК 61131-3, после включения питания или перезагрузки выполняются следующие действия:

1. Системный начальный загрузчик, расположенный по фиксированному адресу встроенного NAND флэш-накопителя контроллера, загружает и передает управление операционной системе на активном разделе данного накопителя.
2. В пользовательском разделе NAND флэш-накопителя или на карте microSD проверяется наличие файла обновления системного программного обеспечения контроллера *norm.dnl* и, при наличии файла, выполняется развертывание и обновление системного программного обеспечения. Пользовательский раздел NAND флэш-накопителя, в котором расположены файлы приложения IDE МЭК 61131-3 и конфигурации, при обновлении остается неизменным, чтобы после обновления контроллер продолжил функционирование с ранее загруженным приложением и заданными ранее параметрами системных сервисов.

В процессе обновления системного ПО индикаторы RUN, APP, I/O и USR1 светятся желтым цветом. При успешном завершении обновления указанные индикаторы светятся зеленым цветом.

3. В пользовательском разделе NAND флэш-накопителя в корневом каталоге системы исполнения приложений МЭК 61131-3 или на карте microSD проверяется наличие файла развертывания приложения и конфигурации системных сервисов контроллера *norm.upl*, подготовленного командой оболочки ПЛК *saveapp*. При обнаружении файла *norm.upl* выполняется развертывание приложения и конфигурации, в процессе чего индикатор APP светится желтым цветом.
4. В пользовательском разделе NAND флэш-накопителя или на карте microSD проверяется наличие файла обновления микропрограмм модулей ввода-вывода *ffw.dnl* и, при наличии файла, выполняется обновление микропрограмм модулей, в процессе которого индикатор I/O светится желтым цветом.
5. Выполняется установка и настройка системных сервисов контроллера, в течение которой индикатор RUN светится непрерывно желтым цветом.
6. Запускается процесс системы исполнения приложений, который открывает файл загруженного приложения, расположенный в каталоге boot корневого каталога системы исполнения, находит и конфигурирует модули ввода-вывода, определенные в конфигурации приложения, запускает сервисы протокола MODBUS TCP и MODBUS, если они присутствуют в конфигурации приложения, а затем запускает циклические и ациклические задачи, которые начинают выполнять пользовательский код программ из списков вызова циклических и ациклических задач.
7. Запускаются сетевые сервисы контроллера. Если сетевые интерфейсы контроллера настроены на работу в режиме динамического назначения IP-адресов (переключатели 1–8 установлены в положение 254 или в веб-конфигураторе для одного или обоих интерфейсов установлен режим DHCP при выключенных переключателях 1–8), системный сервис DHCP запрашивает IP-адреса и ожидает их получения от сервера DHCP, который должен быть физически доступен в сегменте сети, к которому подключен контроллер.

При успешном выполнении перечисленных выше операций контроллер функционирует в нормальном режиме, выполняя пользовательские алгоритмы, реализованные в загруженном приложении. Более подробная информация об индикации нормального режима приведена в п. 4.2.2 настоящего документа.

Системный сервис сторожевого таймера активизируется при включении питания контроллера и перезапускает аппаратный сторожевой таймер контроллера с периодом около 1 секунды. В случае непредвиденного останова микропроцессора контроллер будет перезапущен аппаратным сторожевым таймером не позднее, чем через 5 секунд.

4.4. Система исполнения приложений МЭК 61131-3

4.4.1. Общие сведения

Информация о программной модели приложения контроллера, принципе обмена данными приложения с внешними устройствами, способах загрузки приложения в контроллер и о создании конфигурации сервера OPC UA приведена в руководстве пользователя Fastwel PLC Application Toolkit.

4.4.2. Использование подсистемы целевой визуализации (Target Visualization)

4.4.2.1. Общие сведения

При создании форм визуализации в проектах для СРМ810-03 в дерево проекта будет автоматически добавляться элемент *Target-визуализация (Target Visualization)*.

Если проект для СРМ810-03 разрабатывается на основе проекта для другого целевого устройства, не имеющего в своем составе видеоподсистему, например, СРМ723-01, содержащего формы визуализации, путем замены одного типа целевого устройства на другое операцией **Обновить устройство**, то для того, чтобы формы визуализации отображались на экране монитора, подключенного к VGA-порту контроллера, следует добавить в приложение элемент *Target-визуализация (Target Visualization)*, как показано на рисунке 86.

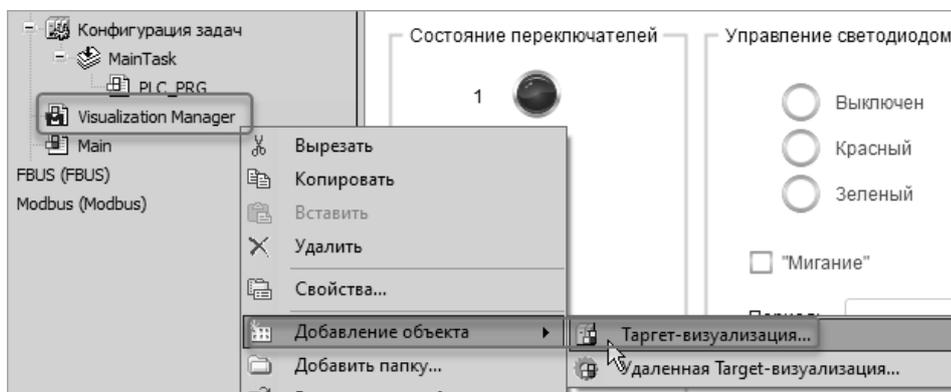


Рисунок 86 – Добавление менеджера целевой визуализации вручную

После добавления данного элемента в приложении появятся дополнительные задачи и программные единицы для отображения форм визуализации и обработки событий от указательных устройств и устройств ввода.

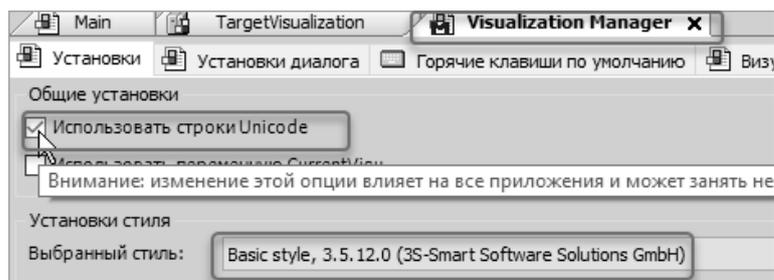


Рисунок 87 – Включение поддержки строк в кодировке Unicode в формах визуализации

При необходимости отображения в формах визуализации символов кириллицы следует убедиться, что в *Менеджере визуализации (Visualization Manager)* отмечен флажок **Общие установки – Использовать строки Unicode**, как показано на рисунке 87.

Использование форм визуализации в проектах IDE МЭК 61131-3 для контроллера CPM810-03 иллюстрируется в примерах программирования, поставляемых в Fastwel PLC Application Toolkit.

4.4.2.2. Активация подсистемы целевой визуализации

Для работы приложения с целевой визуализацией в контроллере следует разрешить использование видеоподсистемы в веб-конфигураторе, отметив флажок **Разрешить видео** на странице **Видео**, как показано на рисунке 88, и нажать **Применить конфигурацию**.

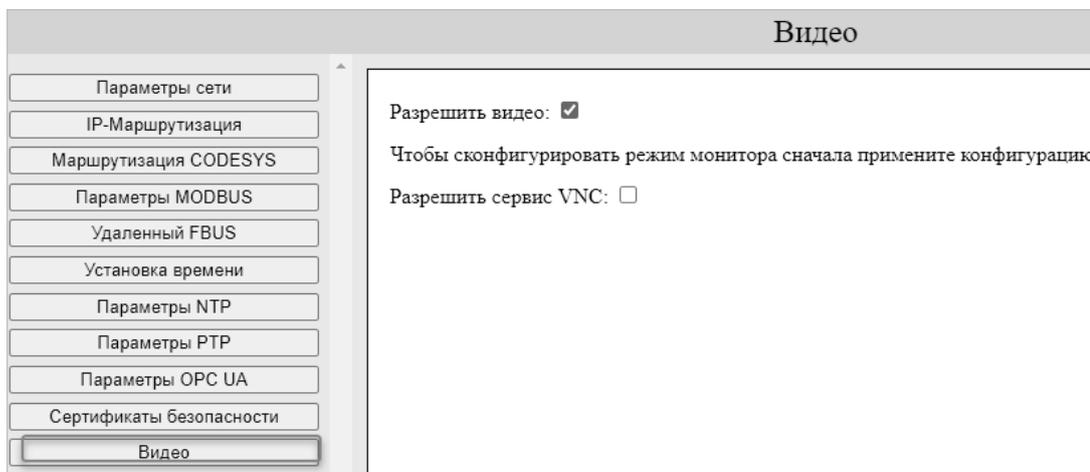


Рисунок 88 – Включение видеоподсистемы в веб-конфигураторе

После перезапуска необходимо в веб-конфигураторе установить требуемое разрешение экрана, как показано на рисунке 89, и вновь нажать **Применить конфигурацию**.

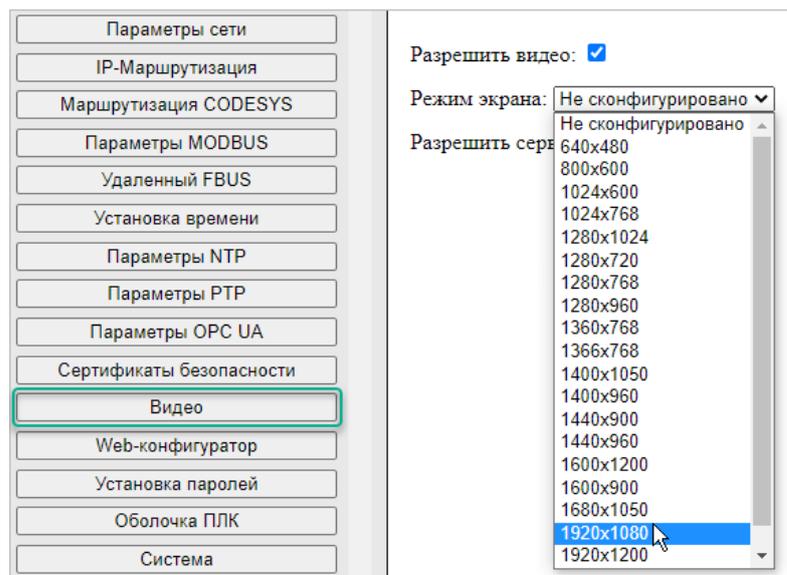


Рисунок 89 – Установка разрешения экрана в веб-визуализации

Обратите внимание, что для монитора будет всегда устанавливаться оптимальное максимальное разрешение, полученное видеоподсистемой контроллера при взаимодействии с монитором.

Если в веб-конфигураторе установлено большее разрешение экрана, чем поддерживается монитором, то часть изображения на экране будет выходить за пределы видимой области, и просмотр изображения за пределами видимой области станет возможным путем перемещения мыши к границам видимой области.

Если установить меньшее разрешение, то изображение на экране будет масштабировано, что приведет к пропорциональному увеличению размера каждого пикселя исходного изображения.

Дополнительная информация о настройке системных параметров видеоподсистемы и подсистемы целевой визуализации приведена в п. 5.12.

4.4.2.3. Удаленный доступ к экранам форм целевой визуализации

В составе системного программного обеспечения контроллера имеется сервис VNC, обеспечивающий возможность подключения к контроллеру по сети и просмотру и управлению содержимым экранов форм визуализации на экране ПК при помощи приложения VNC Viewer, TightVNC Viewer или аналогичных, как показано на рисунке 90.

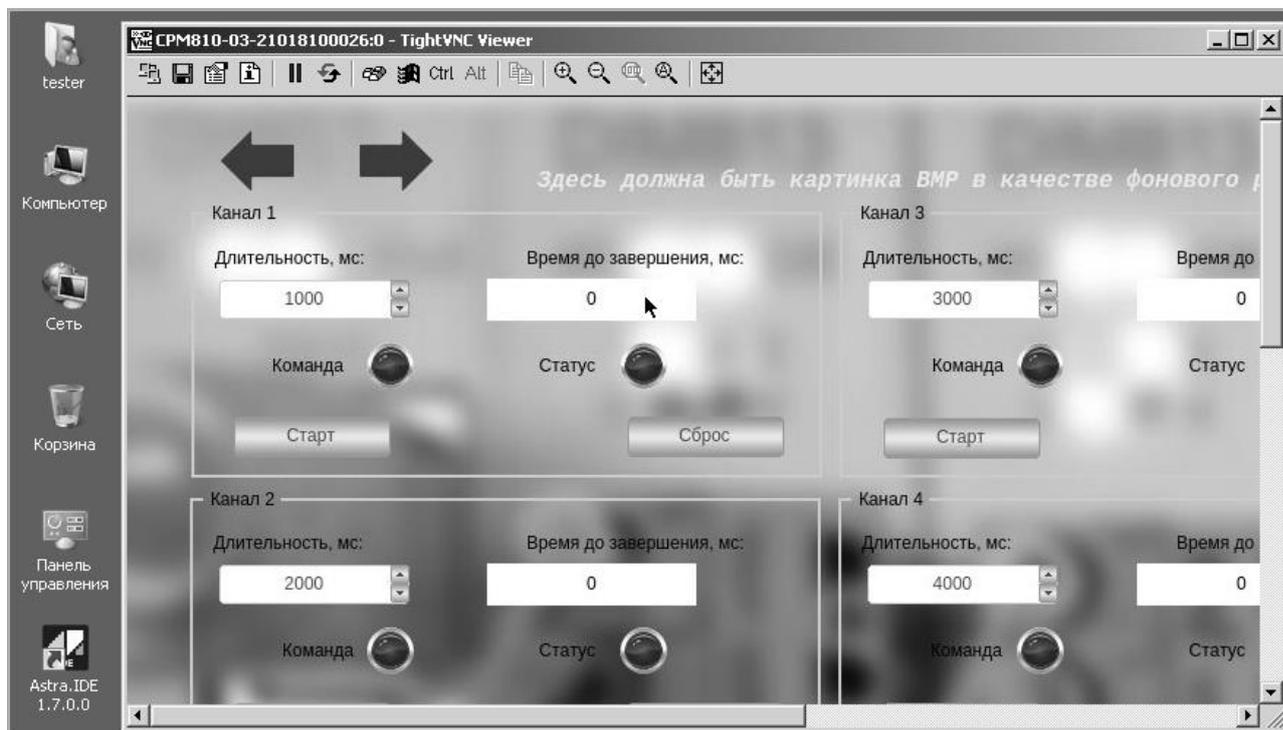


Рисунок 90 – Экранная форма целевой визуализации контроллера в окне TightVNC Viewer

Отображение экранов форм целевой визуализации в окне клиента VNC полностью идентично тому, как они отображаются на экране монитора, поэтому данная функция может использоваться для отладки работы экранов форм целевой визуализации на ПК.

Для включения удаленного доступа к экрану контроллера в веб-конфигураторе на странице **Видео** следует отметить флажок **Разрешить сервис VNC** и нажать **Применить конфигурацию**.

ВНИМАНИЕ!

Не рекомендуется оставлять постоянный доступ к контроллеру по протоколу VNC, т.к. соединение по данному протоколу не защищено паролем.

4.4.3. Диагностические каналы системы исполнения

Во входном образе процесса приложения для контроллера CPM810-03 имеются специальные каналы, показанные на рисунке 91, которые позволяют контролировать некоторые аспекты функционирования приложения.

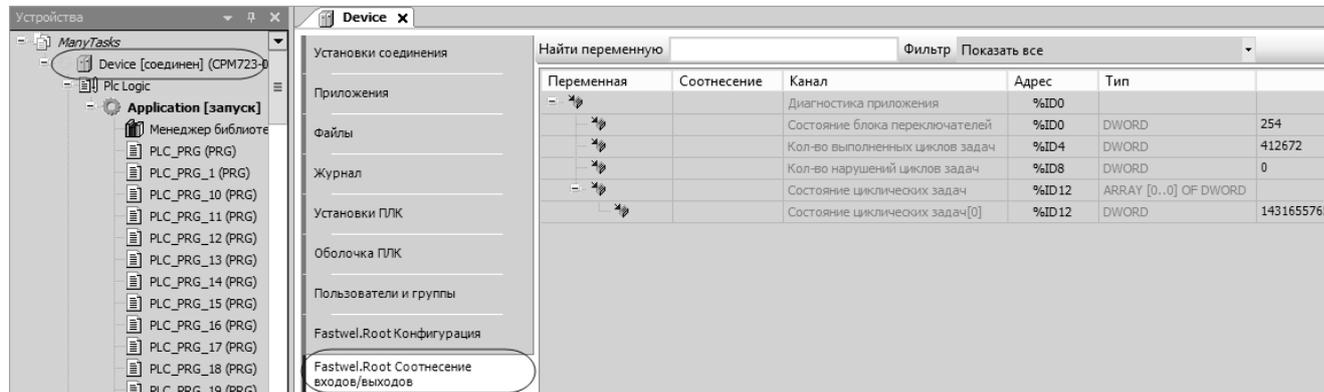


Рисунок 91 – Диагностические каналы системы исполнения

Канал *Состояние блока переключателей* типа DWORD отображает текущее положение переключателей 1...10 контроллера. Логическая 1 в битовой позиции с номером n, начиная с 0, соответствует нахождению переключателя с номером n+1 во включенном положении.

Канал *Кол-во выполненных циклов задач* типа DWORD содержит суммарное количество циклов, выполненных циклическими задачами приложения.

Канал *Кол-во нарушений циклов задач* типа DWORD содержит суммарное количество запаздываний циклических задач к началу выполнения своих очередных циклов.

Канал *Состояние циклических задач* типа DWORD содержит текущие статусы каждой из 16-ти циклических задач в смежных парах битовых позиций:

- 0: задача не сконфигурирована;
- 1: задача активна и успевает укладываться в заданный период;
- 2: задача активна и не всегда успевает укладываться в заданный период;
- 3: задача активна и никогда не успевает укладываться в заданный период.

Диагностические каналы обновляются сервисной задачей системы исполнения с частотой 1 Гц (один раз в секунду).

4.5. Сервис ввода-вывода

4.5.1. Общие сведения

Сервис ввода-вывода системы исполнения контроллера реализует стек протоколов межмодульной шины FBUS 2.0.

Шина FBUS является системой последовательной передачи данных, которая предназначена для организации обмена данными реального времени и конфигурационной информацией между вычислительным устройством (мастером шины) и периферийными модулями (подчиненными узлами).

Физический уровень и топология шины FBUS представлены на рисунке 92.



Рисунок 92 – Физический уровень и топология шины FBUS

Обмен данными между мастером и подчиненными узлами выполняется со скоростью 2 Мбит/с. Подробная информация об остальных уровнях протокола FBUS выходит за рамки настоящего документа.

Контроллер СРМ810-03 обеспечивает возможность работы с тремя наборами модулей ввода-вывода Fastwel I/O-2 и/или Fastwel I/O, каждый из которых может состоять из 64 активных модулей. Модули первого набора подключаются непосредственно к локальной шине FBUS контроллера, а два других набора – к модулям интерфейсным NIM745-01, которые должны быть доступны контроллеру по сети Ethernet. Модуль NIM745-01 является удаленным адаптером шины FBUS.

ВНИМАНИЕ!

Контроллер имеет один локальный порт шины FBUS, выведенный на два соединителя: RJ-45 – на передней панели и показанный на рисунке 2. Не допускается одновременное подключение двух наборов модулей ввода-вывода к двум соединителям шины.

Сервис ввода-вывода обеспечивает выполнение следующих функций:

1. Инициализацию шины и конфигурирование модулей ввода-вывода.
2. Обмен данными реального времени с модулями ввода-вывода.
3. Обработку нештатных ситуаций, связанных с потерей связи с одним или несколькими модулями ввода-вывода.
4. Обновление диагностической информации, доступной прикладной программе, и светодиодную индикацию.

Описание устройства, принципа работы и состояний межмодульной шины FBUS приведено в документах:

ФАПИ.421459.700РЭ. Fastwel I/O. Распределённая система ввода-вывода. Руководство по эксплуатации

ИМЕС.421459.252РЭ. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 1. Общие сведения.

4.5.2. Конфигурация сервиса ввода-вывода

4.5.2.1. Параметры сервиса ввода-вывода

В контроллерах Fastwel I/O обмен данными между контроллером и модулями ввода-вывода по умолчанию осуществляется в так называемом групповом режиме, при котором за одну сетевую транзакцию одновременно осуществляется запись данных во все выходные каналы и чтение данных всех входных каналов модулей ввода-вывода. Тем самым обеспечивается пропускная способность не хуже 165 кбайт/с. Для выбора группового режима обмена параметр **Режим обмена**, устанавливаемый в редакторе устройства (*Мастер FBUS*), должен иметь значение *Групповой*, как показано на рисунке 93.

Кроме того, предусмотрен режим индивидуального обмена с модулями, при использовании которого для каждого модуля, добавленного в конфигурацию контроллера, мастером шины формируется отдельный запрос ввода-вывода. Для выбора режима индивидуального обмена параметр **Режим обмена** должен иметь значение *Индивидуальный*.

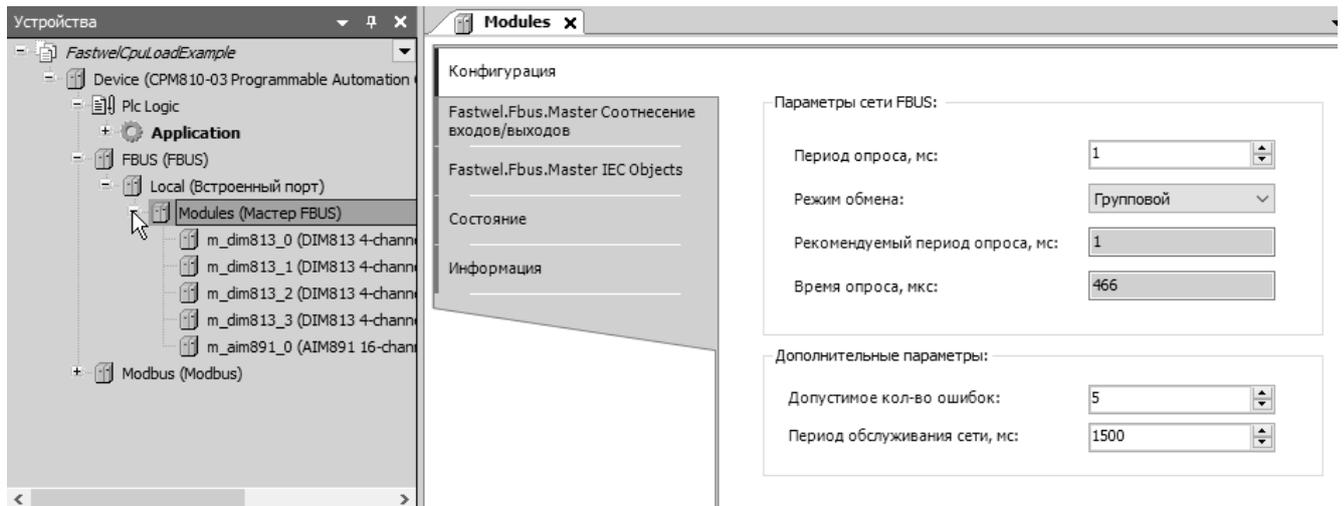


Рисунок 93 – Параметры сервиса ввода-вывода

Период обмена с модулями ввода-вывода определяется параметром **Период опроса, мс**. При использовании режима индивидуального обмена пропускная способность внутренней шины существенно ухудшается за счет появления пауз между запросами к соседним модулям, а также за счет передачи в ответе каждого модуля дополнительных пяти байт (идентификатора мастера шины и поля контрольной суммы). Длительность паузы между обменами для локального порта FBUS отображается на вкладках редактора FBUS – (*Встроенный порт*) и FBUS – (*Удаленный TCP-порт*) в виде параметра **Латентность, мкс**, как показано на рисунке 94.

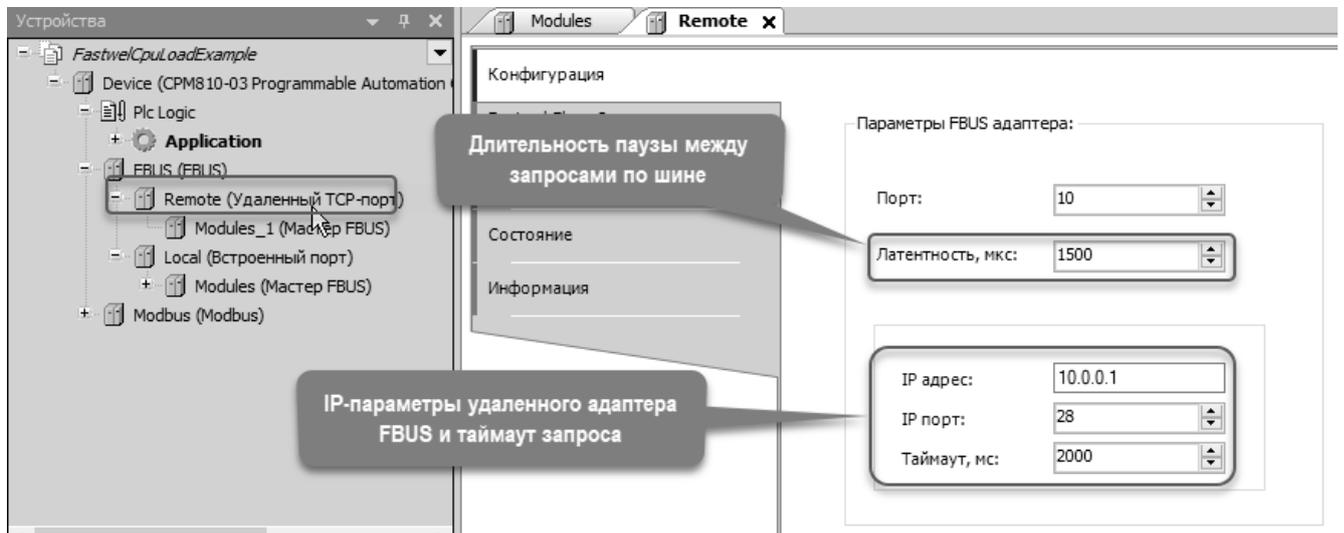


Рисунок 94 – Параметры удаленного порта FBUS

Латентность удаленного порта FBUS может быть скорректирована в зависимости от реальной максимальной задержки между запросами к удаленному адаптеру FBUS NIM745-01, а латентность локального порта FBUS имеет фиксированное значение. Данный параметр предназначен для автоматического вычисления параметров **Рекомендуемый период опроса, мс** и **Время опроса, мкс**, отображаемых на вкладке редактора параметров сервиса, показанных на рисунке 93, в зависимости от количества и состава модулей ввода-вывода в конфигурации контроллера.

Параметр **Допустимое кол-во ошибок** определяет количество ошибок группового обмена подряд, по достижении которого шина переводится в частично неисправное или в неисправное состояние и инициируется периодическое выполнение процедуры поиска неисправных и/или отсутствующих модулей. Период выполнения данной процедуры определяется параметром **Период обслуживания сети, мс** мастера шины.

Для удаленного адаптера FBUS на вкладке редактора параметров порта FBUS (*Удаленный TCP-порт*) должны быть настроены следующие параметры:

IP адрес и IP порт – IP-адрес и IP-порт для доступа к удаленному адаптеру FBUS по сети Ethernet. Значения данных параметров должны соответствовать настройкам NIM745-01, установленным для него при конфигурировании через встроенный веб-сервер. Более подробная информация о конфигурировании NIM745-01 приведена в документе *ФАПИ.421459.700РЭ. Fastwel I/O. Распределённая система ввода–вывода. Руководство по эксплуатации.*

Параметр **Таймаут, мс** предназначен для установки значения интервала времени, в течение которого сервис ввода-вывода контроллера будет ожидать ответ на очередной запрос к модулю ввода-вывода, подключенному к данному удаленному адаптеру FBUS. Обратите внимание, что обмен данными с модулями ввода-вывода, подключенными к разным портам шины FBUS, осуществляется в разных потоках исполнения операционной системы, поэтому таймауты взаимодействия через один порт не влияют на обмен данными через остальные порты FBUS.

Параметр **Порт** содержит номер (10 или 11), при помощи которого производится идентификация удаленного адаптера FBUS и его соотнесение с сетевым интерфейсом контроллера. Более подробная информация о параметре **Порт** приведена в п. 4.5.2.2 настоящего документа.

4.5.2.2. Соотнесение удаленного адаптера шины FBUS с IP-адресом сетевого интерфейса

Параметр **Порт** на вкладке редактора параметров удаленного адаптера FBUS, показанной на рисунке 94, предназначен для установки уникального идентификатора удаленного адаптера в сервисе ввода-вывода.

Сервис ввода-вывода CPM810-03 поддерживает до двух удаленных адаптеров шины FBUS NIM745-01, описания которых могут быть добавлены в конфигурацию приложения. Параметр **Порт** позволяет сервису ввода-вывода отличить один адаптер от другого, а также, при необходимости, определить, через какой сетевой интерфейс контроллера будет осуществляться обмен по сети с тем или иным удаленным адаптером.

Таким образом, если в конфигурации приложения предполагается иметь два набора модулей ввода-вывода, подключенных к удаленным адаптерам FBUS NIM745-01, для соответствующих элементов дерева проекта типа (*Удаленный TCP-порт*) должны быть установлены значения параметров **Порт**: 10 и 11.

Если обмен данными хотя бы с одним из двух удаленных адаптеров предполагается осуществлять через разные сетевые интерфейсы контроллера или только через сетевой интерфейс LAN2, то потребуются соотнести номера портов, заданные параметрами **Параметры FBUS адаптера – Порт**, с сетевыми интерфейсами контроллера при помощи веб-конфигуратора CPM810-03 на странице **Удаленный FBUS**, показанной на рисунке 95.

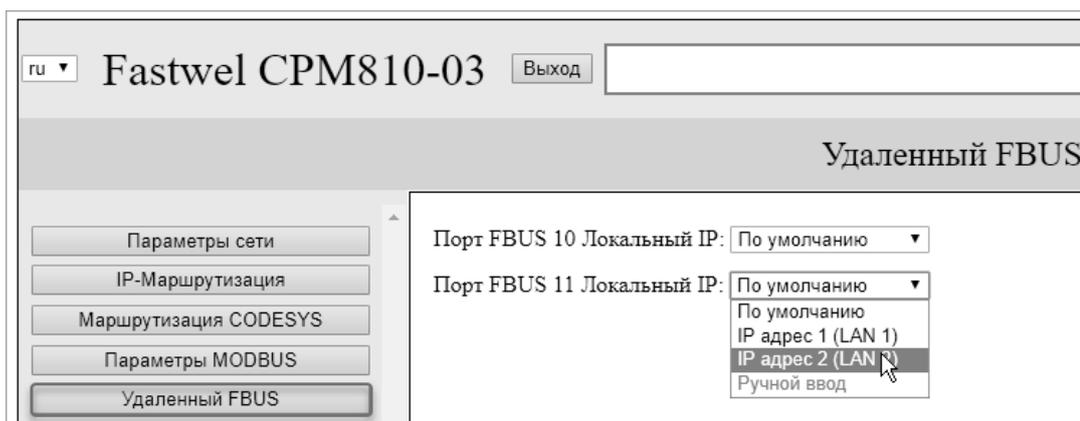


Рисунок 95 – Соотнесение портов удаленных адаптеров FBUS с сетевыми интерфейсами контроллера

Сервис ввода-вывода контроллера с заданным соотнесением номеров портов *По умолчанию* предпринимает попытки установить соединение с удаленными адаптерами FBUS только через сетевой интерфейс LAN1. Если удаленный адаптер FBUS физически подключен к сетевому интерфейсу LAN2, непосредственно или через дополнительное коммутационное оборудование, то для номера порта *n* (где *n* равно 10 или 11) этого адаптера, заданного параметром **Параметры FBUS адаптера – Порт**, в веб-конфигураторе следует установить соотнесение **Порт FBUS *n* Локальный IP: IP адрес 2 (LAN2)**.

При соотношении удаленных адаптеров с сетевыми интерфейсами, IP-адреса которых находятся в разных подсетях, необходимо учитывать, что IP-адрес соотносимого сетевого интерфейса и IP-адрес удаленного адаптера шины FBUS должны принадлежать одной и той же подсети.

После изменения соотношения портов удаленных адаптеров FBUS в веб-конфигураторе нажмите кнопку **Применить конфигурацию**. После перезапуска контроллера новые параметры вступят в силу.

На рисунке 96 приведен пример подключения двух удаленных адаптеров FBUS к сетевым интерфейсам LAN1 и LAN2 контроллера СРМ810-03, настроенным на работу в одной подсети, и соответствующая настройка параметров соотношения номеров портов, назначенных адаптерам в конфигурации приложения, с сетевыми интерфейсами на странице **Удаленный FBUS** веб-конфигуратора СРМ810-03.

Второй пример, в котором два удаленных адаптера FBUS подключены к сетевому интерфейсу LAN2, представлен на рисунке 97.

Более подробная информация о настройке системных параметров контроллера при помощи веб-конфигуратора приведена в разделе 5 настоящего документа.

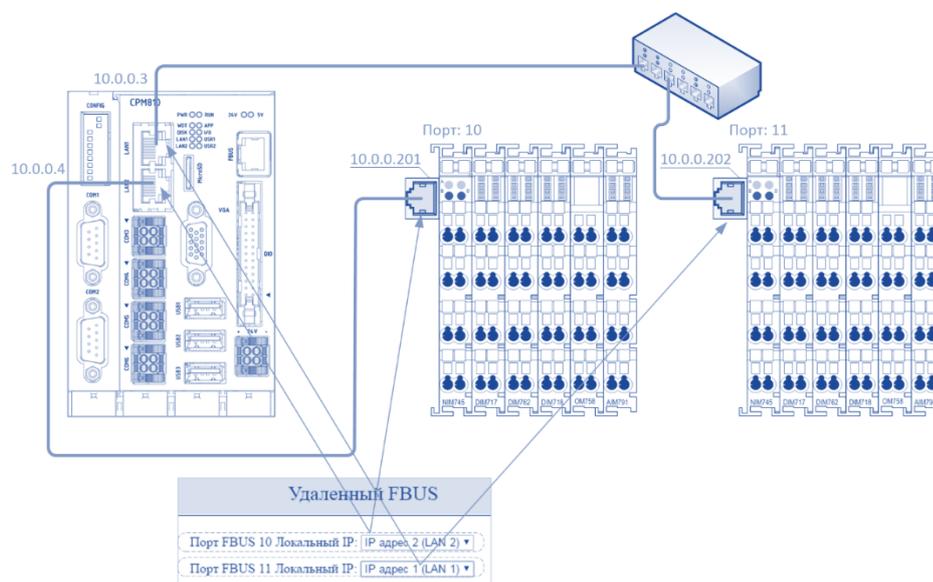


Рисунок 96 – Соотнесение двух удаленных адаптеров FBUS с интерфейсами LAN1 и LAN2

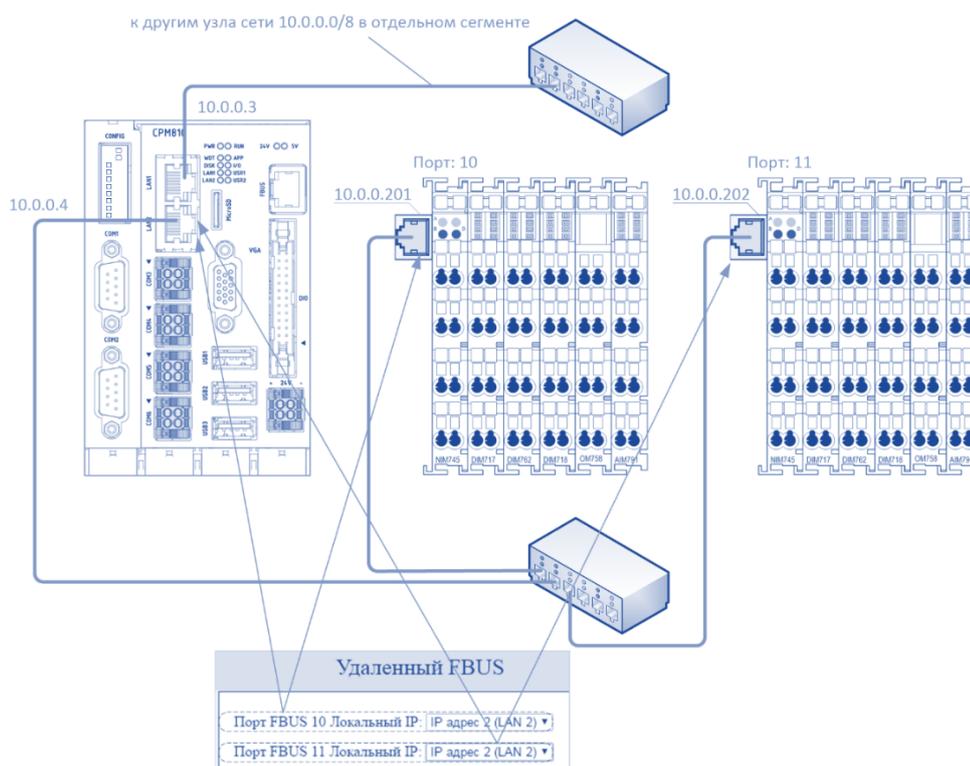


Рисунок 97 – Соотнесение двух удаленных адаптеров FBUS с интерфейсом LAN2

4.5.2.3. Создание конфигурации модулей ввода-вывода

Описания модулей ввода-вывода, которые требуется иметь в составе контроллера на локальной шине, добавляются пользователем к элементу дерева проектной информации *FBUS – Local (Встроенный порт) – Modules (Мастер FBUS)*. Обратите внимание, что имена элементов дерева *Local* и *Modules* могут быть изменены в процессе работы над проектом.

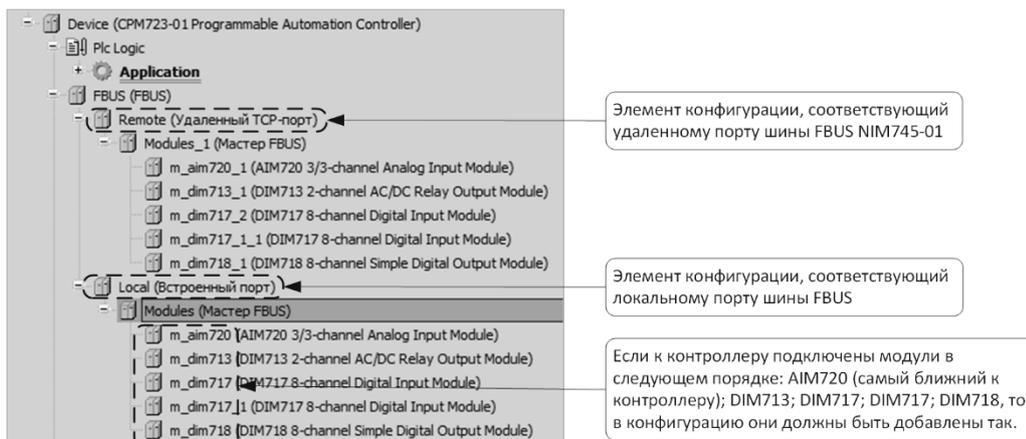


Рисунок 98 – Элементы конфигурации сервиса ввода-вывода FBUS

Для обмена данными с модулями ввода-вывода, подключенными к удаленным адаптерам шины FBUS NIM745-01, в конфигурацию приложения должны быть добавлены описания удаленных адаптеров шины *FBUS – Remote (Удаленный TCP-порт)*, а в их списки *Modules (Мастер FBUS)* должны быть добавлены описания модулей ввода-вывода.

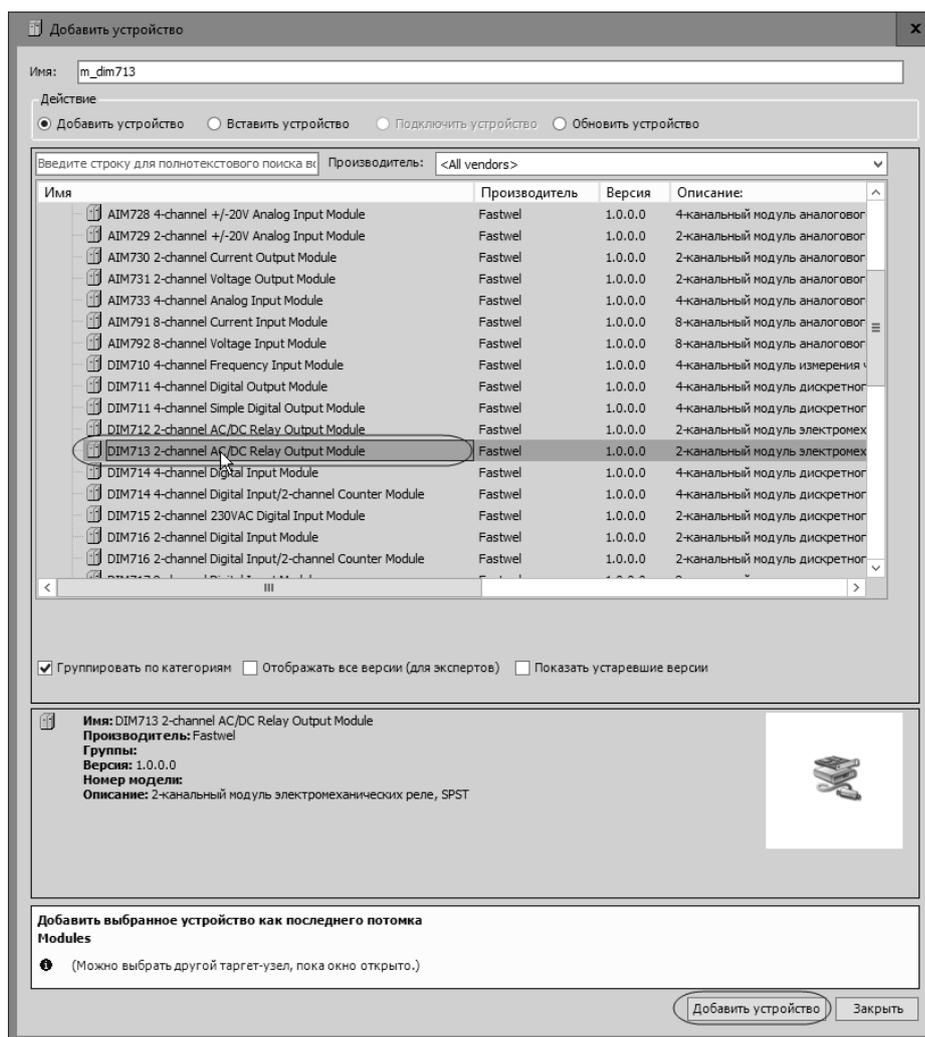


Рисунок 99. Добавление описания модуля ввода-вывода в конфигурацию приложения

Обратите внимание, что описания модулей ввода-вывода должны быть добавлены в список (*Мастер FBUS*) командой контекстного меню **Добавить устройство** или **Вставить устройство** с точным

соблюдением порядка физического подключения модулей к локальной шине контроллера или к удаленному адаптеру NIM745-01. Например, если к контроллеру подключены модули в следующем порядке: AIM720 (самый ближний к контроллеру); DIM713; DIM717; DIM717; DIM718, то конфигурация сервиса ввода-вывода должна выглядеть так, как показано на рисунке 98.

Для добавления описания модуля ввода-вывода в конфигурацию приложения щелкните правой кнопкой мыши над элементом *FBUS (FBUS) – Local (Встроенный порт) – Modules (Мастер FBUS)* или *FBUS (FBUS) – Remote (Удаленный TCP-порт) – Modules (Мастер FBUS)* и выберите команду **Добавить устройство** в контекстном меню. На экране будет отображено окно **Добавить устройство**, показанное на рисунке 99, со списком описаний типов периферийных модулей, которые могут быть вставлены в конфигурацию приложения.

Обратите внимание, что данное окно позволяет как добавлять описание модуля последним элементом списка (**Действие: Добавить устройство**), так и вставлять в любую позицию имеющегося списка модулей (**Действие: Вставить устройство**).

Столбец **Версия** содержит номер версии описательной информации об устройствах, установленной с пакетом адаптации среды разработки, которая используется для представления параметров и каналов ввода-вывода устройств в редакторах устройств IDE МЭК 61131-3. Обратите внимание, что данный номер версии не имеет ничего общего с версией системного программного обеспечения физических устройств.

В списке устройств выберите название типа добавляемого модуля и нажмите кнопку **Добавить устройство**. Описание устройства с именем из поля **Имя** в окне **Добавить устройство** будет добавлено в дерево проекта под элементом *Modules (Мастер FBUS)*.

Обратите внимание, что можно продолжить добавление модулей в проект, не закрывая окно **Добавить устройство**, а выбирая требуемые элементы в списке доступных типов модулей и нажимая **Добавить устройство**.

Закройте окно **Добавить устройство** нажатием кнопки **Заккрыть** и сохраните проект командой **Файл – Сохранить** в главном меню среды разработки.

IDE МЭК 61131-3 поддерживает функцию вставки описаний модулей ввода-вывода в конфигурацию приложения на основе результата сканирования шины работающего контроллера.

Перед добавлением описаний модулей ввода-вывода в конфигурацию приложения путем считывания текущей аппаратной конфигурации выбранной шины контроллера необходимо установить соединение между средой разработки и контроллером, после чего выбрать команду **Поиск устройств** в контекстном меню над элементом проектной информации *Modules (Мастер FBUS)*.

Для добавления в конфигурацию приложения описаний модулей ввода-вывода, физически подключенных к локальному порту FBUS:

1. Включите питание контроллера, если это не было сделано ранее.
2. В IDE МЭК 61131-3 создайте или откройте проект для платформы *CPM810-03 Programmable Automation Controller*.
3. Настройте сетевой путь доступа среды разработки к контроллеру в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.
4. Щелкните правой кнопкой мыши над элементом *FBUS – Local (Встроенный порт) – Modules (Мастер FBUS)* и в появившемся контекстном выполните команду **Поиск устройств**, как показано на рисунке 100.

При наличии связи с контроллером на экран монитора будет выведено окно **Поиск устройств**, показанное на рисунке 101, содержащее описания модулей ввода-вывода, физически подключенных к локальной шине и обнаруженных сервисом ввода-вывода контроллера.

5. Выберите название модуля, подлежащего вставке в конфигурацию приложения, и нажмите кнопку **Копировать в проект**. Окно **Поиск устройств** будет закрыто, а описание модуля появится в дереве проекта.

Для вставки в конфигурацию приложения описаний всех обнаруженных модулей, не выбирая названий каких-либо модулей в окне **Поиск устройств**, нажмите кнопку **Копировать все устройства в проект**. Окно **Поиск устройств** будет закрыто, а описания всех обнаруженных модулей появятся в дереве проекта в списке (*Мастер FBUS*).

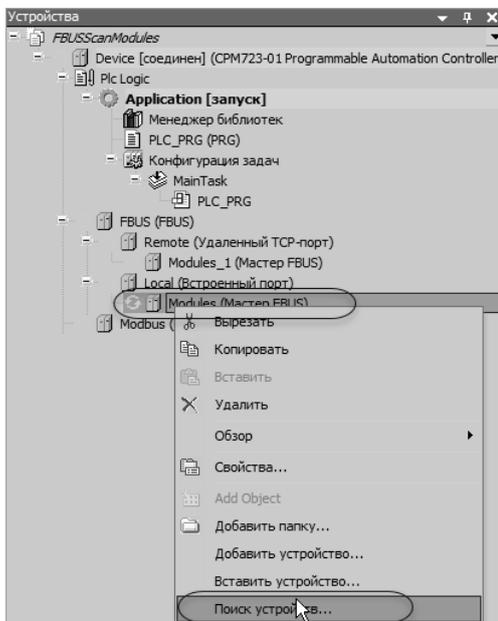


Рисунок 100 – Сканирование локальной шины FBUS

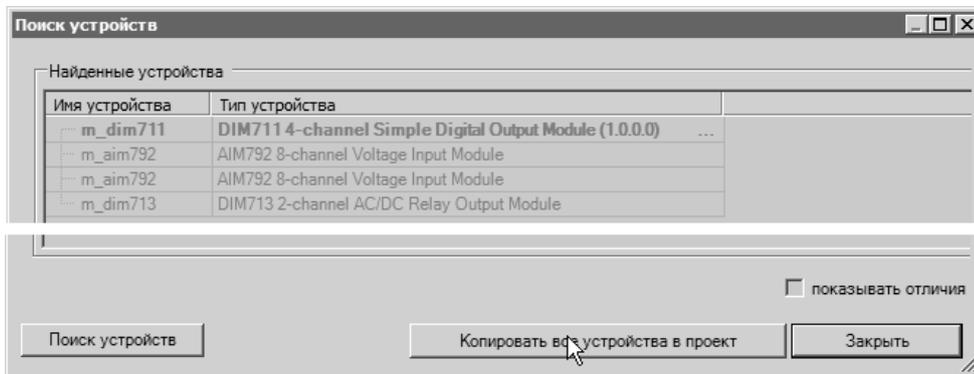


Рисунок 101 – Результат сканирования шины FBUS

6. Для загрузки измененного приложения в контроллер выполните команду **Онлайн – Логин**, после чего нажмите **Да** в диалоговой панели с предложением выполнить полную загрузку приложения в контроллер.
7. После успешной загрузки в среде разработки нажмите кнопку F5 или выполните команду **Отладка – Старт**. Если описания модулей ввода-вывода в загруженном приложении совпадают с фактической аппаратной конфигурацией модулей ввода-вывода, подключенных к локальной шине контроллера, статус исправности шины и модулей будет отображен символом круглой формы зеленого цвета .

Если описания всех или некоторых модулей ввода-вывода в загруженном приложении не совпадают с фактической аппаратной конфигурацией, то статус ошибки порта, шины или отдельных модулей будет отображен символом .

Для добавления в конфигурацию приложения описаний модулей ввода-вывода, физически подключенных к удаленному адаптеру FBUS:

1. Включите питание контроллера и удаленного адаптера FBUS (NIM745-01), если это не было сделано ранее.
2. В IDE МЭК 61131-3 создайте или откройте проект для платформы *CPM810-03 Programmable Automation Controller*.
3. В конфигурацию приложения добавьте описание удаленного порта FBUS, для чего выполните команду **Добавить устройство** в контекстном меню над элементом *FBUS* в дереве проекта, в окне **Добавить устройство** выберите *Удаленный TCP-порт* и нажмите **Добавить устройство**.
4. Откройте вкладку редактора параметров удаленного адаптера FBUS, дважды щелкнув на элементе *FBUS – (Удаленный TCP-порт)* в дереве проекта, или перейдите на уже открытую вкладку редактора, и в поле **Параметры FBUS адаптера – IP-адрес** введите IP-адрес удаленного адаптера FBUS.

5. Сохраните проект командой **Файл – Сохранить** или нажатием сочетания клавиш Ctrl-S.
6. Если это не было сделано ранее, настройте сетевой путь доступа среды разработки к контроллеру в соответствии с указаниями руководства пользователя Fastwel PLC Application Toolkit.
7. Выполните команду **Онлайн – Логин**. Если в контроллере отсутствует выполняющееся приложение или если приложение в контроллере отличается от открытого в среде разработки, на экран будет выведена диалоговая панель с сообщением о необходимости загрузки измененного приложения в контроллер или обновления приложения в контроллере. Нажмите **Да**, и приложение будет загружено в контроллер, после чего система исполнения приложений в контроллере будет ожидать команды запуска приложения **Отладка – Старт** или нажатием клавиши F5.
8. Если параметры удаленного адаптера FBUS настроены правильно, и адаптер доступен в сети, к которой подключен контроллер, статус связи с адаптером будет отображен символом  справа от элемента (*Удаленный TCP-порт*) в дереве проекта.
9. Щелкните правой кнопкой мыши над элементом *FBUS – Remote (Удаленный TCP-порт) – Modules_x (Мастер FBUS)* и в появившемся контекстном выполните команду **Поиск устройств**.
10. При наличии связи с контроллером на экран монитора будет выведено окно **Поиск устройств**, показанное на рисунке 101, содержащее описание модулей ввода-вывода, физически подключенных к шине удаленного адаптера FBUS и обнаруженных сервисом ввода-вывода контроллера.
11. Выберите название модуля, подлежащего вставке в конфигурацию приложения, и нажмите кнопку **Копировать в проект**. Окно **Поиск устройств** будет закрыто, а описание модуля появится в дереве проекта.
Для вставки в конфигурацию приложения описаний всех обнаруженных модулей, не выбирая названий каких-либо модулей в окне **Поиск устройств**, нажмите кнопку **Копировать все устройства в проект**. Окно **Поиск устройств** будет закрыто, а описания всех обнаруженных модулей появятся в дереве проекта в списке (*Мастер FBUS*).
12. Для загрузки измененного приложения в контроллер выполните команду **Онлайн – Отключение**, а затем **Онлайн – Логин**, после чего нажмите **Да** в диалоговой панели с предложением выполнить полную загрузку приложения в контроллер.
13. После успешной загрузки в среде разработки нажмите кнопку F5 или выполните команду **Отладка – Старт**. Если описания модулей ввода-вывода в загруженном приложении совпадают с фактической аппаратной конфигурацией модулей ввода-вывода, подключенных к удаленному адаптеру шины FBUS, статус исправности шины и модулей будет отображен символом круглой формы зеленого цвета .
14. Если описания всех или некоторых модулей ввода-вывода в загруженном приложении не совпадают с фактической аппаратной конфигурацией, то статус ошибки порта, шины или отдельных модулей будет отображен символом .

Обратите внимание, что имеется возможность за один раз выполнить сканирование только одной шины для возможного добавления обнаруженных модулей в проект. После вставки найденных модулей в проект требуется загрузить приложение в контроллер, после чего появится возможность сканирования другой шины.

Большинство описаний модулей ввода-вывода, добавляемых в список (*Мастер FBUS*) дерева проекта, имеют собственные параметры конфигурации, которые могут быть настроены в соответствующих редакторах, а также входные и выходные каналы в образе процесса. Информация о настройке параметров отдельных типов модулей, структуре, составе и назначении их входных и выходных каналов в образе процесса приведена в документе *ИМЕС.00300-03 33 01. Модули ввода-вывода. Руководство программиста*.

При загрузке приложения в контроллер методом полной загрузки системе исполнения передается бинарная конфигурация сервиса ввода-вывода, которая затем используется во время работы сервиса для обмена данными с модулями и для обработки внештатных ситуаций. При изменении конфигурации сервиса ввода-вывода в приложении обновление ранее загруженного приложения в контроллере возможно только методом полной загрузки.

4.5.3. Диагностика

4.5.3.1. Индикация

Индикация функционирования сервиса ввода-вывода осуществляется при помощи светодиодного индикатора I/O следующим образом:

Зеленый цвет (непрерывно) – все модули ввода-вывода, определенные в конфигурации приложения для всех портов шины FBUS, обнаружены и успешно сконфигурированы. Обмен с модулями ввода-вывода, подключенными к контроллеру, успевает завершиться за время, равное значению параметра **Период опроса, мс**.

Зеленый цвет (прерывисто) – все модули ввода-вывода, определенные в конфигурации приложения, обнаружены и успешно сконфигурированы, но обмен данными реального времени с модулями хотя бы через один из портов FBUS, локальный или удаленный, не может быть выполнен за время, заданное параметром **Период опроса, мс** для соответствующих элементов (*Мастер FBUS*), в связи с чем используется расчетное минимально достижимое время обмена данными со всеми модулями в выбранном режиме, отображаемое параметром **Рекомендуемый период опроса, мс**.

Зеленый и красный цвета (попеременно прерывисто) – хотя бы одна из шин FBUS, определенных в конфигурации приложения, находится в *исправном* состоянии, а остальные – либо в *неисправном*, либо в *частично исправном* состоянии. При частично исправном состоянии шины состав и конфигурация обнаруженных модулей ввода-вывода соответствуют заданным в конфигурации приложения, загруженного в контроллер, за исключением некоторых модулей, которых нет среди обнаруженных.

Красный цвет:

кратковременно, в момент первоначального или повторного конфигурирования модулей после загрузки нового приложения или при запуске ранее загруженного приложения;

во время функционирования приложения – конфигурация модулей ввода-вывода, определенная в проекте, не совпадает с аппаратной конфигурацией модулей, подключенных ко всем шинам FBUS контроллера, либо, при отсутствии модулей в конфигурации локальной шины, отсутствует связь со всеми удаленными адаптерами FBUS.

Отсутствие свечения – в конфигурации загруженного приложения отсутствуют описания модулей ввода-вывода.

4.5.3.2. Диагностические каналы сервиса ввода-вывода

Диагностические каналы сервиса ввода-вывода предназначены для реализации контроля связи с удаленными адаптерами межмодульной шины, а также для проверки наличия связи и обмена данными с модулями ввода-вывода в исполняемом коде приложения, загруженного в контроллер.

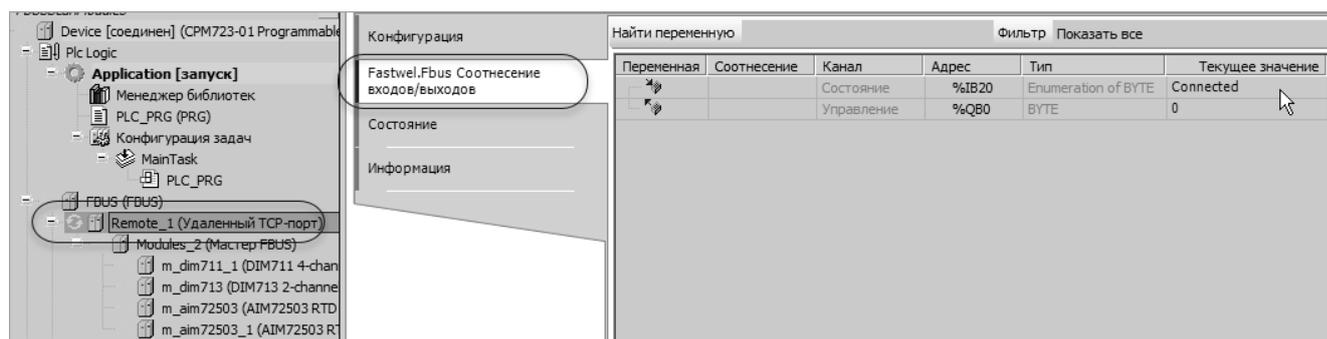


Рисунок 102 – Каналы выбранного порта шины FBUS в образе процесса

Локальный адаптер и удаленный адаптер шины FBUS, представленные в конфигурации приложения элементами *Встроенный порт* и *Удаленный TCP-порт* соответственно, имеют входной канал *Состояние*, показанный на рисунке 102. Данный канал может принимать значение 0 (*Disconnected*), если при работе системы исполнения отсутствует связь с портом, и 1 (*Connected*) – если связь с портом установлена.

Выходной канал *Управление* порта FBUS не используется в текущей версии системы исполнения.

Адреса каналов *Состояние* и *Управление* во входном и выходном образе процесса могут быть изменены пользователем для фиксации начальных адресов областей памяти, занимаемых конфигурацией шины FBUS в образе процесса. Это позволяет избежать сдвига адресов входных и

выходных каналов при вставке или удалении других элементов конфигурации приложения с входными и выходными каналами, расположенными выше данного порта FBUS в образе процесса.

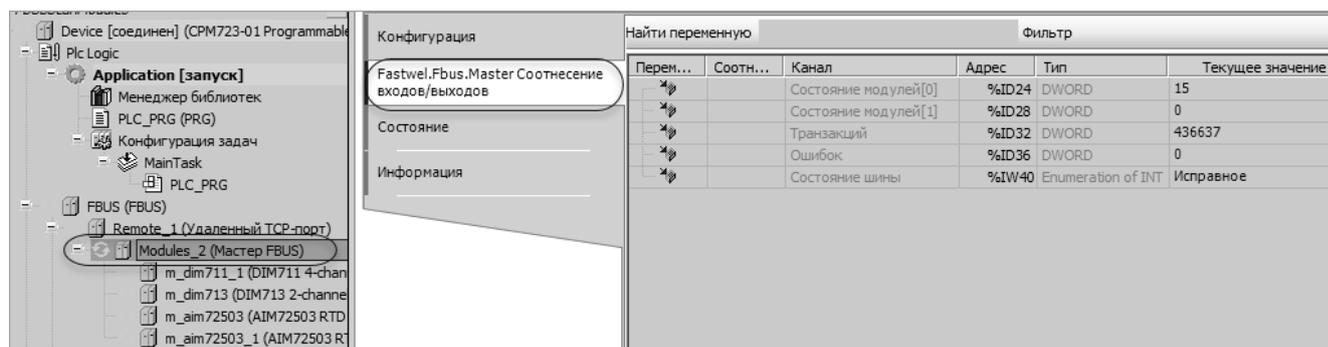


Рисунок 103 – Каналы выбранного мастера шины FBUS

Элемент конфигурации (*Мастер FBUS*), к которому добавляются описания модулей ввода-вывода, имеет набор входных диагностических каналов, показанный на рисунке 103. Данный набор каналов в библиотеке *FastwelFbusIO* представлен типом данных *FBUS_FIAGNOSTICS*:

```

TYPE FBUS_DIAGNOSTICS :
STRUCT
  // Битовая маска состояния 64-х модулей
  dwIoStatus : ARRAY [1..2] OF DWORD;
  // Количество транзакций ввода-вывода
  dwTransactionsCount : DWORD;
  // Количество ошибок ввода-вывода
  dwErrorsCount : DWORD;
  // Состояние шины:
  // 0 -- неопределенное
  // 1 -- исправное
  // 2 -- частично исправное
  // 3 -- неисправное
  iBusState : INT;
END_STRUCT
END_TYPE

```

Канал *dwIoStatus* (*Состояние модулей[0]* и *Состояние модулей[1]*) во время работы контроллера содержит битовую маску состояния 64-х модулей ввода-вывода, подключенных к шине. Логическая единица в некотором бите данного канала свидетельствует о том, что модуль ввода-вывода, номер которого совпадает с номером бита (начиная с 0), описание которого имеется в конфигурации приложения, обнаружен и сконфигурирован.

Канал *dwTransactionsCount* (*Транзакций*) содержит количество операций группового обмена данными с набором модулей ввода-вывода, определенным в конфигурации приложения для данной шины FBUS.

Канал *dwErrorsCount* (*Ошибок*) содержит количество ошибок группового обмена с набором модулей ввода-вывода, определенным в конфигурации приложения для данной шины FBUS.

Обратите внимание, что при работе с модулями NIM741 и NIM742, представленными в конфигурации приложения элементами *NIM741 RS-485 1xUART Stream Module* и *NIM742 RS-232 1xUART Stream Module*, значения на каналах *dwTransactionsCount* и *dwErrorsCount* не изменяются. Это связано с тем, что для реализации последовательных портов на основе данных модулей используются иные механизмы обмена данными. Более подробная информация об элементах *NIM741 RS-485 1xUART Stream Module* и *NIM742 RS-232 1xUART Stream Module* приведена в п. 4.5.4 настоящего документа.

Канал *iBusState* (*Состояние шины*) содержит код текущего состояния данной шины FBUS: 0 – неопределенное; 1 – исправное; 2 – частично исправное; 3 – неисправное.

Например, если в конфигурацию шины FBUS, обслуживаемую через некоторый, локальный или удаленный, адаптер, добавлены описания 34-х модулей, то в случае нахождения шины в полностью исправном состоянии *iBusState* = 1, *dwIoStatus[0]* = 16#FFFF_FFFF, а *dwIoStatus[1]* = 16#0000_0003. Если в процессе работы контроллера утрачена связь с модулями 12 и 32, начиная с 0, то *iBusState* = 2, *dwIoStatus[0]* = 16#FFFF_7FFF, а *dwIoStatus[1]* = 16#0000_0002.

Значения диагностических каналов шины FBUS обновляются с периодом около 1 с.

В библиотеке FastwelFbusIO имеется функциональный блок *FBUS_CHECKER*, обеспечивающий возможность контроля состояния шины FBUS в приложении при соотнесении с диагностическими каналами соответствующих порта и шины FBUS.

4.5.4. Порты последовательной передачи данных на базе модулей NIM741/2, NIM841/2

4.5.4.1. Общие сведения

Система исполнения контроллера CPM810-03 обеспечивает возможность создания в конфигурации контроллера до 32 портов асинхронного последовательного интерфейса RS-485 и RS-232C на базе модулей NIM741, NIM742, NIM841 и NIM842, подключенных к локальной межмодульной шине FBUS.

Описания устройств *NIM741 RS-485 1xUART Stream Module* и *NIM742 RS-232C 1xUART Stream Module* представляют реализацию программной модели модулей NIM741 и NIM742 в виде дополнительного коммуникационного порта последовательного интерфейса (COMn) в системе. В таком представлении программное управление модулями NIM741 и NIM742 осуществляется не через входные и выходные каналы в образе процесса, а посредством функций системной библиотеки SysCom, обеспечивающей доступ к последовательным портам. При этом модули NIM741 и NIM742 не включаются в расписание периодического группового обмена сервиса FBUS, а транзакции ввода-вывода данных выполняются асинхронно относительно вызовов функций библиотеки SysCom.

Кроме того, последовательные порты на основе описаний устройств *NIM741 RS-485 1xUART Stream Module* и *NIM742 RS-232C 1xUART Stream Module* могут использоваться сервисами мастера и подчиненного узла протокола MODBUS RTU/ASCII в качестве портов обмена с другими узлами сети MODBUS RTU/ASCII.

4.5.4.2. Идентификация последовательных портов на базе NIM741/841, NIM742/842

Идентификатор COM-порта на базе модулей NIM741/841 (элементы конфигурации *NIM741/8/841 RS-485 1xUART Stream Module*) и NIM742/842 (элементы конфигурации *NIM742/842 RS-232C 1xUART Stream Module*), подключенных к локальному порту FBUS (элемент конфигурации *Встроенный порт*), передаваемый функции SysComOpen или SysComOpen2 библиотеки SysCom для получения системного идентификатора устройства, формируется по правилу:

100+n

где n – позиция модуля, начиная с 1, на локальной шине контроллера. При этом из нумерации должны быть исключены все вспомогательные модули, не участвующие в обмене по шине FBUS: OM743, OM752, OM753, OM754, OM755, OM756, OM856, OM757, OM857, OM758, OM759, OM796, и модуль оконечный OM750. Таким образом, n – это позиция модуля в списке (*Мастер FBUS*) локального порта (*Встроенный порт*), начиная с 1.

Пример

Пусть к локальному порту шины FBUS контроллера подключены следующие модули ввода-вывода и вспомогательные модули: OM756, AIM721, AIM730, OM751, DIM717, DIM717, NIM741, OM752, AIM722, AIM722, AIM72503, NIM741, NIM742, NIM742, OM750, и приложению требуется получить программный доступ ко всем последовательным портам на основе модулей NIM741 и NIM742 с использованием системной библиотеки SysCom.

Тогда в конфигурацию сервиса ввода-вывода приложения должны быть добавлены следующие элементы:

1. *AIM721 4-channel 0-20mA Analog Input Module*
2. *AIM730 2-channel Current Output Module*
3. *OM751 24VDC Power Supply Module*
4. *DIM717 8-channel Digital Input Module*
5. *DIM717 8-channel Digital Input Module*
6. *NIM741 RS-485 1xUART Stream Module*
7. *AIM722 2-channel 0-20mA Analog Input Module*
8. *AIM722 2-channel 0-20mA Analog Input Module*
9. *AIM72503 RTD Input Module (GOST 6651-2009)*
10. *NIM741 RS-485 1xUART Stream Module*

11. NIM742 RS-232C 1xUART Stream Module

12. NIM742 RS-232C 1xUART Stream Module

В конфигурации присутствуют по два элемента *NIM741 RS-485 1xUART Stream Module* и *NIM742 RS-232C 1xUART Stream Module*, обеспечивая возможность организации двух портов интерфейса RS-485 и двух портов интерфейса RS-232C соответственно. Данные элементы имеют порядковые номера 6, 10, 11, 12. Согласно приведенному выше правилу формирования идентификаторов портов, функции SysComOpen или SysComOpen2 (через структуру COM_Settings) должны быть переданы следующие идентификаторы:

106 (первый NIM741)

110 (второй NIM741)

111 (первый NIM742)

112 (второй NIM742)

Обратите внимание, что идентификатор порт с префиксом COM может быть получен на вкладке редактора устройств типа *NIM741 RS-485 1xUART Stream Module*, *NIM742 RS-232C 1xUART Stream Module*, *NIM841 RS-485 1xUART Stream Module*, *NIM842 RS-232C 1xUART Stream Module* как показано на рисунке 104.

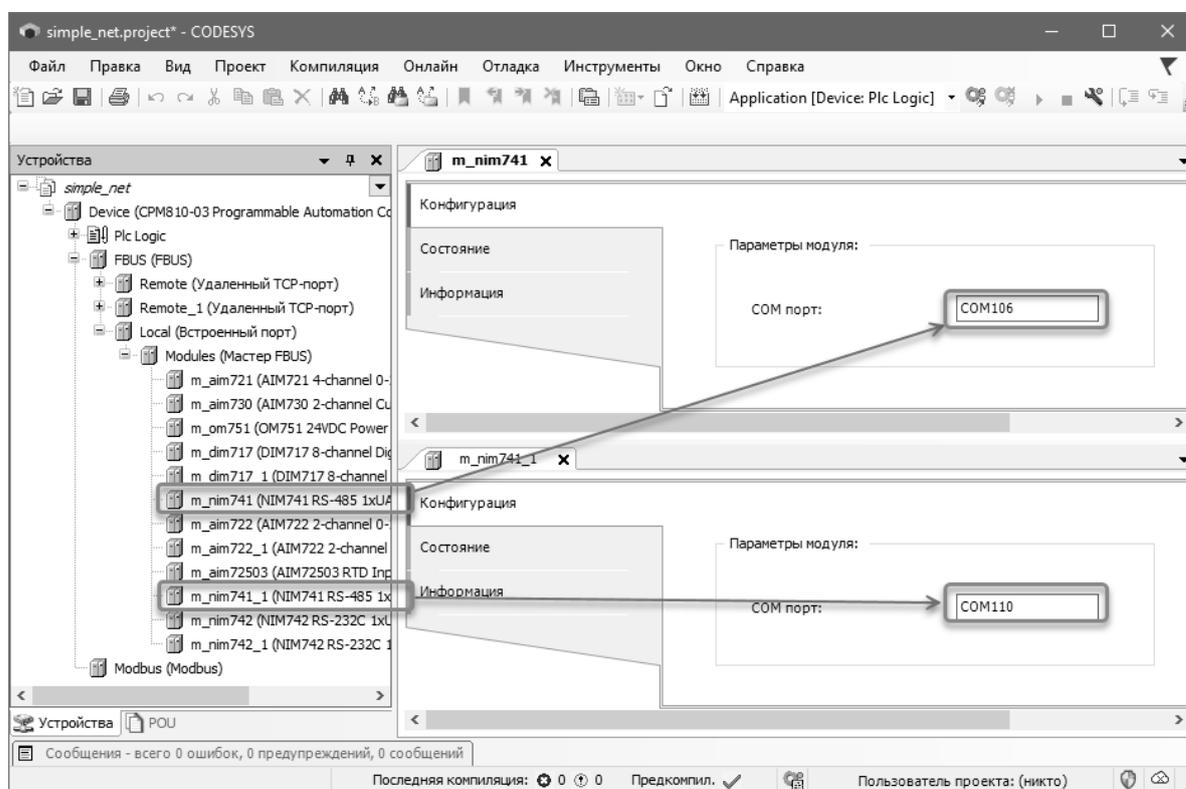


Рисунок 104 – Получение идентификатора последовательного порта на основе NIM741 или NIM742

4.5.4.3. Доступ к последовательным портам из приложения

Обмен данными через последовательные порты на базе NIM741, NIM742, NIM841, NIM842 в приложении может быть реализован при помощи функций системной библиотеки SysCom, а также с использованием встроенных сервисов протокола MODBUS RTU/ASCII.

Описание функций и типов данных библиотеки SysCom находится во встроенной системе документирования библиотеки и отображается на специальной вкладке **Документация** в **Менеджере библиотек** среды разработки CODESYS V3, как показано на рисунке 105. Информация об особенностях применения библиотеки приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Примеры применения библиотеки SysCom содержатся в проектах SerialConsole.project и ModbusRtuClient.project, поставляемых в составе Fastwel PLC Application Toolkit.

Для использования встроенных сервисов мастера и подчиненного узла протокола MODBUS для обмена данными через последовательные порты на базе модулей NIM741, NIM742, NIM841, NIM842

достаточно ввести числовой идентификатор порта, полученный согласно указаниям п. 4.5.4.2, в поле **COM порт** на вкладке редактора параметров сервиса протокола, как показано на рисунке 106.

Более подробная информация о применении встроенных сервисов протокола MODBUS TCP и MODBUS RTU/ASCII приведена в п. 4.6.3 и п. 4.6.4 настоящего документа, а также в документе *ИМЕС.00300-03 33 03. Протокол MODBUS. Руководство по конфигурированию и программированию*.

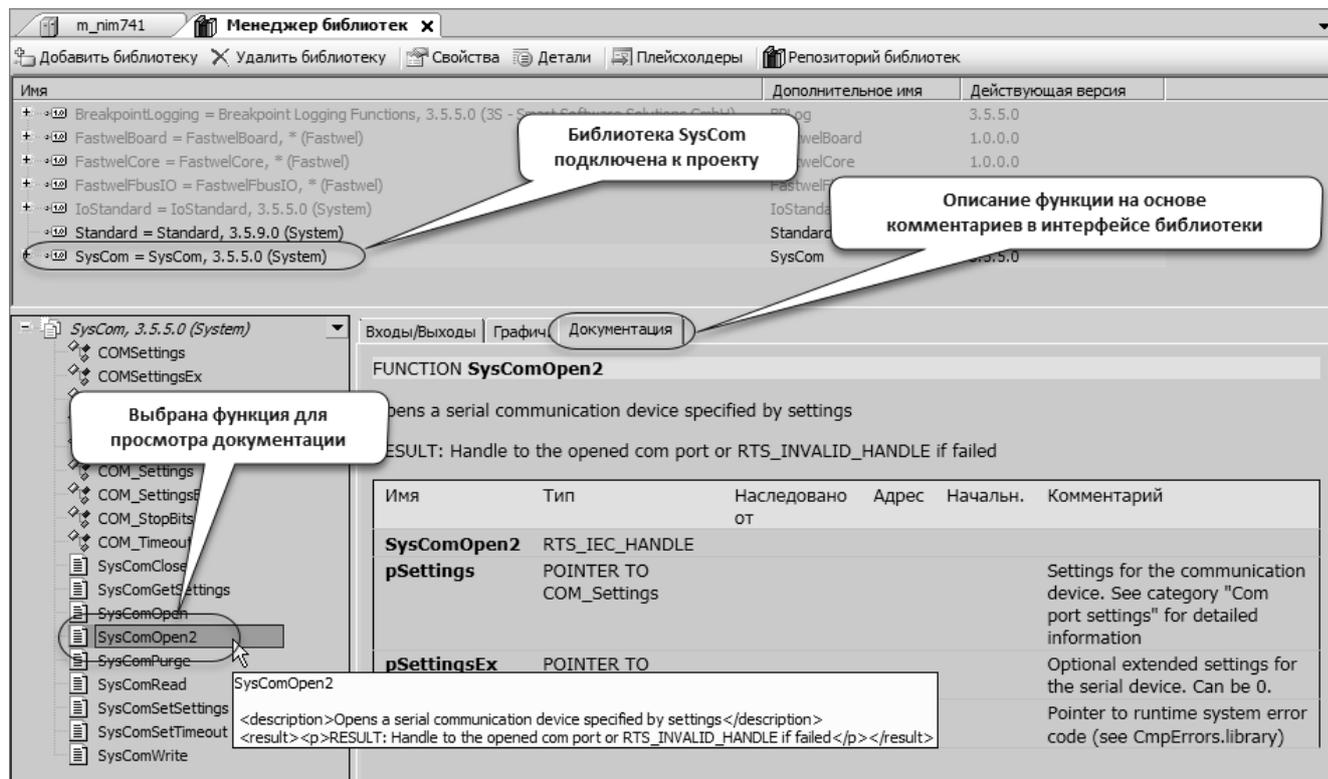


Рисунок 105 – Доступ к встроенной документации библиотеки SysCom

Параметры сети MODBUS SERIAL

Гранулярность опроса, мс:	50
COM порт:	107
Тип протокола:	RTU
Скорость обмена:	38400
Количество бит данных:	8
Четность:	EVEN
Количество стоп-битов:	1

Рисунок 106 – Ввод идентификатора последовательного порта для использования встроенным мастером протокола MODBUS

4.5.5. Получение информации о подключенных модулях ввода-вывода

Для получения информации о модулях ввода-вывода, подключенных к локальному или удаленному порту шины FBUS контроллера можно воспользоваться командой *fiolist* оболочки ПЛК (**PLC Shell**):

1. Настройте параметры сервиса IDE Gateway и установите с контроллером связь через сервисный USB-порт в соответствии с указаниям руководства пользователя Fastwel PLC Application Toolkit.
2. Для получения перечня модулей ввода-вывода, подключенных к локальному порту контроллера в редакторе устройства выберите вкладку **Оболочка ПЛК (PLC Shell)** и в поле ввода команд введите:

fiolist

в области ответного сообщения на команду будет отображен список обнаруженных модулей ввода-вывода, подключенных к локальному порту шины FBUS:

```

FBUS0 [port 0] modules detected: 16!
 1: DIM713 SerN(713.1726 2011/08) Firmware version(2.10) Code(2016547633)
 2: DIM713 SerN(713.0130 2006/05) Firmware version(2.10) Code(2016547633)
 3: DIM713 SerN(713.2721 2014/01) Firmware version(2.10) Code(2016547633)
 4: DIM713 SerN(713.2771 2014/01) Firmware version(2.10) Code(2016547633)
 5: DIM761 SerN(761.0064 2011/05) Firmware version(2.2) Code(2016547786)
 6: DIM760 SerN(760.0018 2007/11) Firmware version(2.2) Code(2016547785)
 7: DIM760 SerN(760.0020 2007/11) Firmware version(2.2) Code(2016547785)
 8: DIM760 SerN(760.0045 2007/11) Firmware version(2.2) Code(2016547785)
 9: DIM760 SerN(760.0259 2011/08) Firmware version(2.2) Code(2016547785)
10: DIM760 SerN(760.0004 2007/07) Firmware version(2.2) Code(2016547785)
11: DIM760 SerN(760.0002 2007/07) Firmware version(2.2) Code(2016547785)
12: AIM726 SerN(726.0036 2011/10) Firmware version(2.12) Code(1930660214)
13: AIM726 SerN(726.0015 2006/05) Firmware version(2.12) Code(1930660214)
14: AIM726 SerN(726.0028 2011/11) Firmware version(2.12) Code(1930660214)
15: AIM727 SerN(727.0113 2012/09) Firmware version(2.12) Code(1930660215)
16: AIM727 SerN(727.0007 2006/05) Firmware version(2.12) Code(1930660215)

```

Строка *FBUS0 [port 0] modules detected:* содержит количество обнаруженных модулей ввода-вывода, после которой выводятся строки, описывающие обнаруженные модули, следующего формата:

Номер: Тип SerN(серийный номер год/месяц) Firmware version(версия микропрограммы)

3. Для получения информации о модулях ввода-вывода, подключенных к удаленному порту шины FBUS, в контроллер предварительно должно быть загружено приложение, в конфигурации которого имеется описание, как минимум, одного удаленного порта шины FBUS в виде элементов (*Удаленный TCP-порт*). В таком случае команде оболочки *fiolist* следует передать номер (начиная с 1) элемента (*Удаленный TCP-порт*), для которого требуется запросить список подключенных модулей ввода-вывода, например:

```
fiolist 1
```

```

FBUS1 [port 11] modules detected: 8!
 1: DIM711 SerN(711.2723 2015/07) Firmware version(2.12) Code(2016547631)
 2: DIM713 SerN(713.0012 2006/04) Firmware version(2.8) Code(2016547633)
 3: AIM72503 SerN(725.0628 2016/03) Firmware version(2.6) Code(61405638)
 4: AIM72503 SerN(725.0597 2016/03) Firmware version(2.6) Code(61405638)
 5: DIM711 SerN(711.2901 2015/07) Firmware version(2.12) Code(2016547631)
 6: AIM792 SerN(792.0079 2015/06) Firmware version(1.3) Code(1930660427)
 7: AIM792 SerN(792.0078 2015/06) Firmware version(1.3) Code(1930660427)
 8: DIM713 SerN(713.1932 2013/01) Firmware version(2.10) Code(2016547633)

```

В данном случае для первого элемента (*Удаленный TCP-порт*) в дереве проекта установлен номер порта 11, и на его шине FBUS обнаружено 8 подключенных модулей.

Если в контроллер загружено приложение с описанием двух удаленных портов шины FBUS, то для получения информации об обнаруженных модулях ввода-вывода, подключенных ко второму удаленному порту, следует ввести команду оболочки ПЛК:

```
fiolist 2
```

```

FBUS2 [port 10] modules detected: 10!
 1: DIM713 SerN(713.1938 2013/01) Firmware version(2.10) Code(2016547633)
 2: DIM713 SerN(713.2718 2014/01) Firmware version(2.10) Code(2016547633)
 3: DIM712 SerN(712.2209 2015/06) Firmware version(2.4) Code(2016547632)
 4: DIM712 SerN(712.0896 2013/01) Firmware version(2.4) Code(2016547632)
 5: DIM712 SerN(712.2257 2015/06) Firmware version(2.4) Code(2016547632)
 6: AIM728 SerN(728.0008 2011/04) Firmware version(2.12) Code(1930660216)
 7: AIM728 SerN(728.0128 2006/07) Firmware version(2.12) Code(1930660216)
 8: AIM728 SerN(728.0154 2006/07) Firmware version(2.12) Code(1930660216)
 9: AIM728 SerN(728.0012 2006/04) Firmware version(2.12) Code(1930660216)
10: AIM729 SerN(729.0110 2006/12) Firmware version(2.12) Code(1930660217)

```

В данном случае для второго элемента (*Удаленный TCP-порт*) в дереве проекта установлен номер порта 10, и на его шине FBUS обнаружено 10 подключенных модулей.

Пример последовательного применения команды *fiolist* для трех разных портов шины FBUS представлен на рисунке 107.

Device x

Установки соединения

Приложения

Файлы

Журнал

Установки ПЛК

Оболочка ПЛК

Пользователи и группы

Fastwel.Root Конфигурация

Fastwel.Root Соотношение входов/выходов

Размещение задачи

Состояние

Информация

```

fiolist
FBUS0 [port 0] modules detected: 16!
1: DIM713 SerN(713.1726 2011/08) Firmware version(2.10) Code(2016547633)
2: DIM713 SerN(713.0130 2006/05) Firmware version(2.10) Code(2016547633)
3: DIM713 SerN(713.2721 2014/01) Firmware version(2.10) Code(2016547633)
4: DIM713 SerN(713.2771 2014/01) Firmware version(2.10) Code(2016547633)
5: DIM761 SerN(761.0064 2011/05) Firmware version(2.2) Code(2016547786)
6: DIM760 SerN(760.0018 2007/11) Firmware version(2.2) Code(2016547785)
7: DIM760 SerN(760.0020 2007/11) Firmware version(2.2) Code(2016547785)
8: DIM760 SerN(760.0045 2007/11) Firmware version(2.2) Code(2016547785)
9: DIM760 SerN(760.0259 2011/08) Firmware version(2.2) Code(2016547785)
10: DIM760 SerN(760.0004 2007/07) Firmware version(2.2) Code(2016547785)
11: DIM760 SerN(760.0002 2007/07) Firmware version(2.2) Code(2016547785)
12: AIM726 SerN(726.0036 2011/10) Firmware version(2.12) Code(1930660214)
13: AIM726 SerN(726.0015 2006/05) Firmware version(2.12) Code(1930660214)
14: AIM726 SerN(726.0028 2011/11) Firmware version(2.12) Code(1930660214)
15: AIM727 SerN(727.0113 2012/09) Firmware version(2.12) Code(1930660215)
16: AIM727 SerN(727.0007 2006/05) Firmware version(2.12) Code(1930660215)

----
fiolist 1
FBUS1 [port 11] modules detected: 8!
1: DIM711 SerN(711.2723 2015/07) Firmware version(2.12) Code(2016547631)
2: DIM713 SerN(713.0012 2006/04) Firmware version(2.8) Code(2016547633)
3: AIM72503 SerN(725.0628 2016/03) Firmware version(2.6) Code(61405638)
4: AIM72503 SerN(725.0597 2016/03) Firmware version(2.6) Code(61405638)
5: DIM711 SerN(711.2901 2015/07) Firmware version(2.12) Code(2016547631)
6: AIM792 SerN(792.0079 2015/06) Firmware version(1.3) Code(1930660427)
7: AIM792 SerN(792.0078 2015/06) Firmware version(1.3) Code(1930660427)
8: DIM713 SerN(713.1932 2013/01) Firmware version(2.10) Code(2016547633)

----
fiolist 2
FBUS2 [port 10] modules detected: 10!
1: DIM713 SerN(713.1938 2013/01) Firmware version(2.10) Code(2016547633)
2: DIM713 SerN(713.2718 2014/01) Firmware version(2.10) Code(2016547633)
3: DIM712 SerN(712.2209 2015/06) Firmware version(2.4) Code(2016547632)
4: DIM712 SerN(712.0896 2013/01) Firmware version(2.4) Code(2016547632)
5: DIM712 SerN(712.2257 2015/06) Firmware version(2.4) Code(2016547632)
6: AIM728 SerN(728.0008 2011/04) Firmware version(2.12) Code(1930660216)
7: AIM728 SerN(728.0128 2006/07) Firmware version(2.12) Code(1930660216)
8: AIM728 SerN(728.0154 2006/07) Firmware version(2.12) Code(1930660216)
9: AIM728 SerN(728.0012 2006/04) Firmware version(2.12) Code(1930660216)
10: AIM729 SerN(729.0110 2006/12) Firmware version(2.12) Code(1930660217)

```

fiolist 2

Рисунок 107 – Получение информации о модулях ввода-вывода на трех шинах FBUS контроллера

4.6. Сетевые сервисы

4.6.1. Общие сведения о веб-конфигураторе контроллера

Настройка параметров системных сервисов контроллера выполняется при помощи веб-браузера, запускаемого на ПК. Информация о совместимых браузерах приведена в 5.1. Более подробная информация о настройке системных параметров контроллера при помощи веб-конфигуратора приведена в разделе 5.

Для входа на веб-сервер контроллера в адресной строке браузера следует ввести IP-адрес сетевого интерфейса, подключенного к той же подсети, что и ПК. Например:

http://10.0.0.100

По умолчанию веб-сервер контроллера устанавливается на IP-порт 80 всех доступных сетевых интерфейсов, поэтому в адресной строке не требуется указывать номер IP-порта.

Рекомендации по проверке доступности контроллера по сети с ПК приведены в руководстве пользователя Fastwel PLC Application Toolkit.

При успешном соединении браузера с веб-сервером в окне браузера будет отображена начальная страница веб-сервера, показанная на рисунке 108, с предложением ввести имя пользователя и пароль.

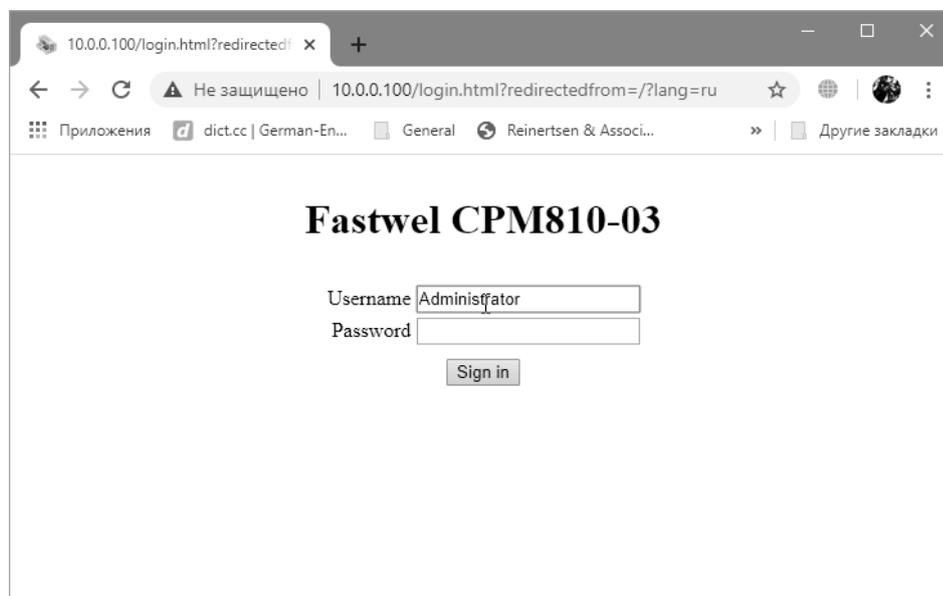


Рисунок 108 – Страница входа на веб-сервер контроллера (Google Chrome 79)

В поле **Username** введите имя пользователя *Administrator*, а в поле **Password** – пароль, а затем нажмите кнопку **Sign in**. Если пароль учетной записи *Administrator* ранее не был изменен, то для должен использоваться пароль *Administrator*. Более подробная информация об управлении доступом к контроллеру приведена в руководстве пользователя Fastwel PLC Application Toolkit.

При вводе неправильного имени пользователя и/или пароля, на поле **Username** будет выведена строка:

Login failed

При успешном входе на сервер в окне браузера будет отображена страница **Параметры сети** веб-конфигуратора контроллера, показанная на рисунке 109.

После каждого изменения параметров системных сервисов требуется полный перезапуск контроллера, выполняемый по нажатию кнопки **Применить конфигурацию**. В связи с этим рекомендуется настроить все необходимые системные сервисы, а затем применить конфигурацию.

Имеется возможность сохранения в файле на ПК значений параметров, которые еще не были актуализированы нажатием кнопки **Применить конфигурацию**. Для этого следует нажать **Сохранить конфигурацию в файле**. Впоследствии данный файл можно загрузить в веб-конфигуратор нажатием кнопки **Загрузить конфигурацию из файла**, а затем нажать **Применить конфигурацию**.

Для повторного чтения параметров системных сервисов контроллера следует нажать сочетание клавиш Ctrl-R.

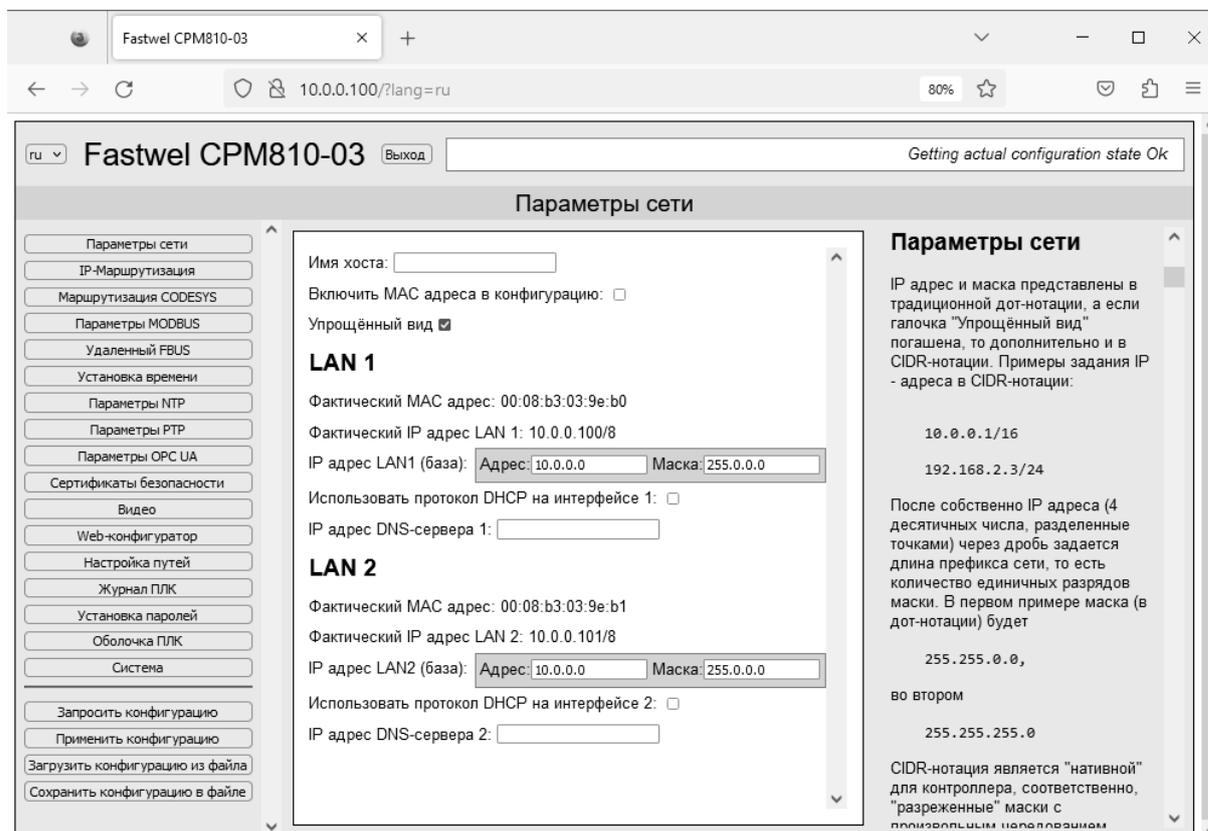


Рисунок 109 – Окно браузера при успешном входе на веб-сервер контроллера (Firefox 101.0)

4.6.2. Настройка IP-параметров

4.6.2.1. Общие сведения

Настоящий подраздел содержит информацию о способах настройки IP-параметров CPM810-03.

Системное программное обеспечение CPM810-03 поддерживает два способа назначения IP-адресов сетевым интерфейсам контроллера:

1. Статическое назначение – IP-адреса устанавливаются в веб-конфигураторе контроллера и с помощью переключателей 1–8.
2. Динамическое назначение – IP-адреса назначаются по сети с использованием протокола DHCP (Dynamic Host Configuration Protocol) сервером DHCP, который должен быть запущен на одном или нескольких узлах сети, выполняющих функции управления сетью (маршрутизаторы, управляемые коммутаторы и т.п.).

В данном документе и при настройке параметров системных сервисов контроллера используется CIDR-нотация для представления адресов IP-сети (см. п. 1.3).

В состав IP-параметров сетевых интерфейсов контроллера не входит так называемый *IP-адрес шлюза по умолчанию* (Default Gateway), которому передаются пакеты с IP-адресами получателей, для которых не удалось разрешить физические (MAC) адреса. Вместо этого сетевой системный сервис контроллера использует соответствующие записи в одной или двух *таблицах IP-маршрутизации*, которые могут быть дополнены пользователем при настройке IP-параметров контроллера.

Для доступа к веб-серверу контроллера при помощи веб-браузера требуется знать текущий IP-адрес сетевого интерфейса контроллера. Указания по выяснению IP-адресов сетевых интерфейсов контроллера приведены в п. 3.2.3 настоящего документа.

4.6.2.2. Статическое назначение IP-адресов

Статическое назначение IP-адресов является наиболее предпочтительным при построении сетей в автоматизированных системах сбора данных и управления, поскольку работоспособность сети не зависит от работоспособности отдельных узлов, на которые возложена функция динамического назначения адресов.

LAN 1

Фактический MAC адрес: 00:08:b3:03:9e:b0

Фактический IP адрес LAN 1: 10.0.0.100/8

IP адрес LAN1 (база): Адрес: Маска:

Использовать протокол DHCP на интерфейсе 1:

IP адрес DNS-сервера 1:

Рисунок 110 – IP-параметры сетевого интерфейса

Для статического назначения IP-адреса некоторому сетевому интерфейсу LANx необходимо:

1. Установить переключатели 1–8 в положение от 0 (все выключены) до 253.
2. В веб-конфигураторе задать базовое значение IP-адреса для параметра **IP адрес LANx (база)**.
3. Снять флажок **Использовать протокол DHCP на интерфейсе x**.
4. Применить параметры нажатием кнопки **Применить конфигурацию**.

Набор IP-параметров сетевого интерфейса на странице **Параметры сети** веб-конфигуратора показан на рисунке 110.

Значения IP-адресов при статическом назначении определяются следующим образом:

IP-адрес LAN1 = базовый IP-адрес 1 + swv1_8

IP-адрес LAN2 = базовый IP-адрес 2 + swv1_8

где *базовый IP-адрес x* – значение параметра **IP адрес LANx (база)**;

swv1_8 – значение, установленное на переключателях 1–8, в диапазоне от 0 до 253.

Если параметры **IP адрес LAN1 (база)** и **IP адрес LAN2 (база)** имеют одинаковое значение *ip-base*, то значения IP-адресов интерфейсов LAN1 и LAN2 определяются следующим образом:

IP-адрес LAN1 = *ip-base* + *swv1_8*

IP-адрес LAN2 = *ip-base* + *swv1_8* + 1

Если переключатели 1–8 выключены, то для параметров **IP адрес LANx (база)** должны быть заданы значения IP-адресов, отличные от адресов подсетей и направленных ширококвещательных сообщений подсетей.

Значение IP-адреса подсети определяется путем применения маски подсети к значению IP-адреса с использованием операции "логическое И".

Например, для IP-адреса 10.0.0.100/8 адрес подсети определяется следующим образом:

(10.0.0.100 AND 255.0.0.0) → **10.0.0.0**

Для IP-адреса 172.16.0.101/12 адрес подсети равен:

(172.16.0.101 AND 255.240.0.0) → **172.16.0.0**

Для IP-адреса 192.168.0.97/24 адрес определяется, как:

(192.168.0.97 AND 255.255.255.0) → **192.168.0.0**

Значение адреса направленного ширококвещательного запроса подсети образуется путем применения побитового OR инвертированной маски подсети к адресу подсети, т.е. все биты адреса подсети, соответствующие нулям в маске, устанавливаются в 1.

Например, для адреса подсети 10.0.0.0/8 адрес направленного ширококвещательного запроса определяется следующим образом:

(10.0.0.0 OR (NOT 255.0.0.0)) → **10.255.255.255**

Для адреса подсети 172.16.0.0/12 адрес направленного ширококвещательного запроса определяется так:

(172.16.0.0 OR (NOT 255.240.0.0)) → **172.31.255.255**

Наконец, для адреса подсети 192.168.0.0/24 адрес направленного широковещательного запроса:

(192.168.0.0 OR (NOT 255.255.255.0)) → **192.168.0.255**

Обратите внимание, что результат выполнения команды *ipinfo* в **Оболочке ПЛК IDE МЭК 61131-3** в блоке IP-параметров каждого интерфейса содержит значение адреса направленного широковещательного запроса в строке Broadcast.

При установке неправильных значений параметров одного или двух сетевых интерфейсов в одной подсети для них будут установлены исходные значения:

IP-адрес интерфейса LAN1: 10.0.0.100/8

IP-адрес интерфейса LAN2: 10.0.0.101/8

При установке неправильных значений параметров одного или двух сетевых интерфейсов в двух подсетях для них будут установлены исходные значения:

IP-адрес интерфейса LAN1: 10.0.0.100/24

IP-адрес интерфейса LAN2: 10.0.1.100/24

Примеры статического назначения IP-адресов показаны на рисунке 111.

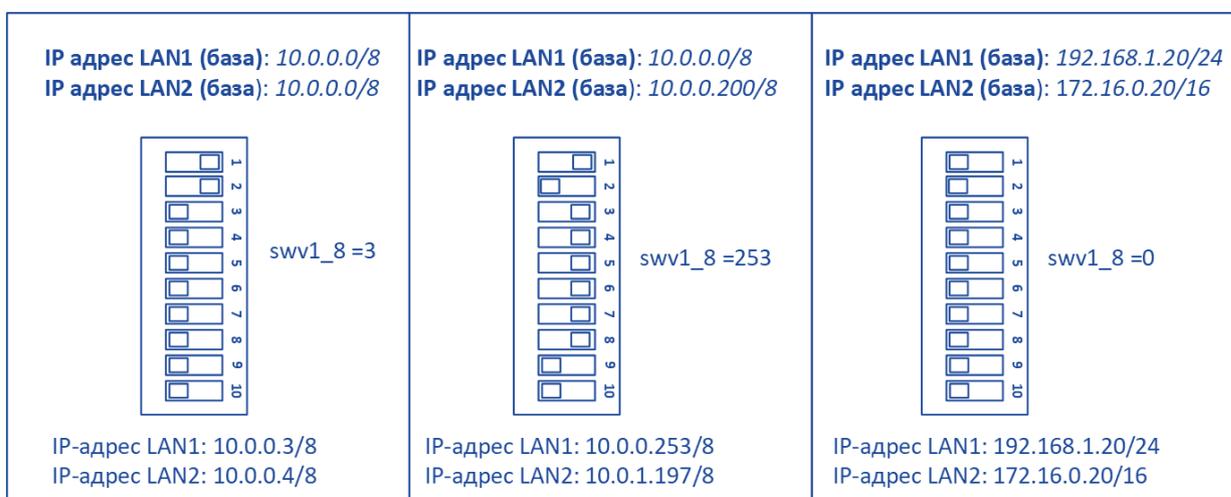


Рисунок 111 – Примеры статического назначения IP-адресов

Обратите внимание, что после изменения положения переключателей 1–8 требуется полный перезапуск контроллера выключением и повторным включением питания, нажатием кнопки сброса, выполнением команды *reboot* оболочки ПЛК в IDE МЭК 61131-3 или нажатием кнопки **Перезапустить контроллер** на странице **Система** веб-конфигуратора.

После завершения настройки системных параметров контроллера в веб-конфигураторе следует нажать кнопку **Применить конфигурацию**, что также приведет к полному перезапуску контроллера. Таким образом, можно настроить системные параметры в веб-конфигураторе, изменить положение переключателей на требуемое, а затем нажать кнопку **Применить конфигурацию** в веб-конфигураторе, чтобы новые параметры вступили в силу.

4.6.2.3. Динамическое назначение IP-адресов

Динамическое назначение IP-адресов, как правило, применяют в двух случаях:

1. При разработке приложения для одного или нескольких контроллеров на ПК с единственным сетевым интерфейсом, который подключен к общей сети предприятия с динамическим распределением адресов по протоколу DHCP и специальным выделенным набором IP-адресов, динамически назначаемым "гостевым" устройствам. Подключение контроллера к такой сети, как правило, возможно только при использовании режима динамического назначения IP-адресов.
2. При необходимости централизованного постоянного распределения IP-адресов узлам промышленной сети. В таком случае один или несколько серверов DHCP запускаются на узлах сети повышенной надежности, а в их конфигурации сервера DHCP создается список IP-адресов, распределяемых узлам сети с определенными MAC-адресами на постоянной основе. Обратите внимание, что в случае замены отдельных узлов при ремонте придется изменить MAC-адреса узлов, устанавливаемых в качестве замены.

Для динамического назначения IP-адресов всем сетевым интерфейсам контроллера СРМ810-03 следует установить значение 254 на переключателях 1–8, как показано на рисунке 112, и перезапустить контроллер выключением/включением питания, нажатием кнопки сброса или командой *reboot*, выполненной в **Оболочке ПЛК**.

Динамическое
назначение IP-адресов



Рисунок 112 – Положение переключателей для динамического назначения IP-адресов

Обратите внимание, что при установке переключателей 1–8 в положение 254, динамическое назначение IP-адресов должно быть выполнено одним сервером DHCP для всех сетевых интерфейсов контроллера.

Если требуется динамическое назначение IP-адреса только для одного сетевого интерфейса, тогда как второй сетевой интерфейс должен иметь статический IP-адрес, то следует выполнить следующие действия:

1. В веб-конфигураторе на странице **Параметры сети** для интерфейса LAN x (где x равно 1 или 2), IP-адрес которого должен быть назначен статически, введите требуемое значение базового адреса в поле **IP адрес LAN x (база)**.
2. Установите требуемое значение от 0 до 253 на переключателях 1–8 (см. п. 4.6.2.2).
3. Для второго интерфейса LAN y (где y равно 2 или 1), IP-адрес которого должен быть назначен динамически сервером DHCP, отметьте опцию **Использовать протокол DHCP на интерфейсе y** .
4. Нажмите кнопку **Применить конфигурацию**.
Произойдет перезапуск контроллера и применение параметров.

Обратите внимание, что до тех пор, пока не завершен процесс динамического назначения IP-адреса, соответствующий сетевой интерфейс не будет доступен сетевым сервисам контроллера, включая веб-сервер. Для контроля завершения назначения IP-адреса можно воспользоваться командой *ipinfo*, выполненной в **Оболочке ПЛК** при наличии соединения между средой разработки и контроллером через последовательный порт.

4.6.2.4. Настройка IP-маршрутизации

При работе контроллера в сети, которая состоит из нескольких подсетей, связанных между собой маршрутизаторами, как показано на рисунке 113, а также при работе в сети Интернет для обеспечения сетевым сервисам контроллера доступа к узлам сети, находящимся в других подсетях, потребуется настройка IP-маршрутизации.

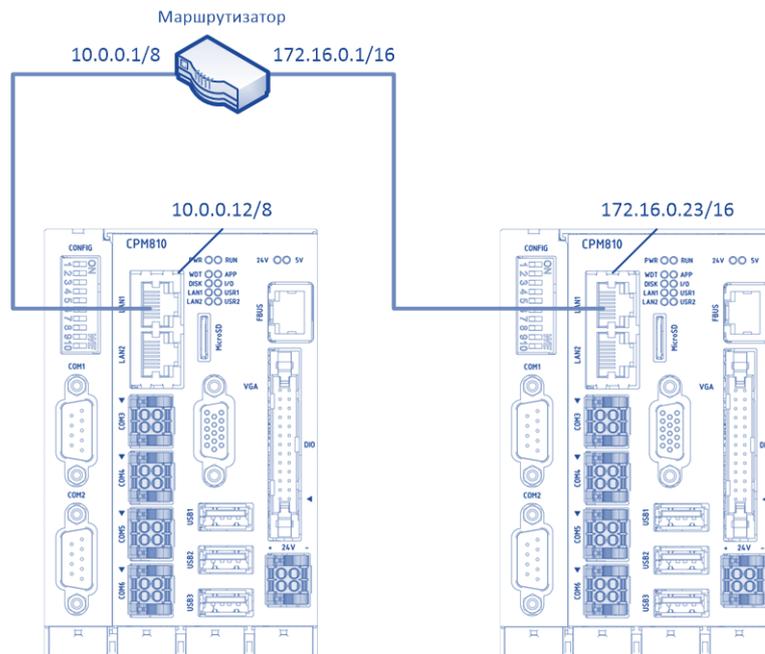


Рисунок 113 – Пример связи контроллеров через маршрутизатор

Настройка IP-маршрутизации выполняется на странице **IP-маршрутизация (IP Routing)** веб-конфигуратора и состоит в добавлении или удалении записей в одной или двух таблицах маршрутизации.

Для добавления записи в таблицу маршрутизации нажмите кнопку  и в появившемся справа от кнопки текстовом поле введите информацию о маршруте в формате:

<ip-адрес узла или подсети>/<длина маски>, <ip-адрес шлюза>

Например, если в сети, показанной на рисунке 113, всем контроллерам подсети 10.0.0.0/8, подключенным к порту маршрутизатора с IP-адресом 10.0.0.1, требуется доступ к контроллерам в подсети 172.16.0.0/16, подключенным к порту маршрутизатора с IP-адресом 172.16.0.1, то в таблицу IP-маршрутизации контроллеров в подсети 10.0.0.0/8 должна быть добавлена запись, показанная на рисунке 114. Данная запись означает, что все пакеты, отправляемые с текущего узла в подсеть с адресом 172.16.0.0/16, должны быть переданы шлюзу по адресу 10.0.0.1.

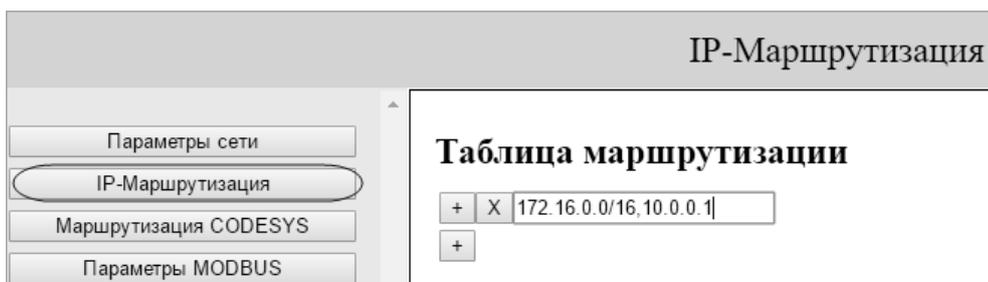


Рисунок 114 – Маршрут в подсеть 172.16.0.0/16 через шлюз 10.0.0.1

При необходимости организации маршрута к отдельному узлу, расположенному в другой подсети, например, 172.16.0.23 (см. рисунок 113), запись в таблице маршрутизации может содержать только IP-адрес целевого узла и адрес шлюза, разделенные запятой:

172.16.0.23, 10.0.0.1

Каждый сетевой интерфейс контроллера имеет отдельную таблицу маршрутизации. Если в сети, показанной на рисунке 115, сетевые сервисы контроллера в подсети 10.0.0.0/8, передающие пакеты через интерфейс LAN2 (10.0.0.13), должны иметь доступ к контроллеру в подсети 172.16.0.0/16 через его интерфейс LAN1 с IP-адресом 172.16.0.23, а сетевые сервисы на LAN1 (10.0.0.12) должны иметь доступ к контроллеру в другой подсети через интерфейс LAN2 с IP-адресом 172.16.0.24, то в таблицы маршрутизации контроллера в подсети 10.0.0.0/8 должны быть добавлены записи, показанные на рисунке 116.

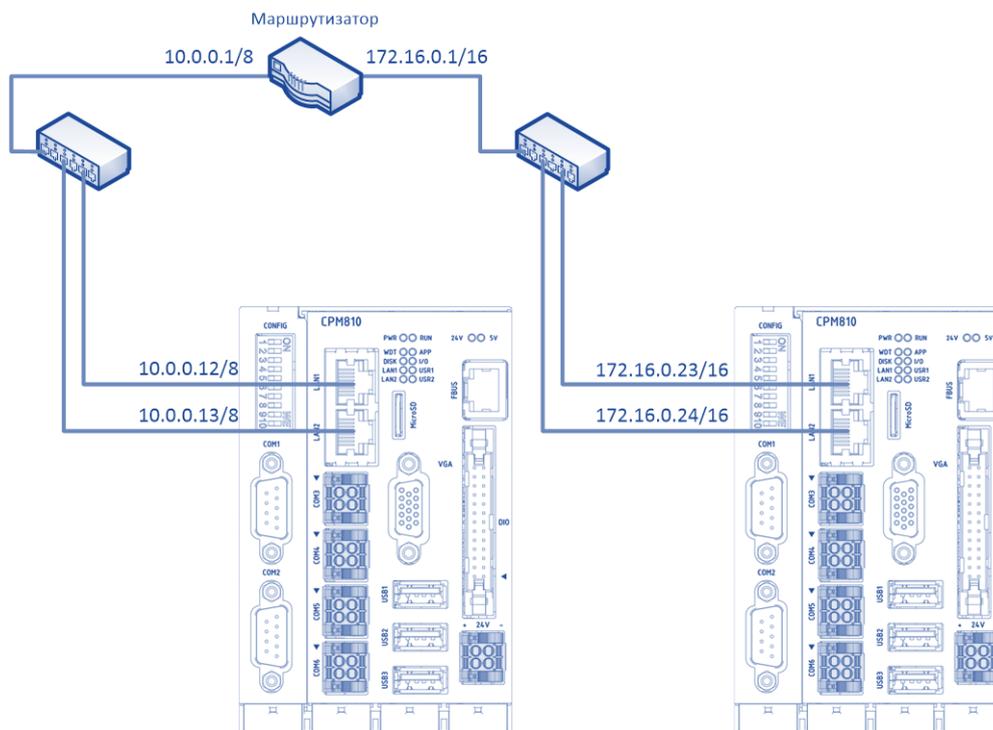


Рисунок 115 – Пример связи контроллеров через маршрутизатор

Таблица маршрутизации LAN 1	
<input type="checkbox"/>	X 172.16.0.24, 10.0.0.1
<input type="checkbox"/>	

Таблица маршрутизации LAN 2	
<input type="checkbox"/>	X 172.16.0.23, 10.0.0.1
<input type="checkbox"/>	

Рисунок 116 – Маршруты к узлам в другой подсети через шлюз 10.0.0.1

Обратите внимание, что для определения адреса шлюза по умолчанию (Default Gateway) в таблице маршрутизации соответствующего сетевого интерфейса необходимо создать запись вида:

0.0.0.0/0, <ip-адрес шлюза>

Для удаления записи из таблицы маршрутизации следует нажать кнопку

По завершении настройки IP-маршрутизации для того, чтобы новые параметры маршрутизации вступили в силу, в веб-конфигураторе необходимо нажать кнопку **Применить конфигурацию**, что приведет к перезапуску контроллера.

4.6.3. Сервис подчиненного узла протокола MODBUS

4.6.3.1. Общие сведения

Системное программное обеспечение контроллера CPM810-03 содержит встроенную реализацию сервиса подчиненного узла протоколов MODBUS TCP и MODBUS RTU/ASCII (далее MODBUS). Встроенная реализация обеспечивает наилучшую производительность при обслуживании сетевых запросов мастеров (клиентов) MODBUS и независимость исходного кода приложения для контроллера от особенностей реализации сетевых протоколов.

Конфигурация сервиса подчиненного узла MODBUS в приложении для CPM810-03 в IDE МЭК 61131-3 представлена элементами типа *Сервер Modbus* и *Сервер Modbus TCP*, которые могут быть добавлены в дерево проекта к элементу (*CPM810-03 Programmable Automation Controller*) – *Modbus* и настроены при помощи соответствующих редакторов устройств.

Конфигурация входных и выходных данных сервисов определяется в виде одной или нескольких областей принимаемых и передаваемых данных, добавляемых к элементам *Сервер Modbus* и *Сервер*

Modbus TCP, в каждой из которых может определено до 8 кбайт входных и выходных данных, как показано на рисунке 117.

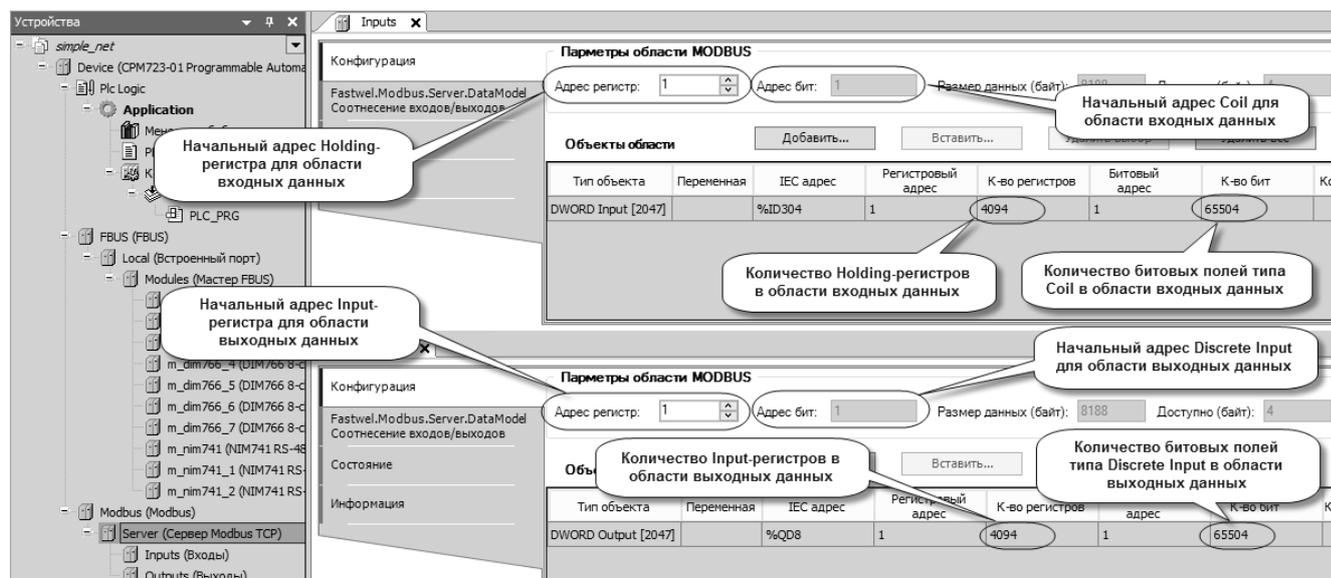


Рисунок 117 – Редакторы областей входных и выходных данных сервиса подчиненного узла MODBUS, открытые во вкладках, расположенных вертикально

Данные области типа *Входы* определяют непрерывный диапазон адресов памяти во входном образе процесса приложения и отображаются на коммуникационные объекты протокола MODBUS, принимаемые сервером от мастера MODBUS в запросах записи и чтения: Holding-регистры (регистры хранения) и битовые поля типа Coil, доступные записи и чтения.

Данные области типа *Выходы* определяют непрерывный диапазон адресов в выходном образе процесса приложения и отображаются на коммуникационные объекты протокола MODBUS, передаваемые сервером в ответ на запросы чтения от мастера: Input-регистры (входные регистры) и битовые поля, доступные для чтения (объекты типа Discrete Input).

В конфигурации каждого экземпляра сервера, добавленного в проект, возможно создать несколько областей типа *Входы* и *Выходы* с разными начальными регистровыми адресами, которые суммарно определяют до 32767 регистров и до 65536 битовых полей на ввод и столько же на вывод. Таким образом, суммарные размеры входных и выходных данных одного экземпляра сервера MODBUS составляет по 65534 байт соответственно.

Соотнесение переменных приложения с областями данных серверов MODBUS выполняется в редакторе образа процесса, отображаемом на соответствующей вкладке по команде меню **Изменить И/О-соотнесение**, или на вкладке **Соотнесение входов/выходов** в редакторах устройств каждой области.

В конфигурации сервера MODBUS TCP и RTU/ASCII имеется группа параметров **Конфигурация областей данных MODBUS** (см. рисунок 119), которая может использоваться для дополнительной настройки нескольких областей ввода-вывода, определенных в конфигурации сервера.

Опция **Запретить запросы к смежным областям** позволяет предотвратить ошибочный доступ клиентов MODBUS TCP к данным нескольких областей ввода или вывода в конфигурации сервера путем передачи серверу запросов записи или чтения регистров или битовых полей, относящихся к разным областям. Пусть, например, в конфигурации сервера определены две области входных данных, *Inputs_0_1* и *Inputs_0_2*, содержащие по 8 байт, с начальными адресами Holding-регистров (регистров хранения) 1 и 5 соответственно, как показано на рисунке 118.

Если для сервера не установлена опция **Запретить запросы к смежным областям**, то мастер MODBUS имеет возможность выполнять запросы чтения и записи данных, относящихся к разным областям со смежными адресами регистров, скажем, с начальным адресом регистра 3 (в пакете запроса 2) и количеством регистров 3. Если с МЭК-адресами этих областей соотнесены структурные переменные или переменные размером более 2 байт, то такие запросы записи могут привести к нарушению целостности данных сервера и, быть может, к переходу контроллера в безопасный режим при ошибке целочисленного деления на 0.

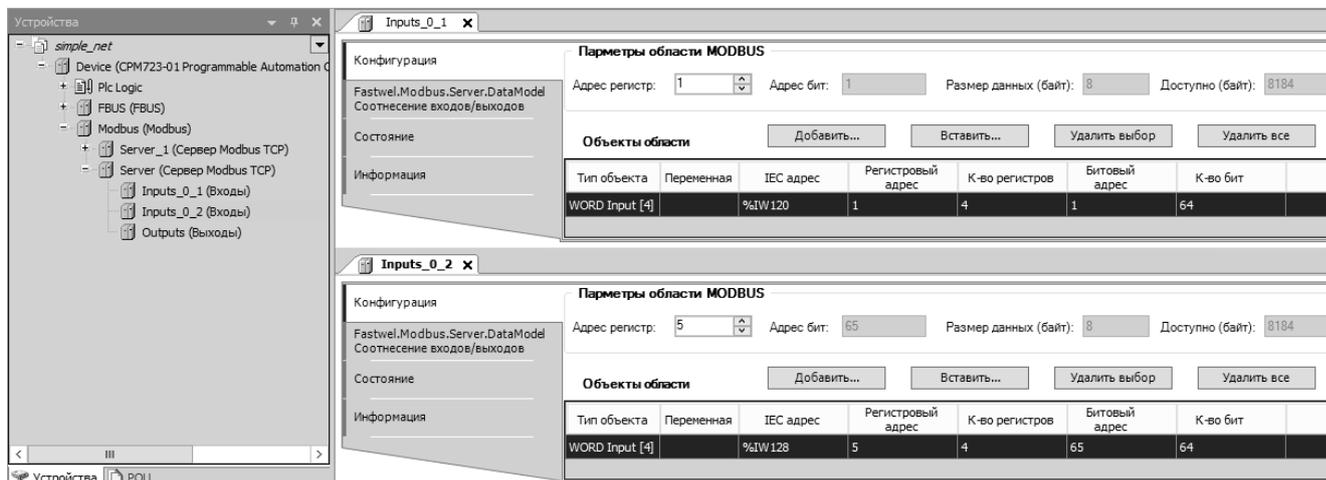


Рисунок 118 – Две области входных данных сервера MODBUS со смежными адресами регистров

Если для сервера установлена опция **Запретить запросы к смежным областям**, то мастер MODBUS будет получать код ошибки в ответ на запросы к областям данных со смежными адресами регистров и битовых полей.

Более подробная информация о конфигурировании сервиса подчиненного узла MODBUS TCP и MODBUS RTU/ASCII приведена в документе *ИМЕС.00300-03 33 03. Протокол MODBUS. Руководство по конфигурированию и программированию*.

Настоящий подраздел содержит информацию об особенностях использования сервиса на контроллере CPM810-03. Обратите внимание, что при вводе значений параметров в текстовые поля на вкладках редакторов устройств для применения изменений следует сохранить проект или переместить фокус ввода на другой элемент пользовательского интерфейса.

4.6.3.2. Сервис подчиненного узла протокола MODBUS TCP

На контроллере CPM810-03 может быть запущен один или два сервера MODBUS TCP, и при этом возможны следующие варианты соотнесения серверов с сетевыми интерфейсами контроллера:

1. Один сервер в конфигурации приложения (один *экземпляр*) обеспечивает обмен данными с мастерами MODBUS TCP в количестве до 32 через все доступные сетевые интерфейсы контроллера. Все мастера MODBUS TCP, установившие соединение с сервером, получают доступ к одним и тем же областям данных, определенных в конфигурации сервера.
Для реализации требуется только добавить и настроить один экземпляр сервера в конфигурации приложения.
2. Два сервера в конфигурации приложения (два *экземпляра*) обеспечивают обмен данными с мастерами через два сетевых интерфейса контроллера, настроенных на работу в одной или двух подсетях. При этом с каждым сервером могут установить соединение до 32 мастеров (клиентов) MODBUS TCP, и каждый мастер, установивший соединение с некоторым экземпляром сервера, получает доступ к областям входных и выходных данных этого экземпляра.

В данном случае каждый экземпляр сервера должен иметь собственный уникальный идентификатор (0 и 1), определяемый параметром **Идентификатор сервера MODBUS TCP** в редакторе устройства *Сервер Modbus TCP* (см. рисунок 119), и идентификаторы должны быть соотнесены с IP-адресами сетевых интерфейсов в веб-конфигураторе на странице **Параметры MODBUS**.

3. Два экземпляра сервера в конфигурации приложения обеспечивают обмен данными с мастерами через один сетевой интерфейс контроллера. С каждым сервером могут установить соединение до 32 мастеров (клиентов) MODBUS TCP, и каждый мастер, установивший соединение с некоторым экземпляром сервера, получает доступ к областям входных и выходных данных этого экземпляра.

Для реализации каждый экземпляр сервера также должен иметь собственный уникальный идентификатор (0 и 1), определяемый параметром **Идентификатор сервера MODBUS TCP** в редакторе устройства *Сервер Modbus TCP*. При этом на странице **Параметры MODBUS** веб-конфигуратора идентификаторы серверов должны

быть соотнесены с IP-адресом одного сетевого интерфейса и для них должны быть заданы разные значения параметра **TCP порт**.

Рисунок 119 – Параметры сервера MODBUS TCP

Три перечисленных варианта соотнесения экземпляров сервера MODBUS TCP с сетевыми интерфейсами контроллера представлены в таблице 2.

Вкладка редактора конфигурации сервера MODBUS TCP показана на рисунке 119.

Параметр **Идентификатор сервера MODBUS TCP** предназначен для соотнесения экземпляра сервера с IP-адресом сетевого интерфейса контроллера. При наличии только одного сервера в конфигурации приложения значение данного параметра должно быть равно 0. При использовании двух экземпляров сервера в конфигурации, для них должны быть установлены разные значения идентификаторов, а в веб-конфигураторе настроено соотнесение каждого экземпляра с IP-адресом интерфейса.

Параметр **ID узла (Unit ID)** определяет идентификатор подчиненного устройства, функционирующего под управлением данного сервера MODBUS TCP, и передается со стороны клиентов (мастеров) в запросах чтения и записи серверу. Обратите внимание, что в мастерах (клиентах) MODBUS TCP некоторых производителей данный параметр по умолчанию может иметь значение 1.

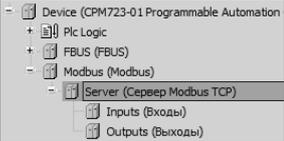
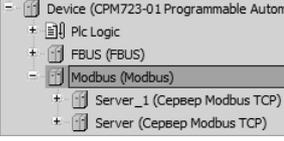
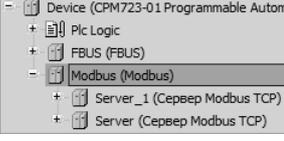
Если отмечен флажок **Подсеть MODBUS RTU**, то настраиваемый экземпляр сервера MODBUS TCP будет также выполнять функцию шлюза в сеть MODBUS RTU через последовательный порт на базе модуля NIM741, NIM841, NIM742 или NIM842, чей числовой идентификатор задан в поле **COM порт**, а параметры обмена и таймаут ответа – в остальных полях группы **Параметры шлюза MODBUS TCP/RTU**.

Обратите внимание, что при включении шлюза в сеть MODBUS RTU параметр **ID узла (Unit ID)** должен быть равен 0. Тогда все запросы чтения и записи к данному экземпляру сервера MODBUS TCP, содержащие 0 в поле Unit ID заголовка пакета MODBUS TCP, будут использоваться для обмена данными с областями входных и выходных данных данного экземпляра сервера, а запросы, содержащие значения от 1 до 255 в поле Unit ID заголовка пакета MODBUS TCP, будут передаваться узлам с соответствующими адресами в сеть MODBUS RTU, подключенную к последовательному порту с номером **COM порт**. Полученные ответы от подчиненных узлов MODBUS будут передаваться запрашивающему мастеру MODBUS TCP.

Информация об идентификации последовательных портов на базе модулей NIM741 и NIM742 приведена в п. 4.5.4.

Краткое описание назначения группы параметров **Конфигурация областей данных MODBUS** приведено в п. 4.6.3.1

Таблица 2 – Варианты соотношения экземпляров сервера MODBUS TCP с сетевыми интерфейсами

Вариант	Конфигурация приложения	Параметры MODBUS в веб-конфигураторе
Один экземпляр сервера установлен на все сетевые интерфейсы	<p>Один экземпляр сервера в дереве проекта:</p>  <p>Параметр Идентификатор сервера MODBUS TCP: 0.</p>	<p>Настройка не требуется, или должны использоваться параметры по умолчанию:</p> <p>Серверы MODBUS:</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 0</p> <p>Локальный IP: По умолчанию ▾</p> <p>Порт: 502</p>
Каждый экземпляр сервера установлен на собственный сетевой интерфейс	<p>Два экземпляра сервера в дереве проекта:</p>  <p>Значения параметра Идентификатор сервера MODBUS TCP каждого экземпляра сервера должны отличаться, например, 0 и 1.</p>	<p>Должно быть выполнено явное соотношение идентификатора каждого сервера с IP-адресом требуемого сетевого интерфейса, например 0 – LAN1, 1 – LAN2:</p> <p>Серверы MODBUS:</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 0</p> <p>Локальный IP: IP адрес 1 (LAN 1) ▾</p> <p>Порт: 502</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 1</p> <p>Локальный IP: IP адрес 2 (LAN 2) ▾</p> <p>Порт: 502</p>
Все экземпляры сервера установлены на один сетевой интерфейс LAN2	<p>Два экземпляра сервера в дереве проекта:</p>  <p>Значения параметра Идентификатор сервера MODBUS TCP каждого экземпляра сервера должны отличаться, например, 0 и 1.</p>	<p>Должно быть выполнено явное соотношение идентификатора каждого сервера с IP-адресом одного сетевого интерфейса, и установлены разные TCP-порты, например 0 – LAN2:502, 1 – LAN2:503</p> <p>Серверы MODBUS:</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 0</p> <p>Локальный IP: IP адрес 2 (LAN 2) ▾</p> <p>Порт: 502</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 1</p> <p>Локальный IP: IP адрес 2 (LAN 2) ▾</p> <p>Порт: 503</p>

4.6.3.3. Сервис подчиненного узла протокола MODBUS RTU/ASCII

На контроллере CPM810-03 может быть запущено до 16 серверов (экземпляров сервера) MODBUS RTU или ASCII, функционирующих через встроенные последовательные порты и порты на базе модулей NIM741, NIM841, NIM742 или NIM842, подключенные к локальной шине ввода-вывода контроллера.

Для встроенных портов интерфейса RS-485 контроллера и для модуля NIM841 Fastwel I/O-2 цепи окончного согласования могут быть подключены к линии переключателями.

Настройка параметров сервера MODBUS RTU/ASCII выполняется на вкладке редактора конфигурации элемента *Сервер Modbus*, показанной на рисунке 120.

Поле **Адрес узла** должно содержать допустимые значения адреса станции в сети MODBUS в диапазоне от 1 до 247.

Рисунок 120 – Параметры сервера MODBUS

Поле **COM порт** предназначено для установки числового идентификатора последовательного порта, на который устанавливается сервер. Информация об идентификации последовательных портов на базе модулей NIM741, NIM841, NIM742, NIM842 приведена в п. 4.5.4.

Выпадающий список **Тип протокола** позволяет выбрать тип реализуемого протокола: *RTU* – потоковый, *ASCII* – символьный.

Параметры **Скорость обмена**, **Количество бит данных**, **Четность** и **Количество стоп-битов** предназначены для установки скорости обмена и формата одиночных кадров, принимаемых и передаваемых по сети.

Краткое описание назначения группы параметров **Конфигурация областей данных MODBUS** приведено в п. 4.6.3.1.

4.6.4. Сервис мастера протокола MODBUS

4.6.4.1. Общие сведения

Системное программное обеспечение контроллера CPM810-03 содержит встроенную реализацию сервиса мастера (клиента) протоколов MODBUS TCP и MODBUS RTU/ASCII (далее MODBUS).

Конфигурация сервиса мастера MODBUS в приложении для CPM810-03 в IDE МЭК 61131-3 представлена элементами типа *Master Modbus* и *Master Modbus TCP*, которые могут быть добавлены в дерево проекта к элементу (*CPM810-03 Programmable Automation Controller*) – *Modbus* и настроены при помощи соответствующих редакторов устройств.

Конфигурация мастера MODBUS должна состоять из списка опрашиваемых подчиненных узлов (элементов типа *Подчиненный узел*), для каждого из которых описаны запросы записи и чтения данных (элементы типа *Коммуникационный объект*), определяющие области входных и выходных данных мастера.

Общим параметром конфигурации мастера MODBUS TCP и мастера MODBUS RTU/ASCII является **Гранулярность опроса, мс** (диапазон от 1 до 60000 мс), который определяет дискретность установки значений периода циклической передачи сетевых запросов с типом передачи от 1 до 250.

Пример конфигурации двух запросов к подчиненному узлу MODBUS представлен на рисунке 121.

Вкладка редактора *WriteRegs_1_123* содержит описание запроса записи 123 регистров хранения (10h – Write Multiple Registers) с начальным адресом регистра 1 (по сети передается на 1 меньше значения адреса), а вкладка редактора *ReadRegs_1_125* содержит описание запроса чтения 125 входных регистров (04h – Read Input Registers).

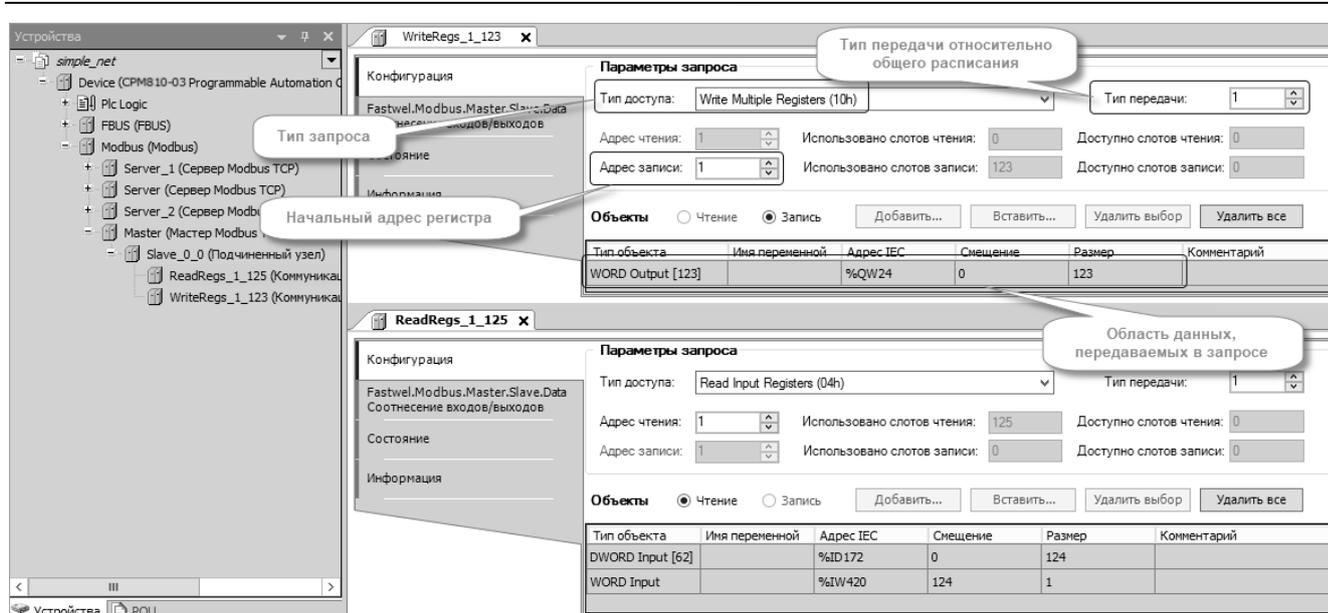


Рисунок 121 – Пример конфигурации мастера MODBUS TCP

Параметр Тип передачи:

- при равенстве 0 исключает данный запрос из расписания опроса подчиненных узлов;
- при равенстве от 1 до 250 определяет общее количество интервалов, заданных параметром **Гранулярность опроса, мс**, по истечении которых данный запрос будет в очередной раз передан подчиненному узлу;
- значения 251–254 зарезервированы и в текущей реализации эквивалентны значению 255;
- при равенстве 255 предписывает передачу запросов записи только при изменении данных, которые должны быть переданы подчиненному узлу.

Конфигурация входных и выходных данных каждого мастера определяется в виде двух областей во входном и выходном образе процесса приложения:

1. Область входных данных мастера состоит из областей диагностических данных каждого подчиненного узла и областей данных, определенных для всех запросов чтения в конфигурации всех подчиненных узлов.
2. Область выходных данных мастера состоит из областей данных, определенных для всех запросов записи в конфигурации подчиненных узлов.

Максимальные размеры областей входных и выходных данных в зависимости от типа запроса перечислены в таблице 3.

Таблица 3 – Информационная емкость запросов к подчиненным узлам

Тип запроса	Образ процесса		Размер		
	Входной (%I)	Выходной (%Q)	Бит	Регистров	Байт
Read Coils (01h)	√	–	2000	–	250
Read Discrete Inputs (02h)	√	–	2000	–	250
Read Holding Registers (03h)	√	–	–	125	250
Read Input Registers (04h)	√	–	–	125	250
Write Single Coil (05h)	–	√	1	–	–
Write Single Register (06h)	–	√	–	1	2
Write Multiple Coils (0Fh)	–	√	1968	–	246
Write Multiple Registers (10h)	–	√	–	123	246
Read/Write Multiple Registers (17h)	√	√	–	125/121	250/242

Обратите внимание, что в запросах чтения и записи битовых полей типа Coils и Discrete Inputs определяются объекты данных типа *1-Bit Input*, *8-Bit Input* или *1-Bit Output*, *8-Bit Output*, содержащие каналы типа BIT. Тип BIT, в отличие от типа BOOL, обеспечивает возможность последовательного размещения (упаковки) нескольких битовых полей во входном или выходном образе процесса. Например, от 1 до 8 объектов типа *1-Bit Input* размещаются в одном байте входного образа процесса приложения.

Соотнесение переменных приложения с областями данных мастеров MODBUS выполняется в редакторе образа процесса, отображаемом на соответствующей вкладке по команде меню **Изменить I/O-соотнесение**, или на вкладке **Соотнесение входов/выходов** в редакторах устройств каждого области.

Более подробная информация о конфигурировании сервиса мастера MODBUS TCP и MODBUS RTU/ASCII приведена в документе *ИМЕС.00300-03 33 03. Протокол MODBUS. Руководство по конфигурированию и программированию*.

Настоящий подраздел содержит информацию об особенностях использования сервиса на контроллере СРМ810-03. Обратите внимание, что при вводе значений параметров в текстовые поля на вкладках редакторов устройств для применения изменений следует сохранить проект или переместить фокус ввода на другой элемент пользовательского интерфейса.

4.6.4.2. Сервис мастера протокола MODBUS TCP

На контроллере СРМ810-03 может быть запущен один или два независимых экземпляра мастера MODBUS TCP.

При использовании двух экземпляров мастера каждому из них должен быть назначен уникальный числовой идентификатор, определяемый параметром **Идентификатор мастера MODBUS TCP** в редакторе устройства *Мастер Modbus TCP*, показанном на рисунке 122.

Обратите внимание, что соотнесение экземпляров мастера MODBUS TCP с сетевыми интерфейсами контроллера практически всегда требуется только тогда, когда оба сетевых интерфейса имеют IP-адреса в одной подсети. Это связано с тем, что в данном режиме сетевой сервис контроллера использует две независимые таблицы IP-маршрутизации (см. п. 4.6.2.4), и при передаче в сеть пакета с некоторым IP-адресом назначения со стороны любого сетевого сервиса требуется знать, через какой физический интерфейс должен быть передан пакет.

Если сетевые интерфейсы контроллера имеют IP-адреса в одной подсети, и при этом в веб-конфигураторе не выполнено соотнесение экземпляров мастера MODBUS TCP с IP-адресами сетевых интерфейсов, то все исходящие пакеты протокола MODBUS TCP и соответствующие служебные пакеты протокола транспортного уровня TCP будут передаваться в сеть через интерфейс LAN1.

Возможны следующие варианты соотнесения экземпляров мастера с сетевыми интерфейсами контроллера:

1. Один или два экземпляра мастера в конфигурации приложения обеспечивают обмен данными с подчиненными узлами MODBUS TCP через интерфейс LAN1 контроллера. Для реализации требуется добавить и настроить один или два экземпляра мастера в конфигурации приложения и, в случае двух экземпляров, каждому установить уникальное значение параметра **Идентификатор мастера MODBUS TCP** (0 или 1) в редакторе устройства *Мастер Modbus TCP*, показанном на рисунке 122.
2. Два экземпляра мастера в конфигурации приложения обеспечивают обмен данными с подчиненными узлами через отдельные сетевые интерфейсы контроллера, LAN1 и LAN2, имеющие IP-адреса в одной подсети.
В данном случае каждый экземпляр мастера должен иметь собственный уникальный идентификатор (0 и 1), определяемый параметром **Идентификатор мастера MODBUS TCP** в редакторе устройства *Сервер Modbus TCP* (см. рисунок 122), и идентификаторы мастеров должны быть соотнесены с IP-адресами разных сетевых интерфейсов в веб-конфигураторе на странице **Параметры MODBUS**.
3. Два экземпляра мастера в конфигурации приложения обеспечивают обмен данными с подчиненными узлами через один из двух сетевых интерфейсов контроллера, с IP-адресами в одной или двух подсетях.
Для реализации каждый экземпляр мастера должен иметь собственный уникальный идентификатор (0 и 1), определяемый параметром **Идентификатор мастера MODBUS TCP** в редакторе устройства *Сервер Modbus TCP* (см. рисунок 122), и идентификаторы мастеров должны быть соотнесены с IP-адресами одного сетевого интерфейса в веб-конфигураторе на странице **Параметры MODBUS**.

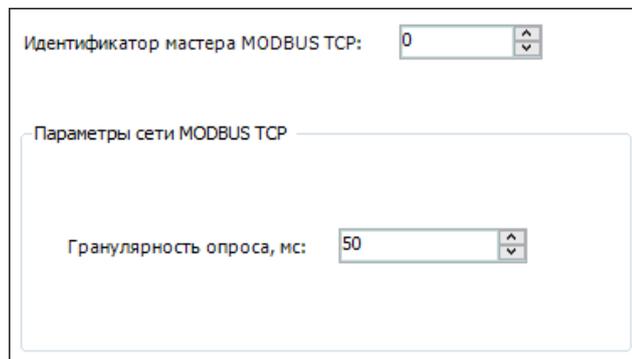


Рисунок 122 – Параметры мастера MODBUS TCP

Перечисленные выше варианты соотношения экземпляров мастера MODBUS TCP с сетевыми интерфейсами контроллера с IP-адресами в одной подсети представлены в таблице 4.

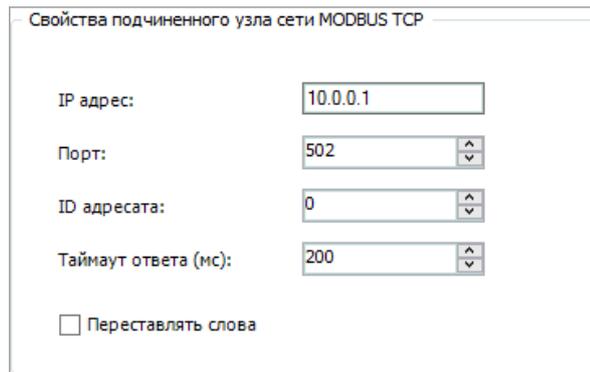
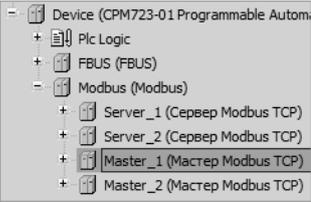


Рисунок 123 – Параметры подчиненного устройства в конфигурации мастера MODBUS TCP

Таблица 4 – Варианты соотношения экземпляров мастера MODBUS TCP с интерфейсами в одной подсети

Вариант	Конфигурация приложения	Параметры MODBUS в веб-конфигураторе
Один или два экземпляра мастера MODBUS TCP функционируют через интерфейс LAN1	<p>Один или два экземпляра мастера в дереве проекта:</p> <p>Значения параметра Идентификатор мастера MODBUS TCP каждого экземпляра мастера должны отличаться, например, 0 и 1.</p>	<p>Настройка не требуется, или должны использоваться параметры по умолчанию:</p> <p>Мастера MODBUS:</p>
Каждый экземпляр мастера MODBUS TCP функционирует через отдельный сетевой интерфейс	<p>Один или два экземпляра мастера в дереве проекта:</p> <p>Значения параметра Идентификатор мастера MODBUS TCP каждого экземпляра мастера должны отличаться, например, 0 и 1.</p>	<p>Должно быть выполнено явное соотношение идентификатора каждого мастера с IP-адресом требуемого сетевого интерфейса, например 0 – LAN1, 1 – LAN2:</p> <p>Мастера MODBUS:</p>

Вариант	Конфигурация приложения	Параметры MODBUS в веб-конфигураторе
Все экземпляры мастера MODBUS TCP функционируют через один сетевой интерфейс LAN2	<p>Один или два экземпляра мастера в дереве проекта:</p>  <p>Значения параметра Идентификатор мастера MODBUS TCP каждого экземпляра мастера должны отличаться, например, 0 и 1.</p>	<p>Должно быть выполнено явное соотнесение идентификатора каждого мастера с IP-адресом требуемого сетевого интерфейса:</p> <p>Мастера MODBUS:</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 0</p> <p>Локальный IP: IP адрес 2 (LAN 2) ▾</p> <p>+ X</p> <p>Идентификатор (как в проекте CODESYS): 1</p> <p>Локальный IP: IP адрес 2 (LAN 2) ▾</p>

Как указывалось в п. 4.6.4.1, в конфигурацию каждого экземпляра мастера MODBUS TCP должны быть включены описания подчиненных узлов (устройств), подлежащих опросу. Обратите внимание, что обмен данными с каждым подчиненным узлом MODBUS TCP выполняется независимо от обмена с остальными подчиненными узлами, поэтому отсутствие связи с любым из подчиненных узлов не влияет на своевременность опроса узлов, с которыми есть связь.

Вкладка редактора устройства для элемента конфигурации *Подчиненный узел*, содержащая параметры подчиненного узла MODBUS TCP, показана на рисунке 123.

Параметр **IP адрес** определяет IP-адрес подчиненного узла. Если предполагается обмениваться данными с подчиненными узлами, находящимися за пределами подсетей, к которым подключен контроллер, следует при помощи веб-конфигуратора настроить маршруты к таким подчиненным узлам в соответствии с указаниями п. 4.6.2.4.

Параметр **Порт** должен содержать номер TCP-порта, на котором установлен сервер MODBUS TCP на подчиненном узле. По умолчанию используется порт 502, закрепленный за протоколом MODBUS TCP.

Параметр **ID адресата** содержит идентификатор подчиненного устройства, функционирующего под управлением удаленного сервера MODBUS TCP, который передается мастером в сетевых запросах к удаленному серверу в заголовке пакета MODBUS TCP. Назначение данного параметра идентично параметру **ID узла (Unit ID)** сервера MODBUS TCP, описание которого приведено в п. 4.6.4.2.

Параметр **Таймаут ответа (мс)** определяет интервал времени, в течение которого мастер ожидает ответа на очередной запрос к удаленному серверу. Данный параметр может иметь значение в диапазоне от 20 до 10000 мс. Если по истечении интервала времени, заданного параметром **Таймаут ответа (мс)**, не получен ответ от подчиненного узла, в области входных диагностических каналов счетчик ошибок *TimeoutErrorsCount* увеличивается на 1.

Опция **Переставлять слова** предписывает мастеру переставлять слова при передаче/приеме данных длиной более двух байт. Данные могут располагаться в памяти подчиненного узла с использованием как Big-Endian, так и Little-Endian размещения. Для подчиненных узлов, использующих Big-Endian размещение данных в памяти, необходимо использовать данную опцию

4.6.4.3. Сервис мастера протокола MODBUS RTU/ASCII

На контроллере CPM810-03 может быть запущено до 16 мастеров (экземпляров мастера) MODBUS RTU или ASCII, функционирующих через встроенные последовательные порты или порты на базе модулей NIM741, NIM841, NIM742 или NIM842, подключенные к локальной шине ввода-вывода контроллера.

Настройка параметров сервера MODBUS RTU/ASCII выполняется на вкладке редактора конфигурации элемента *Мастер Modbus*, показанной на рисунке 124.

Параметры сети MODBUS SERIAL

Гранулярность опроса, мс:	50
COM порт:	1
Тип протокола:	RTU
Скорость обмена:	38400
Количество бит данных:	8
Четность:	EVEN
Количество стоп-битов:	1

Рисунок 124 – Параметры мастера MODBUS RTU или ASCII

Описание параметра **Гранулярность опроса** приведено в п. 4.6.4.1.

Поле **COM порт** предназначено для установки числового идентификатора последовательного порта, через который должен функционировать мастер. Информация об идентификации последовательных портов на базе модулей NIM741, NIM841, NIM742 или NIM842 приведена в п. 4.5.4.

Выпадающий список **Тип протокола** позволяет выбрать тип реализуемого протокола: *RTU* – потоковый, *ASCII* – символьный.

Параметры **Скорость обмена**, **Количество бит данных**, **Четность** и **Количество стоп-битов** предназначены для установки скорости обмена и формата одиночных кадров, принимаемых и передаваемых по сети.

Свойства подчиненного узла сети MODBUS SERIAL

Адрес узла:	1
Таймаут ответа (мс):	200
<input type="checkbox"/> Переставлять слова	
Регулировка частоты опроса	
<input type="checkbox"/> Включена	
Кол-во таймаутов подряд :	0
Ограничение периода опроса (мс):	0

Рисунок 125 – Параметры подчиненного устройства в конфигурации мастера MODBUS RTU/ASCII

Как указывалось в п. 4.6.4.1, в конфигурацию каждого экземпляра мастера MODBUS должны быть включены описания подчиненных узлов (устройств), подлежащих опросу. Вкладка редактора устройства для элемента конфигурации *Подчиненный узел*, содержащая параметры подчиненного узла MODBUS RTU или ASCII, показана на рисунке 125.

Параметр **Адрес узла** определяет адрес подчиненного устройства (станции) в сети MODBUS в диапазоне от 1 до 247.

Параметр **Таймаут ответа (мс)** определяет интервал времени, в течение которого мастер ожидает ответ на очередной запрос к подчиненному узлу. Данный параметр может иметь значение в диапазоне от 20 до 10000 мс. Если по истечении интервала времени, заданного параметром **Таймаут ответа (мс)**, не получен ответ от подчиненного узла, в области входных диагностических каналов данного подчиненного узла счетчик ошибок *TimeoutErrorsCount* увеличивается на 1.

Обратите внимание, что обмен данными с подчиненными узлами MODBUS RTU или ASCII, обслуживаемыми одним мастером, выполняется последовательно. Таким образом, при потере или отсутствии связи с некоторым подчиненным узлом полное время опроса остальных узлов, обслуживаемых данным мастером сети, увеличится на интервал времени, длительность которого равна произведению таймаута ожидания ответа от данного подчиненного узла на количество запросов, которое требовалось передать в его адрес на текущем цикле сети.

В связи с этим рекомендуется использовать минимально возможное значение таймаута ответа и минимально возможное количество запросов к каждому подчиненному узлу, по возможности отдавая предпочтение групповым запросам чтения и записи данных Read Input Registers (04h), Write Multiple Registers (10h), Read/Write Multiple Registers (17h), Read Holding Registers (03h), Write Multiple Coils (0Fh).

Дополнительным способом решения проблемы потери или отсутствия связи с некоторыми подчиненными узлами является использование механизма регулировки частоты опроса с подчиненными узлами, с которыми утрачена связь. Для включения и настройки механизма регулировки следует в конфигурации каждого подчиненного узла отметить флажок **Регулировка частоты опроса – Включена**, после чего в поле **Количество таймаутов подряд** (диапазон от 1 до 65000, 0 – выключает механизм независимо от опции **Включена**) установить количество таймаутов обмена с данным подчиненным узлом, последовавших друг за другом, после которых период опроса данного подчиненного узла будет увеличен до значения, установленного в поле **Ограничение периода опроса (мс)**. Таким образом, до восстановления связи с данным подчиненным узлом он будет опрашиваться значительно реже, чем определено расписанием согласно гранулярности опроса, установленной для мастера, и значениями параметра **Тип передачи** для всех сетевых запросов, относящихся к подчиненному узлу, с которым утрачена связь.

После восстановления связи с данным подчиненным узлом для него будет восстановлено исходное расписание опроса.

4.6.5. Сервис протокола NTP

4.6.5.1. Общие сведения

Системное программное обеспечение контроллера СРМ810-03 имеет в своем составе сервис клиента синхронизации времени по сети по протоколу NTP v4, обеспечивающий возможность установки и синхронизации системного времени контроллера по сети от одного или нескольких серверов NTP.

Сервис NTP контроллера также может выполнять функции сервера NTP для установки и синхронизации времени на других узлах сети, на которых запущены клиенты NTP.

Обратите внимание, что при использовании сервиса NTP недопустимо использовать динамическое назначение IP-адресов интерфейсам, к которым привязывается сервис NTP.

Активизация и настройка параметров сервис NTP выполняются в веб-конфигураторе контроллера на странице **Параметры NTP**, показанной на рисунке 126.

Рисунок 126 – Страница настройки параметров сервиса NTP в веб-конфигураторе

4.6.5.2. Настройка клиента NTP

Для настройки сервиса в качестве клиента NTP, получающего время от одного удаленного сервера NTP или нескольких, отметьте опцию **Разрешить NTP**, с помощью кнопки **+** добавьте запись в список **Внешние NTP серверы** и введите IP-адрес серверов NTP. Опция **prefer** предписывает сервису использовать выбранный сервер NTP в качестве основного. Опция **iburst** предназначена для формирования нескольких подряд запросов к серверу при запуске контроллера, что позволяет ускорить процесс синхронизации.

Список **Привязка сервиса NTP к интерфейсам** предназначен для соотнесения сервиса с сетевыми интерфейсами контроллера. При отсутствии элементов в списке клиент NTP будет посылать запросы к удаленным серверам через сетевой интерфейс LAN1.

При наличии более одного элемента в списке привязки клиент NTP посылает запросы к удаленным серверам NTP через сетевой интерфейс, указанный в качестве последнего элемента в списке.

Характерные примеры конфигурации клиента NTP представлены в таблице 5.

Обратите внимание, что при необходимости использования сервиса NTP только в качестве клиента, синхронизирующего время с удаленными серверами NTP, параметр **LocalStratum** должен быть пустым или равным 15.

Алгоритм синхронизации времени, используемый сервисом NTP, для предотвращения неправильного поведения приложения контроллера в случае резких скачков системного времени выполняет плавное приближение системного времени к времени текущего выбранного сервера NTP. Минимальное время синхронизации составляет от 10 до 20 минут и зависит от стабильности хода часов на выбранном сервере.

По завершении настройки сервиса NTP нажмите **Применить конфигурацию** на главной странице веб-конфигуратора.

Таблица 5 – Примеры конфигурации клиента NTP

Вариант	Конфигурация сетевых интерфейсов	Конфигурация NTP
<p>LAN1: базовый адрес 172.16.0.0/16 LAN2: базовый адрес 10.0.0.0/8 Только клиент NTP, требуется посылать запросы к серверам 10.0.0.239 и 10.0.0.5 через интерфейс LAN2</p>	<p>Имя хоста: <input type="text"/></p> <p>Включить MAC адреса в конфигурацию: <input type="checkbox"/></p> <p>LAN 1</p> <p>Фактический MAC адрес: 00:08:b3:02:9d:e1 Фактический IP адрес LAN 1: 10.0.0.3/8 IP адрес LAN1 (база): <input type="text" value="10.0.0.0/8"/></p> <p>Использовать протокол DHCP на интерфейсе 1: <input type="checkbox"/></p> <p>IP адрес DNS-сервера 1: <input type="text"/></p> <p>LAN 2</p> <p>Фактический MAC адрес: 00:08:b3:02:9d:e2 Фактический IP адрес LAN 2: 172.16.0.3/16 IP адрес LAN2 (база): <input type="text" value="172.16.0.0/16"/></p> <p>Использовать протокол DHCP на интерфейсе 2: <input type="checkbox"/></p> <p>IP адрес DNS-сервера 2: <input type="text"/></p>	<p>Разрешить NTP: <input checked="" type="checkbox"/></p> <p>Привязка сервиса NTP к интерфейсам:</p> <p>+ X IP адрес 2 (LAN 2) ▾ +</p> <p>Внешние NTP серверы:</p> <p>+ X 10.0.0.239 prefer <input type="checkbox"/> iburst <input checked="" type="checkbox"/> + X 10.0.0.5 prefer <input checked="" type="checkbox"/> iburst <input checked="" type="checkbox"/> +</p> <p>LocalStratum: <input type="text"/></p> <p>Сервер NTP не используется (LocalStratum не задан). Запросы к серверам NTP передаются через LAN2. Определены два сервера NTP: 10.0.0.239 и 10.0.0.5 (основной).</p>
<p>LAN1: базовый адрес 10.0.0.0/8 LAN2: базовый адрес 10.0.0.0/8 Только клиент NTP, требуется посылать запросы к серверам 10.0.0.239 и 10.0.0.5 через интерфейс LAN2</p>	<p>Имя хоста: <input type="text"/></p> <p>Включить MAC адреса в конфигурацию: <input type="checkbox"/></p> <p>LAN 1</p> <p>Фактический MAC адрес: 00:08:b3:02:9d:e1 Фактический IP адрес LAN 1: 10.0.0.3/8 IP адрес LAN1 (база): <input type="text" value="10.0.0.0/8"/></p> <p>Использовать протокол DHCP на интерфейсе 1: <input type="checkbox"/></p> <p>IP адрес DNS-сервера 1: <input type="text"/></p> <p>LAN 2</p> <p>Фактический MAC адрес: 00:08:b3:02:9d:e2 Фактический IP адрес LAN 2: 10.0.0.4/8 IP адрес LAN2 (база): <input type="text" value="10.0.0.0/8"/></p> <p>Использовать протокол DHCP на интерфейсе 2: <input type="checkbox"/></p> <p>IP адрес DNS-сервера 2: <input type="text"/></p>	<p>Разрешить NTP: <input checked="" type="checkbox"/></p> <p>Привязка сервиса NTP к интерфейсам:</p> <p>+ X IP адрес 2 (LAN 2) ▾ +</p> <p>Внешние NTP серверы:</p> <p>+ X 10.0.0.239 prefer <input type="checkbox"/> iburst <input checked="" type="checkbox"/> + X 10.0.0.5 prefer <input checked="" type="checkbox"/> iburst <input checked="" type="checkbox"/> +</p> <p>LocalStratum: <input type="text"/></p> <p>Сервер NTP не используется (LocalStratum не задан). Запросы к серверам NTP передаются через LAN2. Определены два сервера NTP: 10.0.0.239 и 10.0.0.5 (основной).</p>

Обратите внимание, что сервис NTP версии 3, имеющийся в некоторых операционных системах Windows, несовместим с сервисом NTP контроллера и практически любых других сетевых устройств с поддержкой протокола NTP.

4.6.5.3. Настройка сервера NTP

При активизации сервис NTP контроллера автоматически запускается в качестве клиента и сервера. При этом, если параметр **LocalStratum** (*стратум*), представляющий качество системных часов контроллера или степень доверия к системным часам, не установлен или равен 15, то ни один клиент NTP не будет синхронизировать свое системное время с системными часами данного контроллера.

Если клиент NTP данного контроллера установил соединение с удаленным сервером NTP, имеющим стратум от 14 и ниже, то стратум часов данного контроллера автоматически примет значение, на единицу большее стратума часов сервера. Таким образом, если стратум удаленного сервера равен 13 и ниже, то стратум сервера данного контроллера примет на 1 большее значение (от 14 и ниже), и все клиенты с меньшим стратумом, для которых в качестве источника синхронизации времени выбран сервер данного контроллера, будут синхронизировать свое время с временем данного контроллера.

Если данный контроллер предполагается использовать в сети в качестве единственного сервера NTP, следует установить его **LocalStratum** равным не менее 14.

Список **Привязка сервиса NTP к интерфейсам** служит для соотнесения сервера NTP с сетевыми интерфейсами контроллера. Если список оставлен пустым, то сервер будет обслуживать запросы клиентов NTP на всех доступных сетевых интерфейсах контроллера. Если требуется установить сервер на определенный сетевой интерфейс, с помощью кнопки  добавьте элемент списка привязки и выберите требуемый сетевой интерфейс.

По завершении настройки сервиса NTP нажмите **Применить конфигурацию** на главной странице веб-конфигуратора.

Как указывалось ранее в п. 4.6.5.2, длительность синхронизации зависит от стабильности передачи сообщений между клиентами и сервером, и составляет от 10 до 20 минут.

4.6.6. Сервис протокола RTP

Системное программное обеспечение контроллера СРМ810-03 содержит сервис протокола синхронизации времени RTP IEEE 1588 v2 (далее RTPv2), поддерживающий функции мастера и подчиненного узла RTPv2 и предназначенный для синхронизации системного времени контроллера с системным временем на другом узле сети, выполняющем функцию мастера RTPv2, а также для синхронизации системного времени других узлов сети, поддерживающих функции подчиненного узла RTPv2.

Активизация сервиса выполняется в веб-конфигураторе на странице **Параметры RTP** путем отметки опции **Разрешить RTP**, показанной на рисунке 127.

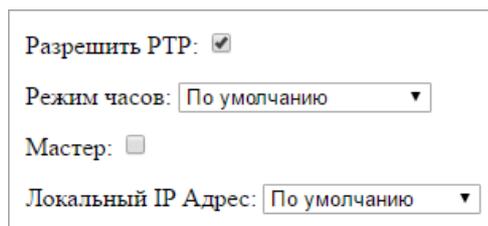


Рисунок 127 – Параметры сервиса RTPv2 в веб-конфигураторе

Сервис RTPv2 может быть соотнесен с IP-адресом одного сетевого интерфейса контроллера путем выбора соответствующего значения параметра **Локальный IP Адрес**. При выборе значения *По умолчанию* будет использоваться IP-адрес сетевого интерфейса LAN1.

При выборе опции **Мастер** становится доступным параметр **Класс часов**, показанный на рисунке 128 и определяющий качество (точность) системных часов данного мастера, а также возможность одновременной работы сервиса RTPv2 в режиме подчиненного узла:

1. Если опция **Мастер** не отмечена, то сервис функционирует только в качестве подчиненного узла RTPv2 с классом часов 255.
2. Если опция **Мастер** отмечена, и установлено значение **Класс часов** от 128 до 254, то сервис будет функционировать в качестве мастера и подчиненного узла RTPv2. При появлении в сети другого мастера RTPv2, у которого значение класса часов меньше, чем у данного, сервис RTPv2 контроллера становится подчиненным RTPv2, синхронизируя свои системные часы с часами появившегося мастера.

3. Если опция **Мастер** отмечена, и установлено значение **Класс часов** менее 127, то сервис будет функционировать только в качестве мастера RTRv2. При появлении в сети другого мастера RTRv2, у которого значение класса часов меньше, чем у данного, сервис RTRv2 контроллера перестает выполнять функцию мастера, но при этом не синхронизирует свои системные часы с часами появившихся мастеров.

Разрешить RTR:

Режим часов: По умолчанию ▼

Мастер:

Локальный IP Адрес: IP адрес 1 (LAN 1) ▼

Класс часов:

Рисунок 128 – Настройка сервиса RTRv2 для работы в качестве мастера

Параметр **Режим часов** определяет тип протокола взаимодействия между мастером и подчиненным узлом RTRv2:

1. *End-to-End* (совпадает со значением *По умолчанию*) – предназначен для работы в сетях с различной топологией, в которых на пути между мастером и подчиненными узлами протокола RTRv2 присутствуют узлы сети и коммуникационное оборудование, которые не поддерживают протокол RTRv2. В этом случае текущий активный мастер RTRv2 передает в сеть групповые запросы протокола *Announce*, *Sync*, *Follow_Up* и отвечает сообщением *Delay_Resp* на групповые запросы *Delay_Req* от других подчиненных узлов RTRv2 с большим значением класса часов.
2. *Peer-to-Peer* – предназначен для работы в сетях с различной топологией, в которых на пути между мастером и подчиненными узлами RTRv2 присутствуют только узлы, поддерживающие протокол RTRv2. В этом случае текущий активный мастер RTRv2, помимо групповых запросов *Announce*, *Sync*, *Follow_Up*, обменивается с другими узлами RTRv2 групповыми сообщениями с префиксом *Path_*, предназначенными для дополнительной компенсации задержек передачи в сегментах сети между отдельными промежуточными узлами на пути между мастером и подчиненными узлами. При использовании протокола RTRv2 в локальных сетях в подавляющем большинстве случаев не требуется использовать режим часов *Peer-to-Peer*.
3. *Без запроса задержки* – данный режим предназначен для синхронизации времени только с использованием групповых запросов мастера *Announce*, *Sync*, *Follow_Up* без запросов *Delay_Req* от каждого подчиненного узла. Таким образом обеспечивается возможность значительного сокращения сетевого трафика, связанного с синхронизацией времени по протоколу RTRv2, и при этом качество синхронизации времени не ухудшается, если все узлы находятся в пределах одного-двух физических сегментов сети.

По завершении настройки сервиса RTR нажмите **Применить конфигурацию** на главной странице веб-конфигуратора.

Обратите внимание, что при расхождении системного времени между мастером и подчиненным узлом на величину, превышающую 1 секунду, установка времени мастера на всех подчиненных узлах производится мгновенно. При расхождении часов мастера и подчиненных узлов на величину менее 1 секунды, происходит плавное приближение времени подчиненных узлов к времени мастера с использованием пропорционально-интегрального регулятора. Это позволяет избежать частых коррекций системных часов и связанных с этим накладных расходов.

Если приложению, запускаемому на контроллере, при включении питания требуется гарантированно синхронизировать системное время с одним из доступных в сети мастеров RTRv2, достаточно при обработке системного события *LegacyOnInit* вызвать функцию *SysTimeRtcSet* из библиотеки *SysTime*, передав ей нулевое значение в качестве параметра.

Если в качестве одного из мастеров RTR предполагается использовать управляемые коммутаторы МОХА, предварительно убедитесь, что выбранный коммутатор действительно поддерживает протокол RTRv2 и, главное, передает актуальное время, а не содержимое внутренних часов от начала эпохи (1970). После этого установите в конфигурации коммутатора режим часов (**Clock Mode**) v2 *E2E* (End-to-End)

или v2 P2P (Peer-to-Peer), тип временной шкалы (**Timescale type**) *ARB*, и разрешите RTP на портах, к которым подключены СРМ810-03. При необходимости синхронизировать время на коммутаторах МОХА со стороны одного или более мастеров RTP на контроллерах СРМ810-03, в конфигурации RTP контроллеров должно быть установлено значение класса часов, меньшее 248, которое обычно установлено по умолчанию для параметра **Clock Class** в конфигурации МОХА.

4.6.7. Сервис протокола FTP

Сервис протокола FTP в составе системного программного обеспечения контроллера предназначен для передачи файлов между контроллером и ПК по сети Ethernet при помощи приложений, поддерживающих функцию клиента FTP.

По умолчанию сервис FTP установлен на все сетевые интерфейсы контроллера.

Начиная с версии 3.4.5.0 системного программного обеспечения контроллера, в веб-конфигураторе имеется страница **Параметры FTP**, показанная на рисунке 129, которая обеспечивает возможность отключения сервиса FTP, включения режима FTP поверх SSL (FTPS), включения сервиса защищенного протокола SFTP, выбора сетевых интерфейсов, на которые должны быть установлены сервисы, и выбора номеров TCP-портов, которые прослушивают сервисы FTP и FTPS.

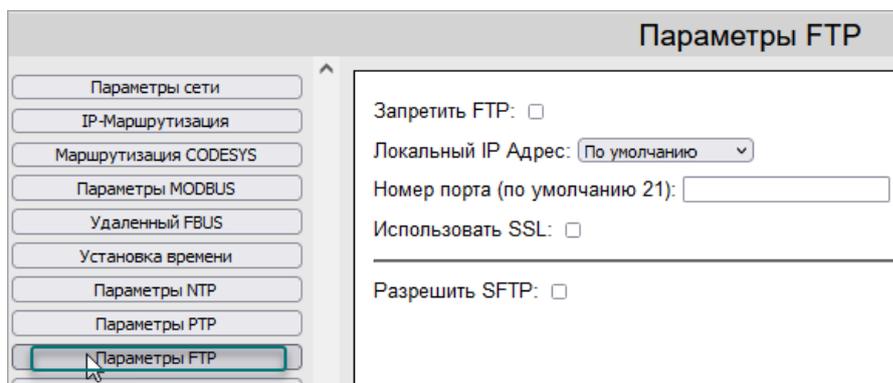


Рисунок 129 – Страница настройки параметров FTP

Флажок **Запретить FTP** предназначен для отключения или включения сервиса FTP на интерфейсе или двух интерфейсах, выбранных в поле **Локальный IP Адрес**.

Флажок **Использовать SSL** предназначен для включения режима SSL и использования протокола FTPS сервисом FTP с запретом соединений по открытому (нешифрованному) FTP.

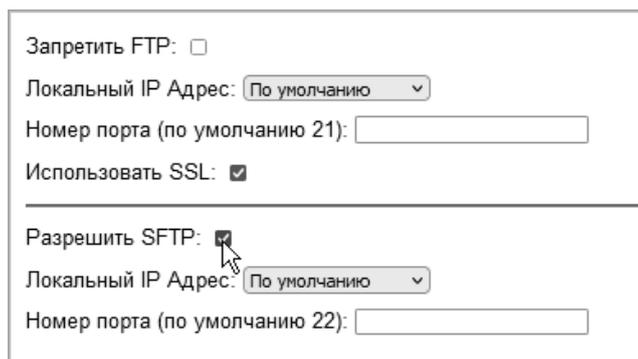


Рисунок 130 – Включение протокола SFTP

Флажок **Разрешить SFTP**, показанный на рисунке 130, позволяет дополнительно включить сервис протокола SFTP и отдельно установить номер TCP-порта, а также выбрать один или оба сетевых интерфейса, на которые должен устанавливаться сервис.

Для работы протоколов FTPS и SFTP используется самоподписанный сертификат TLS Web Server Authentication, который по умолчанию автоматически генерируется системным программным обеспечением контроллера со сроком действия 1 месяц. Более подробная информация о сертификатах приведена в п. 5.11.

Для подключения к контроллеру по протоколам SFTP, FTP или FTPS может использоваться одна из двух встроенных системных учетных записей пользователей контроллера:

Administrator – данная учетная запись обеспечивает полный доступ к корневому каталогу системы исполнения контроллера и, помимо выгрузки и загрузки пользовательских файлов, позволяет загружать в контроллер файл обновления системного программного обеспечения контроллера *norm.dnl*, файл обновления микропрограмм модулей ввода-вывода *ffw.dnl* и файл развертывания пользовательского приложения и системных параметров контроллера *norm.upl*.

Everyone – данная учетная запись дает доступ к каталогу пользователя */user* и к смонтированному в него подкаталогу */user/microsd1*, соответствующего установленной карте *microSD*, и другим подкаталогам, соответствующим подключенным к портам *USB* контроллера съемным дисковым накопителям. Каталог пользователя является корневым для функций библиотек *SysFile* и *SysDir*, вызываемых с использованием относительных путей к файлам и каталогам: т.е. все файлы и каталоги, создаваемые при помощи функций библиотеки с использованием относительного пути или без пути, создаются в каталоге */user*.

По умолчанию для учетной записи *Everyone* задан пустой пароль, а для учетной записи *Administrator* определен пароль *Administrator*. Более подробная информация об управлении доступом приведена в руководстве пользователя *Fastwel PLC Application Toolkit*.

Указания по передаче файлов между ПК и контроллером приведены в руководстве пользователя *Fastwel PLC Application Toolkit*.

5. Настройка системных параметров

5.1. Общие сведения

Настройка параметров системных сервисов контроллера выполняется при помощи веб-браузера, запускаемого на ПК. Для настройки системных параметров контроллеров через имеющийся в составе СПО контроллеров встроенный веб-сервер на ПК должен быть установлен хотя бы один из следующих веб-браузеров:

- Mozilla Firefox® версии не ниже 52;
- Google Chrome® версии не ниже 56;
- Opera® версии не ниже 44;
- Microsoft Edge® версии не ниже 44.

Актуальная информация о совместимых браузерах приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Для входа на веб-сервер контроллера в адресной строке браузера следует ввести IP-адрес сетевого интерфейса контроллера, подключенного к той же подсети, что и ПК. Например:

`http://10.0.0.100`

По умолчанию веб-сервер контроллера устанавливается на IP-порт 80 всех доступных сетевых интерфейсов, поэтому в адресе не требуется указывать номер IP-порта.

Рекомендации по проверке доступности контроллера по сети с некоторого ПК приведены в руководстве пользователя Fastwel PLC Application Toolkit. Указания по выяснению IP-адресов сетевых интерфейсов контроллера приведены в п. 3.2.3.

При успешном соединении браузера с веб-сервером в окне браузера будет отображена начальная страница веб-сервера, показанная на рисунке 131, с предложением ввести имя пользователя и пароль.

Рисунок 131 – Начальная страница встроенного веб-сервера

В поле **Username** введите имя пользователя *Administrator*, а в поле **Password** – пароль, а затем нажмите кнопку **Sign in**. Если пароль для учетной записи *Administrator* ранее не изменялся, то по умолчанию используется пароль *Administrator*. Более подробная информация об управлении доступом к контроллеру приведена в руководстве пользователя Fastwel PLC Application Toolkit.

При вводе неправильного имени пользователя и/или пароля, над полем **Username** будет выведена строка:

Login failed

При успешном входе на сервер в окне браузера будет отображена страница **Параметры сети** веб-конфигуратора контроллера, показанная на рисунке 132.

В левом верхнем углу страницы находится выпадающий список , позволяющий переключить язык пользовательского интерфейса веб-конфигуратора.

После изменения системных параметров требуется нажать кнопку **Применить конфигурацию**, что приведет к полному перезапуску контроллера. В связи с этим рекомендуется сначала выполнить настройку всех необходимых параметров, а затем применить конфигурацию.

Имеется возможность сохранения в файле на ПК значений параметров, которые еще не были актуализированы нажатием кнопки **Применить конфигурацию**. Для этого следует нажать **Сохранить конфигурацию в файле**. Впоследствии данный файл можно загрузить в веб-конфигуратор нажатием кнопки **Загрузить конфигурацию из файла**, а затем нажать **Применить конфигурацию**.

Для повторного чтения параметров системных сервисов контроллера следует нажать сочетание клавиш Ctrl-R.

The screenshot shows the 'Параметры сети' (Network Parameters) configuration page. On the left is a sidebar with menu items like 'Параметры сети', 'IP-Маршрутизация', 'Маршрутизация CODESYS', etc. The main area is divided into sections for 'LAN 1' and 'LAN 2'. Each section includes fields for 'Имя хоста', 'Включить MAC адреса в конфигурацию', 'Упрощенный вид', 'Фактический MAC адрес', 'Фактический IP адрес LAN', 'IP адрес LAN (база)', 'Использовать протокол DHCP на интерфейсе', and 'IP адрес DNS-сервера'. The right sidebar contains a summary of the IP addresses and masks for both interfaces.

Рисунок 132 – Страница "Параметры сети" веб-конфигуратора CPM810-03

5.2. Параметры сети

Страница **Параметры сети (Network parameters)**, показанная на рисунке 132, предназначена для настройки IP-параметров сетевых интерфейсов контроллера и адресов DNS-серверов для каждого интерфейса. Информация о способах настройки IP-параметров приведена в пп. 4.6.2.2 – 4.6.2.3.

Краткая информация о назначении параметров сети приведена в таблице 6.

Таблица 6 – Описание параметров и полей страницы Параметры сети

Параметр/поле	Описание
Имя хоста (Host name)	<p>Определяет имя контроллера в сети, отображаемое в окне сканирования среды разработки и используемое другими сетевыми сервисами. Имя хоста должно состоять из символов латинского алфавита, цифр от 0 до 9, символов подчеркивания или короткого тире, и иметь длину не более 48 символов.</p> <p>Если параметр не установлен, то имя контроллера сети формируется автоматически следующим образом: <i>CPMXXX-ZZ-YYMMXXXNNNN</i>, где: <i>XXX</i> – числовой суффикс модели контроллера, например, <i>CPM810</i>; <i>ZZ</i> – числовой суффикс исполнения, например, <i>CPM810-02</i>; <i>YYMM</i> – соответствует году и месяцу производства контроллера, если он выпущен до 2021 г., начиная с 2021 г., часть имени соответствует году и кварталу производства контроллера, например, <i>CPM810-03-23Q18100061</i> соответствует 1-му кварталу 2023 г.; <i>xxxNNNN</i> – часть серийного номера, например, <i>CPM810-03-23018100061</i>.</p>
Включить MAC адреса в конфигурацию (Include MAC addresses into configuration)	<p>ВНИМАНИЕ! Не используйте данную опцию, если нет абсолютной уверенности в необходимости ее использования.</p> <p>Предназначено для ввода MAC-адресов интерфейсов LAN1/2 вручную.</p>
Упрощенный вид (Simplified view)	Параметр предназначен для переключения отображения IP-адресов из формата CIDR (см. п. 1.4) в формат <i>адрес:маска подсети</i> .
Фактический MAC адрес (Actual MAC address)	Содержит текущее значение MAC-адреса сетевого интерфейса.
Фактический IP адрес LAN 1/2 (Actual IP address 1/2)	Содержит текущее значение IP-адреса сетевого интерфейса.
IP адрес LAN1/2 (база) (IP Address LAN1/2 (base))	Базовый IP-адрес сетевого интерфейса в CIDR-представлении при статическом назначении. Информация о статическом назначении IP-адресов приведена в п. 4.6.2.2.
Использовать протокол DHCP на интерфейсе 1/2	Данная опция предназначена для выбора динамического назначения IP-адреса для выбранного сетевого интерфейса.

Параметр/поле	Описание
<i>(Use DHCP on interface 1/2)</i>	Информация о динамическом назначении IP-адресов приведена в п. 4.6.2.3.
<i>IP адрес DNS-сервера 1/2 (IP address of DNS 1/2)</i>	IP-адрес DNS-сервера (сервера доменных имен). Данный параметр требуется устанавливать в случае, если приложение контроллера использует функции библиотеки SysSocket и устанавливает соединение с удаленными узлами по имени.

5.3. IP-Маршрутизация

Данная страница предназначена для добавления записей в таблицы IP-маршрутизации контроллера. Информация о настройке IP-маршрутизации приведена в п. 4.6.2.4.

5.4. Маршрутизация CODESYS

Данная страница содержит опцию **Разрешить параллельную маршрутизацию**, которая предназначена для сетевого поиска и взаимодействия по протоколу IDE МЭК 61131-3 с контроллерами, подключенными к данному контроллеру через одну или несколько подсетей.

Информация об использовании параллельной маршрутизации приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Рисунок 133 – Страница "Маршрутизация CODESYS"

На данной странице можно включить или выключить поддержку взаимодействия контроллера с IDE МЭК 61131-3 через последовательный порт COM1 или COM2 и выбрать скорость обмена, как показано на рисунке 133. В виду больших объемов информации, передаваемой между IDE МЭК 61131-3 и контроллером при загрузке приложений, для связи среды разработки с контроллером рекомендуется использовать интерфейсы сети Ethernet.

5.5. Параметры MODBUS

5.5.1. Общие сведения

Системное программное обеспечения контроллера обеспечивает возможность функционирования до двух экземпляров сервера и до двух экземпляров мастера (клиента) протокола MODBUS TCP. Количество одновременно запущенных экземпляров сервера и мастера определяется конфигурацией приложения, загруженного в контроллер из IDE МЭК 61131-3.

При разработке приложения каждому экземпляру сервера и мастера пользователем должен быть присвоен уникальный числовой идентификатор, предназначенный для *соотнесения* экземпляра сервера или мастера с IP-адресом сетевого интерфейса контроллера. Информация об установке идентификаторов сервера и мастера MODBUS TCP приведена в п. 4.6.3.2 и 4.6.4.2 соответственно.

Применительно к серверу термин *соотнесение идентификатора с IP-адресом* означает, что при запуске приложения экземпляр сервера с некоторым числовым идентификатором устанавливается на IP-адрес, поставленный в соответствие данному идентификатору, и обслуживает входящие запросы от клиентов MODBUS TCP.

Применительно к мастеру термин *соотнесение идентификатора с IP-адресом* означает, что при запуске приложения экземпляр клиента с некоторым числовым идентификатором передает запросы к

серверам MODBUS TCP через сетевой интерфейс с IP-адресом, поставленным в соответствие данному идентификатору.

Страница **Параметры MODBUS** содержит два списка, **Серверы MODBUS** и **Мастера MODBUS**, предназначенные для настройки соотношений экземпляров сервера и клиента MODBUS TCP с IP-адресами сетевых интерфейсов. По умолчанию оба списка не содержат ни одной записи, в результате чего используются следующие неявные правила соотношения экземпляров сервера и клиента:

1. В системе функционирует единственный экземпляр сервера MODBUS TCP, который устанавливается на IP-адреса всех доступных сетевых интерфейсов контроллера с использованием TCP-порта 502.
2. В системе может функционировать один или два экземпляра мастера MODBUS TCP, при этом каждый запрос мастера некоторому удаленному серверу передается через сетевой интерфейс, соответствующий первой найденной записи в таблице IP-маршрутизации для IP-адреса получателя запроса.

Таким образом, запросы двух мастеров MODBUS TCP на данном контроллере к серверам в одной или двух подсетях передаются через соответствующие сетевые интерфейсы.

При назначении IP-параметров в одной подсети запросы мастеров MODBUS TCP к серверам передаются только через интерфейс LAN1.

5.5.2. Настройка параметров серверов MODBUS TCP

Настройка параметров серверов MODBUS TCP на странице **Параметры MODBUS** требуется в следующих ситуациях:

1. В конфигурации приложения определены один или два экземпляра сервера MODBUS TCP и требуется установить каждый экземпляр на определенный сетевой интерфейс.
2. В конфигурации приложения определены два экземпляра сервера MODBUS TCP, и требуется установить оба экземпляра на один и тот же сетевой интерфейс.
3. Независимо от количества экземпляров сервера MODBUS TCP (один или два) при доступе к серверам требуется использовать TCP-порт, отличный от установленного по умолчанию 502.

Для добавления записи соотношения экземпляра сервера нажмите кнопку под названием списка **Серверы MODBUS**, как показано на рисунке 134, в поле **Идентификатор (как в проекте CODESYS)** введите числовой идентификатор экземпляра сервера, присвоенный данному экземпляру в конфигурации приложения, в выпадающем списке **Локальный IP** выберите IP-адрес требуемого интерфейса (*IP адрес 1 (LAN1)* или *IP адрес 2 (LAN2)*) и, при необходимости, измените номер TCP-порта в поле **Порт**.

При необходимости настройте соотношение для второго экземпляра сервера, добавив еще одну запись в список.

Обратите внимание, что при установке двух экземпляров сервера на один и тот же интерфейс для разных экземпляров необходимо использовать разные номера TCP-портов, задаваемые в поле **Порт**.

Для удаления записи из списка нажмите кнопку , расположенную справа от кнопки соответствующей записи.

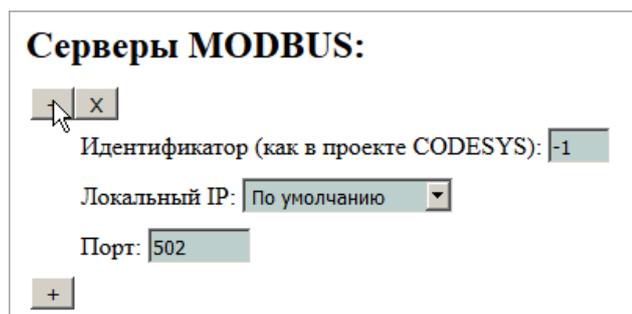


Рисунок 134 – Добавление записи соотношения идентификатора сервера с IP-адресом сетевого интерфейса

Примеры вариантов соотношения экземпляров сервера MODBUS TCP приведены в таблице 2 п. 4.6.3.2.

По завершении настройки соотношений экземпляров сервера MODBUS TCP нажмите кнопку **Применить конфигурацию**.

5.5.3. Настройка параметров мастеров MODBUS TCP

Настройка параметров мастеров MODBUS TCP на странице **Параметры MODBUS** требуется в следующих ситуациях:

1. В конфигурации приложения определены один или два экземпляра мастера MODBUS TCP, и требуется установить каждый экземпляр на отдельный (собственный) сетевой интерфейс.
2. В конфигурации приложения определены два экземпляра мастера MODBUS TCP, и требуется установить оба экземпляра на один и тот же сетевой интерфейс.

Для добавления записи соотношения экземпляра мастера нажмите кнопку **+** под названием списка **Мастера MODBUS**, в поле **Идентификатор (как в проекте CODESYS)** введите числовой идентификатор, присвоенный данному экземпляру в конфигурации приложения, и в выпадающем списке **Локальный IP** выберите IP-адрес требуемого интерфейса (*IP адрес 1 (LAN1)* или *IP адрес 2 (LAN2)*), как показано на рисунке 135.

При необходимости настройте соотношение для второго экземпляра мастера, добавив еще одну запись в список.

Для удаления записи из списка нажмите кнопку **X**, расположенную справа от кнопки **+** соответствующей записи.

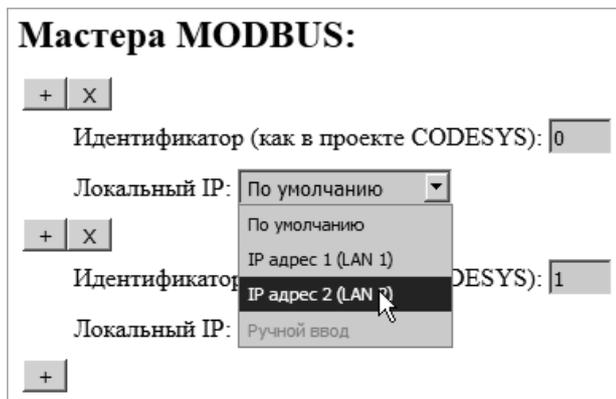


Рисунок 135 – Записи в списке соотношений мастеров MODBUS TCP

Примеры вариантов соотношения экземпляров мастера MODBUS TCP приведены в таблице 4 п. 4.6.4.2.

По завершении настройки соотношений нажмите кнопку **Применить конфигурацию**.

5.6. Удаленный FBUS

Страница **Удаленный FBUS** предназначена для соотношения удаленных адаптеров шины FBUS (NIM745-01) с IP-адресами сетевых интерфейсов контроллера.

Применительно к удаленному адаптеру шины FBUS термин *соотношение порта (10 или 11) с IP-адресом* означает, что при запуске приложения сервис ввода-вывода будет передавать запросы к модулям ввода-вывода, определенным в конфигурации порта шины FBUS идентификатором 10 или 11, через сетевой интерфейс с IP-адресом, соотношенным с данным идентификатором.

Подробная информация о способах соотношения удаленных адаптеров FBUS приведена в п. 4.5.2.2 настоящего документа.

5.7. Установка времени

Страница **Установка времени**, показанная на рисунке 136, позволяет прочитать и установить системное время контроллера.

Для получения текущего времени контроллера нажмите кнопку **Запросить время из контроллера**. Обратите внимание, что в полях **Время контроллера:Часовой пояс браузера** и **Время контроллера:UTC** всегда отображаются прогнозируемые значения времени контроллера в часовом поясе браузера и в формате UTC (Всемирное координированное время). Прогнозируемое значение времени образуется путем сложения значений текущего времени браузера и положительной или отрицательной разницы между временем браузера и временем контроллера. Т.е. после выполнения запроса времени из контроллера разница вычисляется один раз, после чего показания часов в полях **Время контроллера** в окне браузера обновляются самим браузером путем сложения полученной разницы с временем на ПК, где запущен браузер.

	Часовой пояс браузера	UTC
Время браузера	2017-06-29 17:33:40	2017-06-29 14:33:40
Время контроллера	2017-06-29 17:34:39	2017-06-29 14:34:39
Время браузера = Время контроллера + <input type="text" value="0"/> : <input type="text" value="0"/> : <input type="text" value="59.028"/>		
<input type="button" value="Обнулить разницу"/>		
<input type="button" value="Запросить время из контроллера"/> <input type="button" value="Установить время в контроллере"/>		

Рисунок 136 – Страница Установка времени веб-конфигуратора

Установка времени в контроллере выполняется нажатием кнопки **Установить время в контроллере**, при этом в контроллер передается время UTC ПК, на котором запущен браузер. Перед установкой времени в контроллере для минимизации расхождения между временем UTC ПК, на котором запущен браузер, и устанавливаемым временем UTC контроллера следует нажать кнопку **Обнулить разницу**.

Получение текущего и передача нового значения времени в контроллер по нажатию соответствующих кнопок выполняются с многократным обратным чтением для минимизации влияния задержек передачи запросов по сети.

Часовые пояса в браузере и на контроллере могут отличаться, в результате чего реальное установленное время в контроллере будет отличаться от времени в часовом поясе браузера.

5.8. Параметры NTP

Страница **Параметры NTP** предназначена для настройки параметров сервиса протокола NTP v4, который входит в состав системного программного обеспечения контроллера и служит для синхронизации времени по сети. Информация о сервисе NTP и о способах настройки его параметров приведена в п. 4.6.5 настоящего документа.

5.9. Параметры RTP

Страница **Параметры RTP** предназначена для настройки параметров сервиса протокола синхронизации времени RTP IEEE 1588 v2, входящего в состав системного программного обеспечения контроллера. Информация о сервисе RTP контроллера и настройке его параметров приведена в п. 4.6.6 настоящего документа.

5.10. Параметры OPC UA

Страница **Параметры OPC UA**, показанная на рисунке 137, доступна в системном программном обеспечении контроллера, начиная с версии 3.0.x.x, и предназначена для активации и настройки параметров встроенного сервера OPC Unified Architecture.

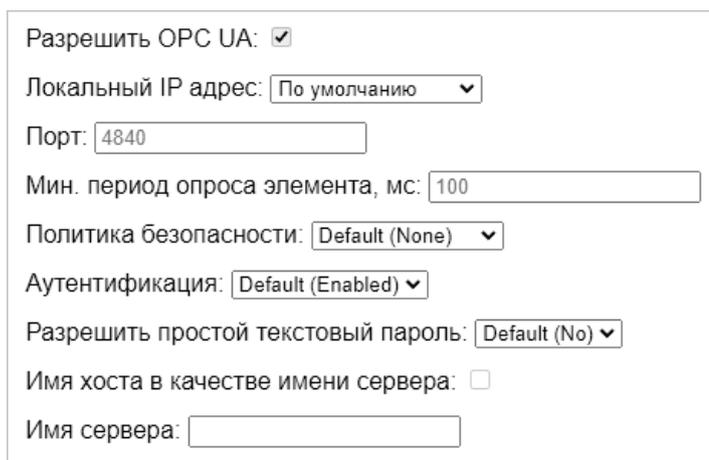


Рисунок 137 – Страница настройки параметров сервера OPC UA

Опция **Разрешить OPC UA** активирует сервер OPC UA в системе исполнения приложений контроллера. По умолчанию данная опция выключена.

Параметр **Порт** определяет порт TCP, используемый сервером для ожидания входящих соединений клиентов OPC UA. По умолчанию используется стандартный порт сервера OPC UA 4840.

Параметр **Мин. период опроса элемента** предназначен для установки минимального возможного значения периода опроса (в мс) значений переменных, публикуемых сервером OPC UA. Данный параметр соответствует атрибуту *MinimumSamplingInterval* согласно OPC Unified Architecture. Specification. Part 4: Services и означает, что значение уточненного сервером периода опроса значений элементов мониторинга (переменных) *revisedSamplingInterval*, передаваемого сервером клиенту при добавлении элементов мониторинга в подписку, будет не менее *MinimumSamplingInterval*.

Опция **Имя хоста в качестве имени сервера** позволяет выбрать имя хоста, назначенное контроллеру, в качестве имени сервера OPC UA, отображаемого первым при обзоре адресного пространства сервера и являющегося первым элементом символьного идентификатора путей к элементам данных сервера. Например, если на странице **Параметры сети** веб-конфигуратора в поле **Имя хоста** установлено имя *CPM810-03-NODE200*, а на странице **Параметры OPC UA** отмечен флажок **Имя хоста в качестве имени сервера**, то при подключении к КП клиентским приложением Unified Automation UaExpert имя хоста будет первым элементом просмотра адресного пространства сервера, как показано на рисунке 138.

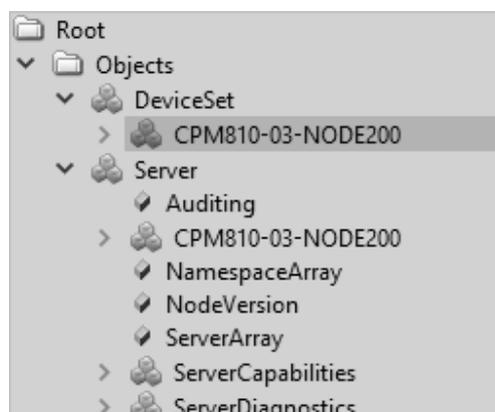


Рисунок 138 – Имя хоста в качестве имени сервера OPC UA

Поле **Имя сервера** (начиная с версии 3.3.4.0 СПО контроллера) позволяет задать произвольное имя сервера OPC UA в виде строки длиной не более 127 символов латинского алфавита, цифр и символов подчеркивания.

Параметр **Политика безопасности** определяет способ взаимодействия клиентов OPC UA с сервером контроллера с точки зрения защищенности обмена данными. Значение данного параметра по

умолчанию *None* устанавливает открытое (незащищенное) взаимодействие с использованием бинарного протокола OPC UA.

Значение *Basic256Sha256* предписывает серверу участвовать в обмене данными с клиентами только через защищенный канал с использованием цифровой подписи и шифрования каждого сообщения между клиентом и сервером.

При выборе значения *Basic256Sha256* для параметра **Политика безопасности** становится доступным для установки параметр **Режим безопасности сообщений**, который может принимать значения *Sign* или *Sign & Encrypt*.

При установке режима *Sign* у клиентов OPC UA будет возможность создания защищенных сеансов связи с сервером с использованием либо только цифровой подписи каждого сообщения (*Sign*), либо цифровой подписи и шифрования (*Sign & Encrypt*).

Режим *Sign & Encrypt* позволяет клиенту использовать только режим обмена сообщениями с цифровой подписью и шифрованием.

Более подробная информация о безопасном взаимодействии клиентов OPC UA с сервером приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Значение *Only Plain Text* параметра **Режим безопасности сообщений** обеспечивает возможность использования незащищенного взаимодействия клиентов OPC UA с сервером независимо от значения параметра **Политика безопасности**.

Значение *All* параметра **Режим безопасности сообщений** совпадает с устанавливаемым по умолчанию *Default* и позволяет использовать как незащищенное, так и защищенное взаимодействие клиентов OPC UA с сервером либо только с цифровой подписью сообщений (*Sign*), либо с цифровой подписью и с шифрованием (*Sign & Encrypt*).

При необходимости организации защищенного взаимодействия клиентов OPC UA с сервером контроллера следует установить параметр **Политика безопасности:Basic256Sha256** и **Режим безопасности сообщений: Sign** или *Sign & Encrypt*.

Параметр **Аутентификация** позволяет включить механизм аутентификации пользователя при создании сеанса связи клиента с сервером путем передачи от клиента серверу имени пользователя и пароля. Значение по умолчанию *Enabled* делает аутентификацию необязательной, но возможной, *Disabled* запрещает аутентификацию, а *Enforced* разрешает создавать сеансы связи клиентов с сервером только после успешной передачи имени пользователя и пароля.

По умолчанию соединение с сервером OPC UA разрешено устанавливать с именами и паролями учетных записей, относящихся к группе *Administrators*, в том числе системных учетных записей *Administrator* и *Everyone*, однако, поскольку учетная запись *Everyone* по умолчанию имеет пустой пароль, она не может использоваться для создания соединений клиентов с сервером OPC UA с включенной аутентификацией. Более подробная информация о правах и учетных записях пользователей приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Параметр **Разрешить простой текстовый пароль** с установленным значением *Yes*, позволяет использовать аутентификацию при создании сеансов связи между клиентами OPC UA и сервером без шифрования пароля учетной записи пользователя, передаваемого в процессе создания сеанса связи клиента с сервером.

Если данный параметр имеет значение *No*, то пароль учетной записи пользователя при передаче от клиента серверу будет шифроваться с использованием алгоритма SHA256 RSA-OAEP. Чтобы иметь возможность шифрования пароля при аутентификации клиентов OPC UA, для программного компонента *CtrlOPCUAServer* системы исполнения контроллера должен быть установлен или сгенерирован действительный цифровой сертификат.

При использовании политики безопасности *Basic256Sha256* значение данного параметра должно быть *No*.

5.11. Сертификаты безопасности

Страница **Сертификаты безопасности** предназначена для управления цифровыми сертификатами подсистемы безопасности, используемой некоторыми программными компонентами системы исполнения контроллера. К операциям управления относятся: импорт файлов сертификатов,

полученных в соответствующих организациях, создание (генерация) собственных самоподписанных сертификатов, перенос из карантинной зоны в доверенную или недоверенную зону полученных по сети сертификатов клиентских приложений от других вычислительных устройств, пытающихся создать защищенный канал с соответствующей серверной частью на контроллере.

При открытии страницы **Сертификаты безопасности** отображается кнопка **Запросить** для запроса у подсистемы безопасности контроллера и отображения в виде списка информации о сертификатах из категории, выбранной в выпадающем списке слева:

1. *Все* – все сертификаты, зарегистрированные подсистемой безопасности контроллера, включая собственные и сертификаты приложений, полученные от приложений на других узлах сети, включая доверенные, недоверенные и находящиеся в карантинной зоне, т.е. сертификаты, надежность которых не подтверждена пользователем с функцией администратора подсистемы безопасности.
2. *Собственные* – сертификаты программных компонентов системного программного обеспечения контроллера, поддерживающих функции безопасности, включая создание защищенных каналов связи с шифрованием и цифровой подписью и зашифрованных приложений МЭК 61131-3.
3. *Доверенные* – сертификаты приложений, полученные от других узлов сети, с которыми допускается взаимодействие по сети. Сертификаты от приложений других узлов сети могут попасть в эту категорию только по инициативе пользователя, выполняющего функции администратора подсистемы безопасности контроллера.
4. *Карантин* – сертификаты приложений, полученные от других узлов сети, для которых пока не определена степень доверия. При первой попытке создания защищенного канала связи с контроллером происходит обмен цифровыми сертификатами, и сертификат приложения другого узла сети, участвующего в создании защищенного канала, сначала попадает в карантинную зону.
5. *Недоверенные* – сертификаты приложений, полученные от других узлов сети, с которыми не допускается взаимодействие по сети. Сертификаты от приложений других узлов сети попадают в эту категорию по инициативе пользователя, выполняющего функции администратора подсистемы безопасности контроллера.

При выборе категории *Собственные* в выпадающем списке справа от кнопки **Запросить** появится кнопки **Импорт**, а снизу – кнопку **Генерировать** и выпадающий список для выбора программных компонент системного программного обеспечения, для которых требуется сгенерировать самоподписанные сертификаты, как показано на рисунке 139:

1. *Для всех компонент* – генерация самоподписанных цифровых сертификатов по нажатию кнопки **Генерировать** будет выполнена для всех программных компонентов: *СтрOPCUAServer*, *СтрFastwelCore*, *СтрApp* и *СтрSecureChannel*. При нажатии **Генерировать** с выбранной опцией **Для всех компонент** будут удалены все имеющиеся собственные сертификаты, сгенерированные для данного сетевого имени контроллера (имени хоста), после чего будут созданы новые самоподписанные сертификаты для всех перечисленных далее программных компонентов системы исполнения.
2. *СтрFastwelCore* – начиная с версии 3.4.5.0 системного ПО контроллера, генерация самоподписанного сертификата для встроенных серверов FTPS, SFTP и веб-сервера в режиме HTTPS.
3. *СтрOPCUAServer* – генерация самоподписанного сертификата только для встроенного сервера OPC UA, предназначенного для создания защищенных сеансов связи с клиентскими приложениями на других узлах сети. Более подробная информация о защищенном взаимодействии клиентов OPC UA со встроенным сервером контроллера приведена в руководстве пользователя Fastwel PLC Application Toolkit.
4. *СтрApp* – генерация самоподписанного сертификата для шифрования приложения, загружаемого в контроллер. Более подробная информация о шифровании приложений приведена в руководстве пользователя Fastwel PLC Application Toolkit.
5. *СтрSecureChannel* – генерация самоподписанного сертификата для создания защищенного соединения между IDE МЭК 61131-3 и контроллером. Более подробная информация о защищенном взаимодействии между средой разработки и контроллером приведена в руководстве пользователя Fastwel PLC Application Toolkit.



Рисунок 139 – Дополнительные элементы управления на странице Сертификаты безопасности

Начиная с версии 3.4.5.0 системного ПО контроллера, подсистема безопасности контроллера по умолчанию содержит, как минимум, один цифровой сертификат для компонента *CmpFastwelCore* со сроком действия один месяц, который отображается при нажатии **Запросить** с выбранной категорией *Все* или *Собственные*, как показано на рисунке 140.

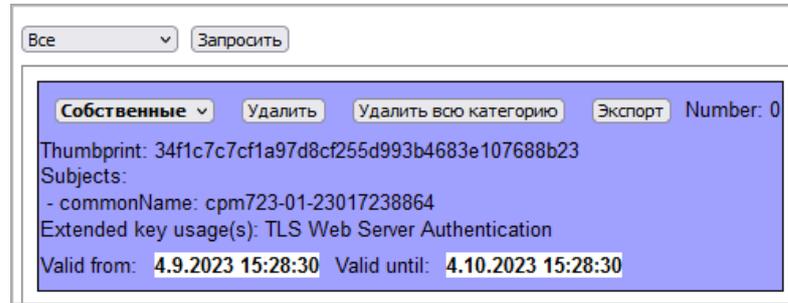


Рисунок 140 – Исходный сертификат компонента *CmpFastwelCore*

Данный сертификат генерируется автоматически, если при запуске системного программного обеспечения контроллера для текущей даты и времени по встроенным часам контроллера не имеется ни одного действительного сертификата для компонента *CmpFastwelCore*, срок действия которого включает в себя текущую дату и время.

Начиная с версии 3.4.5.0 системного ПО контроллера, информация о текущих используемых сертификатах отображаются в прямоугольных областях фиолетового цвета, а информация о просроченных сертификатах – в областях оранжевого цвета.

Для создания собственного самоподписанного цифрового сертификата для одного программного компонента с возможностью переноса на другой контроллер методом развертывания приложения:

1. На странице **Параметры сети** введите имя узла сети в поле **Имя хоста** и нажмите **Применить конфигурацию**. Задаваемое имя хоста должно состоять из символов латинского алфавита, цифр от 0 до 9, символов подчеркивания или короткого тире, и иметь длину не более 48 символов.
2. После перезапуска контроллера войдите веб-конфигуратором на страницу **Сертификаты безопасности**, выберите категорию **Собственные** в выпадающем списке слева от кнопки **Запросить**.
3. Выберите требуемый компонент в выпадающем списке справа от появившейся кнопки **Генерировать** и нажмите **Генерировать**.
4. В поле **действительных дней** на появившейся заставке введите требуемую длительность действия сертификата и нажмите кнопку **Начать генерацию самоподписанных сертификатов**. Через 8-15 секунд на заставке появится список с информацией о текущем состоянии сертификатов трех программных компонентов, показанный на рисунке 141.
5. При появлении над списком сообщения "*Затребованные сертификаты сгенерированы*. Нажмите кнопку **Заккрыть**" нажмите кнопку **Заккрыть**. Информация о сгенерированном сертификате будет отображена в списке сертификатов.

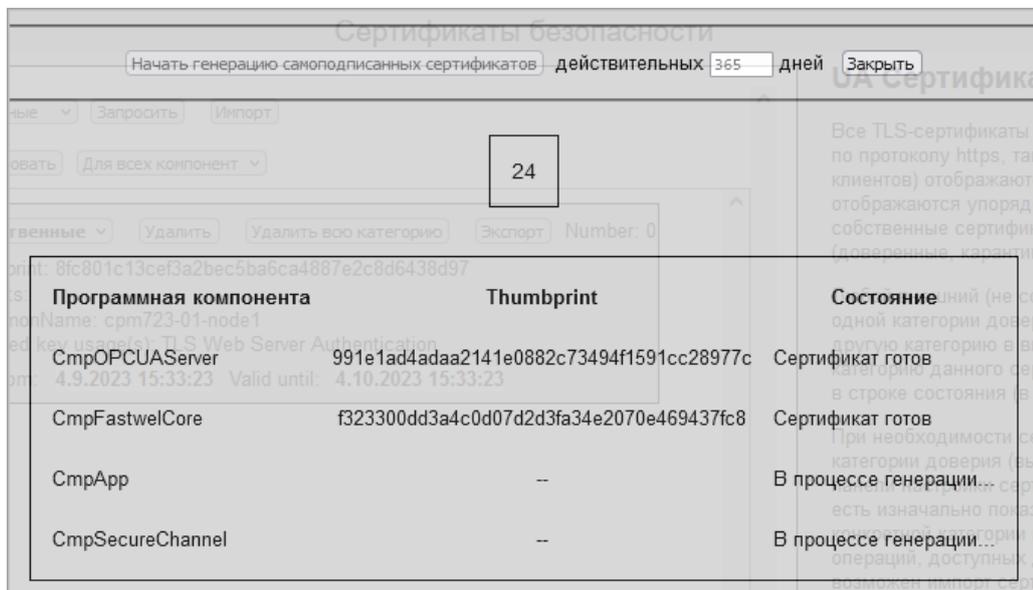


Рисунок 141 – Отображение процесса генерации сертификата для компонента *CmpSecureChannel*

При первой попытке установить защищенное соединение приложение на другом узле сети обменивается с контроллером цифровыми сертификатами, и полученный сертификат приложения помещается в карантин подсистемой безопасности контроллера. Информация о сертификатах в карантине отображается по нажатию кнопки **Запросить** в прямоугольных областях желтого цвета, как показано на рисунке 142.

Для перевода сертификата в число доверенных (в доверенную зону) выберите опцию **Доверенные** в выпадающем списке внутри прямоугольной области отображения информации о сертификате, как показано на рисунке 143. В случае успешного перевода сертификата в доверенную зону информация о сертификате будет отображена с прямоугольной области зеленого цвета.

Для перевода сертификата в число недоверенных (в недоверенную зону) выберите опцию **Недоверенные** в выпадающем списке внутри прямоугольной области отображения информации о сертификате. При успешном переводе сертификата в недоверенную зону информация о сертификате будет отображена с прямоугольной области красного цвета.

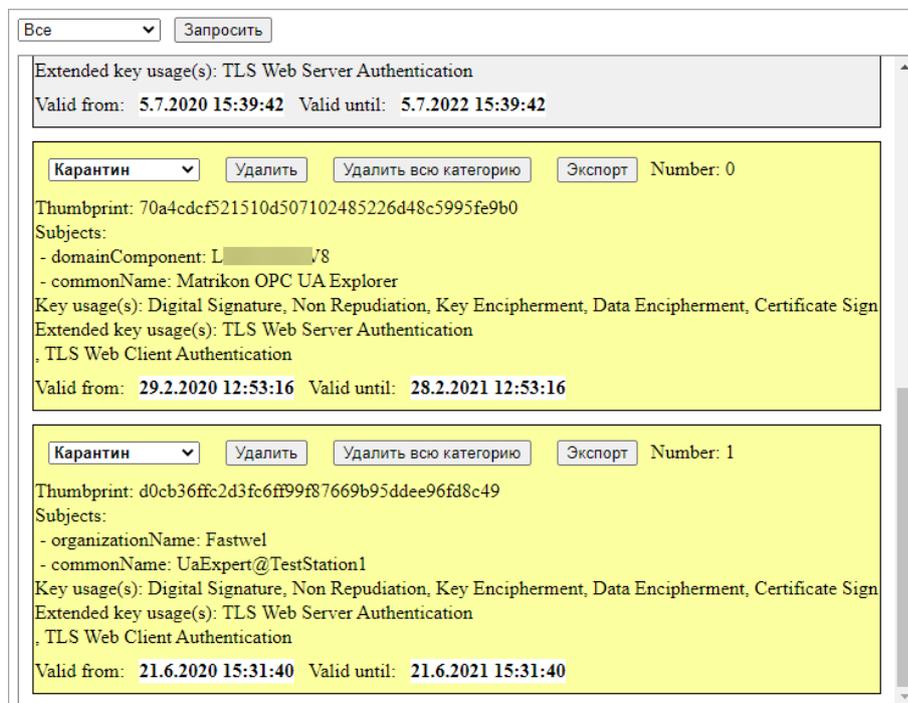


Рисунок 142 – Сертификаты приложений в карантине

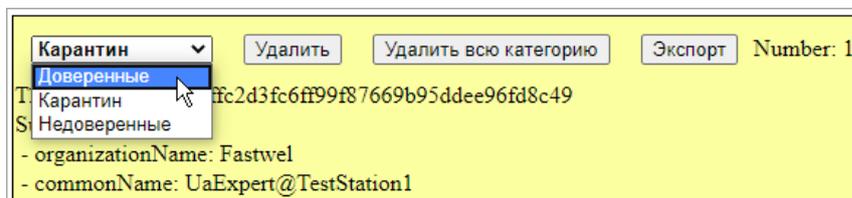


Рисунок 143 – Перевод сертификата в доверенные

Кнопка **Удалить** в области отображения информации о сертификате позволяет удалить соответствующий сертификат.

Кнопка **Удалить всю категорию** предназначена для удаления всех сертификатов категории, выбранной в выпадающем списке области отображения информации о сертификате.

Кнопка **Экспорт** позволяет выгрузить и сохранить данный сертификат в файле на компьютере.

5.12. Видео

Страница **Видео** предназначена для включения и настройки параметров видеоподсистемы контроллера.

Обратите внимание, что при включении видеоподсистемы, помимо видеоадаптера, в операционной системе контроллера запускаются соответствующие компоненты видеоподсистемы, а в системе исполнения приложений МЭК 61131-3 запускаются компоненты целевой визуализации. Таким образом, функционирование приложения IDE МЭК 61131-3 с подключенным компонентом *Target Visualization* возможно только после включения видеоподсистемы в системных параметрах контроллера.

При поставке по умолчанию видеоподсистема выключена, для включения следует отметить флажок **Разрешить видео**, как показано на рисунке 144, а затем нажать **Применить конфигурацию**.

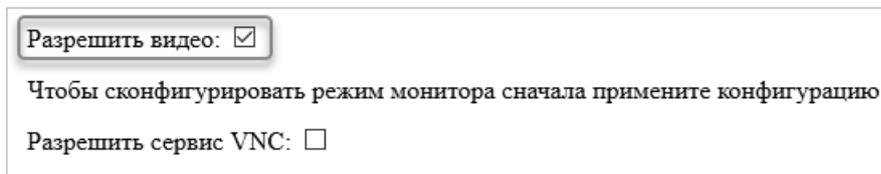


Рисунок 144 – Включение видеоподсистемы контроллера

После перезапуска следует повторно войти на страницу **Видео** веб-конфигуратора, установить требуемое разрешение экрана в выпадающем списке **Режим экрана** и нажать **Применить конфигурацию**.

Обратите внимание, что после перезапуска контроллера для подключенного к порту VGA видеомонитора будет установлено оптимальное разрешение, полученное от него при взаимодействии видеодрайвера с монитором.

Если в выпадающем списке **Режим экрана** установлено разрешение экрана с шириной и/или высотой менее оптимального, то изображение на экране будет пропорционально увеличено (растянуто). Например, если при оптимальном разрешении видеомонитора 1920×1080 установить параметр **Режим экрана**: 1024×768, то изображение на экране будет пропорционально растянуто по ширине в 1,875 раза и по высоте – примерно в 1,4 раза, в результате чего нижняя и правая части изображения могут оказаться вне видимой области. Чтобы это избежать этого дефекта, следует в свойствах элемента *TargetVisualization* в дереве проекта IDE МЭК 61131-3 выбрать опцию масштабирования **Изотропная** или **Анизотропная** с автоматическим определением размера, как показано на рисунке 145.

Если в выпадающем списке **Режим экрана** установлено разрешение экрана, любой из параметров которого (ширина и/или высота) более оптимального, то правая и/или левая часть изображения на экране будут за пределами видимой области, и для просмотра этих частей потребуется переместить курсор мыши к нижней или правой границе экрана.

Параметр **Разрешить сервис VNC** после применения конфигурации активирует встроенный сервис удаленного доступа к содержимому экрана монитора, подключенного к контроллеру, при этом физический видеомонитор может быть выключен. Для удаленного подключения и просмотра форм визуализации возможно использовать приложение VNC Viewer или аналогичное.

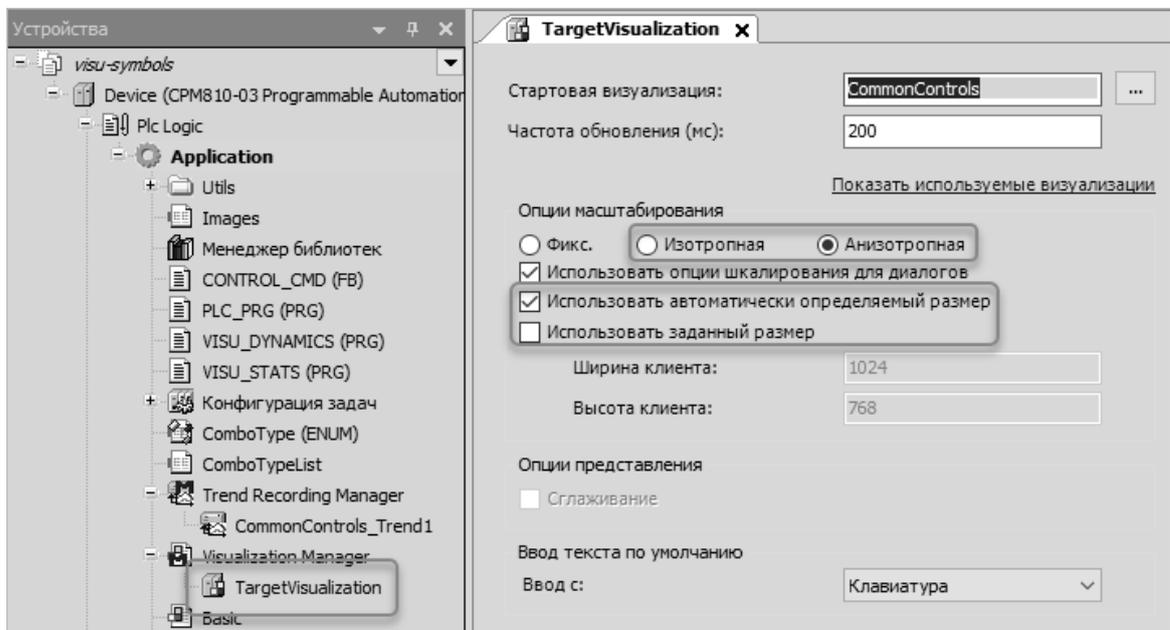


Рисунок 145 – Установка анизотропного масштабирования с автоматическим определением размера видимой области

ВНИМАНИЕ!

Не рекомендуется оставлять постоянный доступ к контроллеру по протоколу VNC, т.к. соединение по данному протоколу не защищено паролем.

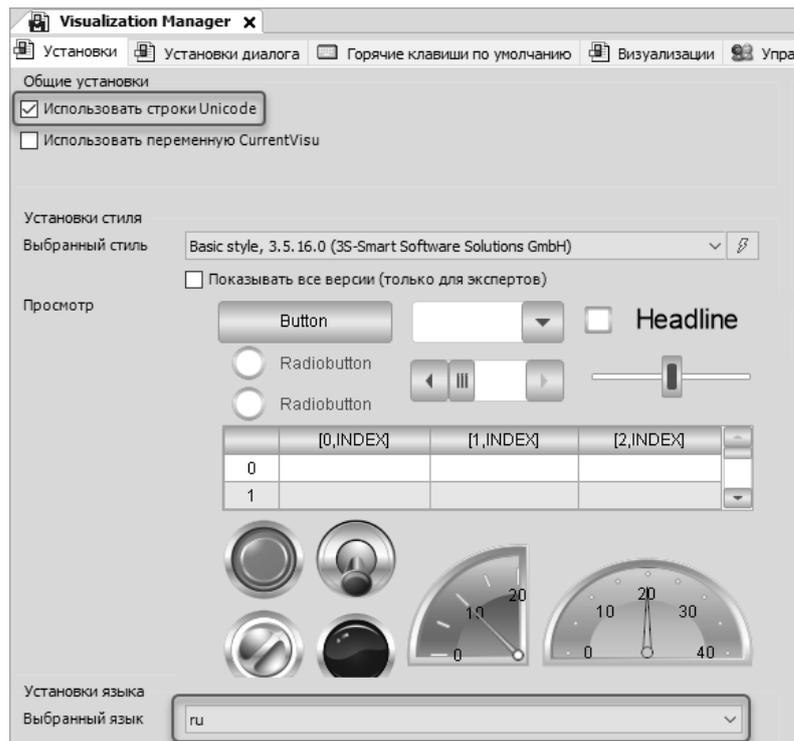


Рисунок 146 – Включение поддержки русского языка ввода в формах визуализации

Для включения поддержки ввода текстовых строк на русском языке в формах визуализации следует отметить опцию **Использовать строки Unicode** и установить **Выбранный язык: ru** в свойствах **Менеджера визуализации (Visualization Manager)** проекта IDE МЭК 61131-3, как показано на рисунке 146.

Строковые переменные, используемые в приложении МЭК 61131-3 для взаимодействия с формами визуализации, должны иметь тип WSTRING.

При использовании клавиатуры для переключения языка ввода с английского на русский и наоборот следует использовать сочетание клавиш Alt+Shift.

5.13. Web-конфигуратор

Страница **Web-конфигуратор**, показанная на рисунке 147, служит для настройки параметров веб-конфигуратора контроллера, реализуемого встроенным веб-сервером, а также для полного отключения веб-сервера.

The screenshot shows a configuration form with the following elements:

- A checkbox labeled "Запретить Web-конфигуратор:" which is currently unchecked.
- An "IP-адрес:" label followed by a dropdown menu showing "По умолчанию".
- A "Порт (по умолчанию 80):" label followed by an empty text input field.
- A checkbox labeled "Использовать HTTPS:" which is currently checked.
- An "HTTPS порт (по умолчанию 443):" label followed by an empty text input field.

Рисунок 147 – Страница настройки параметров веб-конфигуратора

Выпадающий список **IP-адрес** позволяет выбрать IP-адрес сетевого интерфейса (LAN1 или LAN2), на который устанавливается веб-сервер. При использовании значения *По умолчанию* веб-сервер устанавливается на все доступные сетевые интерфейсы контроллера.

Поле **Порт (по умолчанию 80)** предназначено для ввода номера TCP-порта, на который будет устанавливаться веб-сервер. Обратите внимание, что после изменения исходного значения TCP-порта и применения конфигурации, при входе на веб-сервер в адресной строке браузера после IP-адреса контроллера и двоеточия потребуется ввести номер установленного порта. Например, если установлен номер порта 8090, то для соединения с веб-сервером контроллера по адресу 10.0.0.3 в адресной строке браузера нужно ввести:

`http://10.0.0.3:8090/?lang=ru`

Дополнительная опция `/?lang=ru` позволяет выбрать русский язык интерфейса веб-конфигуратора при первом же входе на веб-сервер контроллера.

После установки опции **Запретить Web-конфигуратор** и нажатия кнопки **Применить конфигурацию** контроллер будет функционировать без запущенного веб-сервера, в результате чего не будет возможности просмотреть или изменить его системные параметры при помощи браузера.

В дальнейшем для повторного включения веб-конфигуратора:

1. Переведите контроллер в заводской режим согласно указаниям п. 4.2.1.
2. Выполните вход на веб-сервер.
3. Снимите опцию **Запретить Web-конфигуратор** на странице **Web-конфигуратор**.
4. Нажмите **Применить конфигурацию**.

После перезапуска контроллера следует вернуть переключатели в первоначальное положение (до перехода в заводской режим) и еще раз перезапустить контроллер.

Обратите внимание, что в заводском режиме для входа на веб-сервер контроллера придется использовать исходные IP-адреса сетевых интерфейсов LAN1: 10.0.0.100/8 или LAN2: 10.0.0.101/8, но при этом потребуется ввести пароль учетной записи *Administrator*, заданный до перевода контроллера в заводской режим.

Опция **Использовать HTTPS** предназначена для включения режима защищенного подключения к веб-конфигуратору с использованием самоподписанного сертификата компонента `SmrFastweICore` (см. п. 5.11). При включении опции **Использовать HTTPS** также становится доступной поле ввода TCP-порта для подключения по протоколу HTTPS. Например, для соединения с веб-сервером контроллера с разрешенной опцией **Использовать HTTPS** по адресу 10.0.0.4 по протоколу HTTPS в адресной строке браузера нужно ввести:

`https://10.0.0.4`

В окне браузера будет выведено сообщение, аналогичное показанному на рисунке 148.

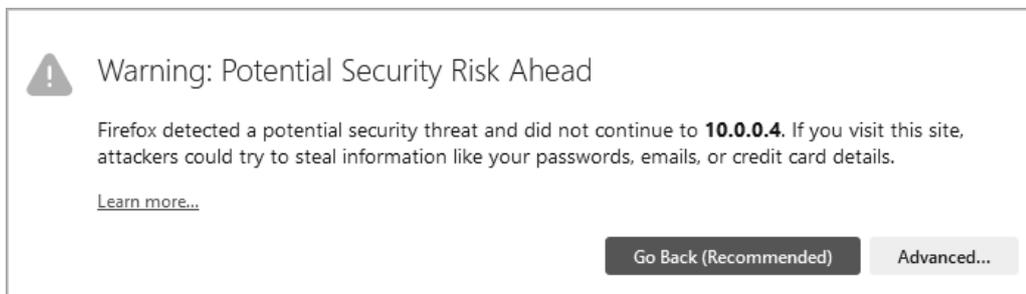


Рисунок 148 – Сообщение браузера (Firefox) при подключении к контроллеру по HTTPS

Для продолжения подключения нажмите кнопку **Advanced** и **Accept the Risk and Continue**, как показано на рисунке 149.

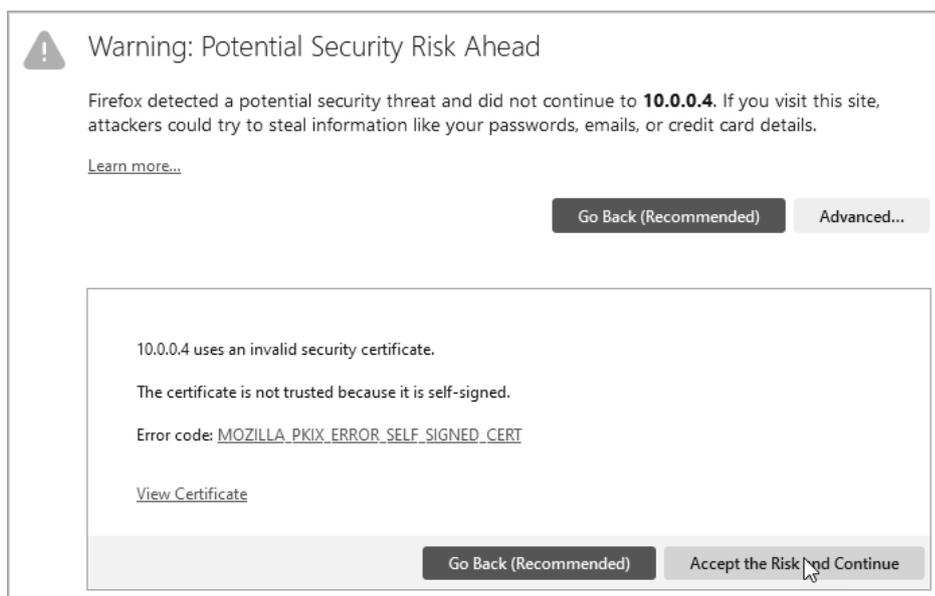


Рисунок 149 – Разрешение подключения по HTTPS к узлу с самоподписанным сертификатом

5.14. Установка паролей

Информация об учетных записях пользователей и управлении доступом приведена в руководстве пользователя Fastwel PLC Application Toolkit.

Страница **Установка паролей**, показанная на рисунке 150, предназначена для изменения или сброса паролей двух встроенных учетных записей *Administrator* и *Everyone* подсистемы управления доступом системного программного обеспечения контроллера.

Показать редактируемые пароли:

Administrator: Не изменен

Сбросить Отменить изменения

Everyone: Не изменен (пустой)

Сбросить Отменить изменения

Рисунок 150 – Страница установки и сброса паролей

Кнопка **Сбросить** служит для превращения пароля соответствующей учетной записи в пустую строку. При сброшенном пароле в соответствующем поле справа от имени учетной записи отображается строка *Введен пустой*, если пароль был только что сброшен, или *Не изменен (пустой)*, если в настоящий момент для данной учетной записи используется пустой пароль.

ВНИМАНИЕ!

При вводе непустого пароля следует использовать только символы таблицы ASCII. Не следует использовать символы кириллицы и других национальных кодировок.

Если нет полной уверенности в правильности "слепого" ввода нового пароля, до начала ввода следует отметить флажок **Показать редактируемые пароли**. В таком случае текущие установленные пароли не будут отображены в соответствующих полях страницы, однако вводимый новый пароль будет отображаться в течение нескольких секунд, после чего в поле ввода пароля будет вновь отображена строка *Введен новый*, свидетельствующая о намерении изменения ранее установленного пароля.

Если после ввода нового пароля до нажатия кнопки **Применить конфигурацию** новый пароль по какой-то причине забыт, можно отменить ввод нажатием кнопки **Отменить изменения**.

Применение вновь введенных паролей произойдет после нажатия кнопки **Применить конфигурацию** и последующего перезапуска контроллера. Обратите внимание, что учетные записи *Administrator* и *Everyone* используются для доступа к контроллеру по всем сервисным каналам связи: HTTP/HTTPS – для входа на веб-сервер; FTP/FTPS/SFTP – для загрузки и выгрузки файлов по протоколам FTP/FTPS/SFTP, а также при подключении к контроллеру из IDE МЭК 61131-3.

Если для учетной записи *Everyone* установлен непустой пароль, то для доступа к контроллеру из IDE МЭК 61131-3 потребуется использовать реквизиты одной из двух встроенных учетных записей: *Administrator* или *Everyone*.

Обратите внимание, что забытый пароль может быть сброшен только сервисной службой предприятия-изготовителя.

5.15. Оболочка ПЛК

Страница **Оболочка ПЛК (PLC Shell)** позволяет выполнять команды оболочки ПЛК IDE МЭК 61131-3 на соответствующей вкладке редактора устройства, представляющего контроллер.

Внешний вид страницы **Оболочка ПЛК** после ввода команды *ipinfo* показан на рисунке 151.

Для выполнения команды оболочки ПЛК следует ввести команду с параметрами в поле **Выполнить** и нажать **Ввод**. Результат выполнения команды будет отображен под полем ввода команды.

Для отображения информации о поддерживаемых командах введите символ ? в поле **Выполнить** и нажмите **Ввод**.

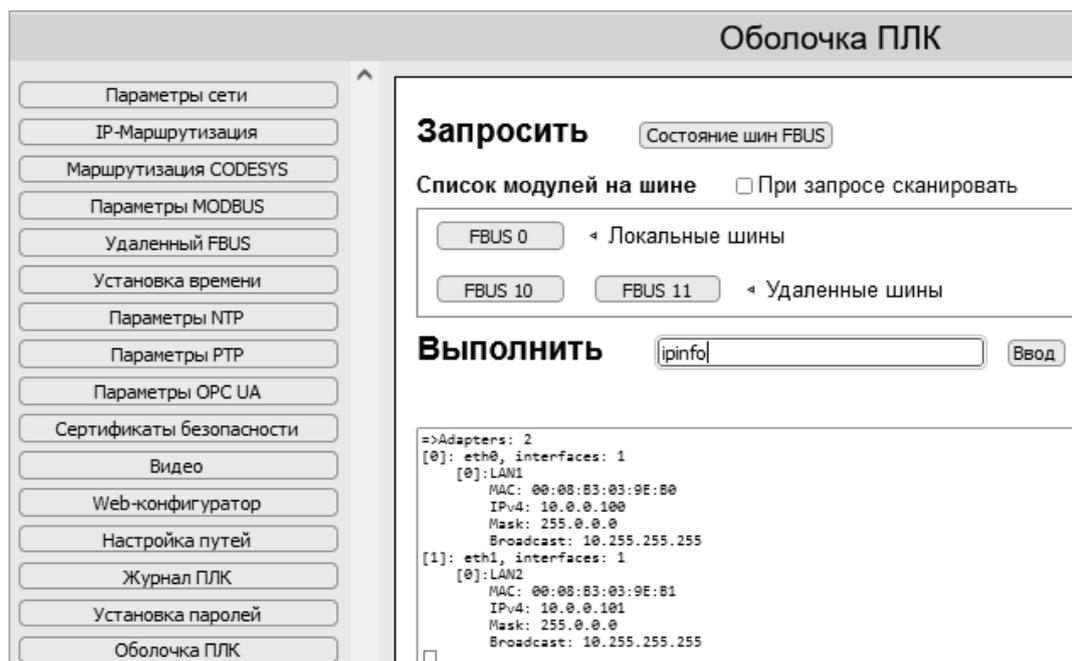


Рисунок 151 – Внешний вид страницы Оболочка ПЛК

Команды получения информации о состоянии шин FBUS и о серийных номерах и версиях микропрограмм модулей ввода-вывода представлены кнопками:

Состояние шин FBUS – запрашивает и отображает информацию о состоянии всех поддерживаемых локальных и удаленных шин FBUS.

FBUS 0 – запрашивает и отображает информацию о серийных (заводских) номерах и версиях микропрограмм периферийных модулей, подключенных к локальной шине FBUS.

FBUS 10, FBUS 11 – данные кнопки предназначены для запроса и отображения информации о серийных (заводских) номерах и версиях микропрограмм периферийных модулей, подключенных к удаленным шинам FBUS, реализованным на базе модулей NIM745-01.

Опция **При запросе сканировать (On query rescan FBUS)** позволяет выполнить полное сканирование шины при запросе информации о модулях на шине в ситуации, когда шина находится в частично исправном или неисправном состоянии.

Обратите внимание, что при запросе информации о модулях с установленной опцией **При запросе сканировать** обмен данными приложения с периферийными модулями временно прекращается, что может привести к непредвиденным последствиям.

5.16. Система

5.16.1. Общие сведения

На странице **Система** отображена информация о версии системного программного обеспечения контроллера и серийный (заводской) номер контроллера. Кроме того, с данной страницы могут быть выполнены некоторые сервисные операции:

1. Включение и отключение поддержки USB-устройств.
2. Обновление системного программного обеспечения контроллера.
3. Обновление микропрограмм модулей ввода-вывода, подключенных ко всем портам шины FBUS контроллера.
4. Развертывание приложения IDE МЭК 61131-3 и конфигурации контроллера.
5. Возврат исходных заводских значений системным параметрам контроллера.
6. Перезапуск контроллера.

5.16.2. Включение и отключение поддержки USB-устройств

USB-устройства могут быть оказывать влияние на работоспособность контроллера, поэтому по умолчанию поддержка USB-устройств выключена. Для включения поддержки USB следует снять отметку с флажка **Отключение USB-устройств** и нажать **Применить конфигурацию**.

5.16.3. Загрузка системных файлов

Кнопка **Загрузить системный файл** служит для записи в контроллер одного из системных файлов:

norm.dnl – файл обновления системного программного обеспечения контроллера;

ffw.dnl – файл обновления микропрограмм модулей ввода-вывода;

norm.upl – файл развертывания приложения IDE МЭК 61131-3 и конфигурации контроллера, сформированный на другом (эталонном) контроллере командой оболочки ПЛК *saveapp*.

После нажатия кнопки **Загрузить системный файл** справа от нее отображается кнопка **Обзор... (Browse...)**, позволяющая выбрать требуемый системный файл на ПК, на котором запущен браузер. В процессе загрузки файла в окне браузера отображается информация о выполнении загрузки, а по окончании загрузки на странице **Система** выводится сообщение *Системный файл загружен*.

После успешной загрузки для применения обновления или развертывания приложения следует нажать кнопку **Перезапустить контроллер**.

Более подробная информация об обновлении системного программного обеспечения контроллера и модулей ввода-вывода приведена в руководстве пользователя Fastwel PLC Application Toolkit и в

документе ИМЕС.421459.252РЭ. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 1. Общие сведения.

5.16.4. Восстановление системных параметров контроллера

Для возврата исходных значений всех системных параметров контроллера и исходных паролей встроенных учетных записей следует нажать кнопку **Восстановить заводские настройки**. После нажатия кнопки произойдет перезапуск контроллера и восстановление системных параметров.

Обратите внимание, что после выполнения команды **Восстановить заводские настройки**, состояние контроллера будет отличаться от исходного в следующих аспектах:

1. Если в контроллер ранее было загружено приложение IDE МЭК 61131-3, то после восстановления заводских настроек работа приложения будет продолжена. Для удаления приложения воспользуйтесь командой **Онлайн – Сброс заводской** в среде разработки.
2. После восстановления заводских настроек все имеющиеся цифровые сертификаты подсистемы безопасности остаются неизменными в тех же зонах, где они находились до восстановления заводских настроек.

Таким образом, если до восстановления заводских настроек в поле **Имя хоста** на странице **Параметры сети** было задано имя узла сети, совпадающее со значением **commonName** в имеющихся цифровых сертификатах, то после восстановления заводских настроек потребуется повторно ввести это имя узла сети в поле **Имя хоста** на странице **Параметры сети** и применить конфигурацию.

Если этого не сделать, то имеющиеся цифровые сертификаты будут недействительными, что сделает невозможным создание безопасных сеансов связи с контроллером.

5.16.5. Перезапуск контроллера

Для полного перезапуска контроллера следует нажать кнопку **Перезапустить контроллер**. Обратите внимание, что все изменения системных параметров, произведенные в браузере после последнего успешного применения конфигурации, будут утрачены после перезапуска контроллера.

При любом перезапуске и при включении питания контроллера индикаторы RUN, APP, I/O и USR1 выключаются, затем кратковременно светятся красным цветом, после чего опять выключаются. Далее, в зависимости от необходимости выполнить настройку системных параметров, обновить системное программное обеспечение или развернуть приложение IDE МЭК 61131-3 из файла *norm.upl*, светятся желтым цветом один из индикаторов RUN, APP или I/O. При успешном завершении процесса обновления системного ПО или развертывания приложения четыре указанных индикатора кратковременно светятся зеленым цветом.

Информация об индикации контроллера приведена в документе ИМЕС.421459.252РЭ1. Комплекс программно-технический Fastwel I/O-2. Руководство по эксплуатации. Часть 2. Составные части комплекса.

5.17. Настройка путей

Система исполнения приложений МЭК 61131-3 по умолчанию сохраняет записи журнала ПЛК в файлах, создаваемых в каталоге *user/plc_log* встроенного дискового накопителя.

Кроме того, при включенной архивации трендов и аварийных событий в конфигурации соответствующих компонентов приложения базы данных трендов и аварийных событий сохраняются в каталогах *user/trend* и *user/alarms* встроенного дискового накопителя.

Страница **Настройка путей** веб-конфигуратора, показанная на рисунке 152, предназначена для выбора других каталогов сохранения перечисленной выше информации, в том числе создаваемых на съемных USB-накопителях.

Разрешить изменение стандартных путей:

Тренды

trend

Журнал аварийных событий (alarms)

alarms

Журнал ПЛК (PlcLog)

plc_log

Рисунок 152 – Страница настройки путей с включенной опцией "Разрешить изменение стандартных путей"

Опция **Разрешить изменение стандартных путей** раскрывает варианты изменения путей для хранения баз данных трендов, аварийных событий и файлов журнала ПЛК.

Установка нового пути для сохранения данных любой из перечисленных категорий выполняется следующим образом:

1. При необходимости подключите съемный дисковый накопитель к контроллеру.
2. Включите опцию **Разрешить изменение стандартных путей**.
3. Нажмите кнопку **Просмотр** справа от поля, содержащего текущее значение пути для выбранной категории сохраняемых данных. Под кнопкой **Просмотр** будет отображена область, содержащая доступные для выбора каталоги относительно текущего корневого каталога, как показано на рисунке 153.
4. Создайте новый подкаталог нажатием кнопки или выберите имеющийся в списке нажатием кнопки .

При выборе подкаталога, соответствующего корневному каталогу съемного дискового накопителя, например, *usb2p1*, рекомендуется создать в корневом каталоге хотя бы один подкаталог для сохранения данных выбранной категории, как показано на рисунке 154.

Для перехода на один уровень вверх по иерархии каталогов нажмите кнопку .

Для перехода в корневой каталог иерархии каталогов нажмите кнопку .

Для удаления ненужного каталога нажмите .

5. Для установки требуемого пути для сохранения данных выбранной категории нажмите кнопку , расположенную слева от выбранного подкаталога.

Выбранный путь относительно корневого каталога пользователя user будет отображен в текстовом поле слева от кнопки **Просмотр**, при этом в случае, если выбранный путь расположен на съемном дисковом накопителе, справа от названия категории сохраняемых данных будет отображена строка *сменный носитель*, как показано на рисунке 155.

6. В веб-конфигураторе нажмите кнопку **Применить конфигурацию**. После перезапуска контроллера настройки путей вступят в силу.

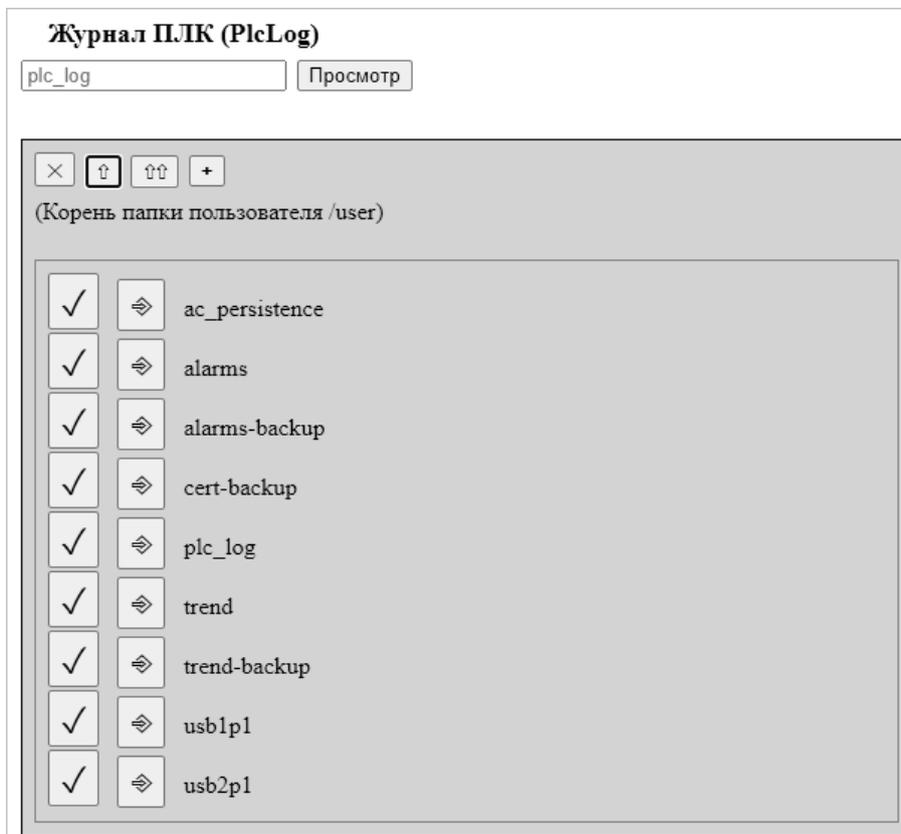
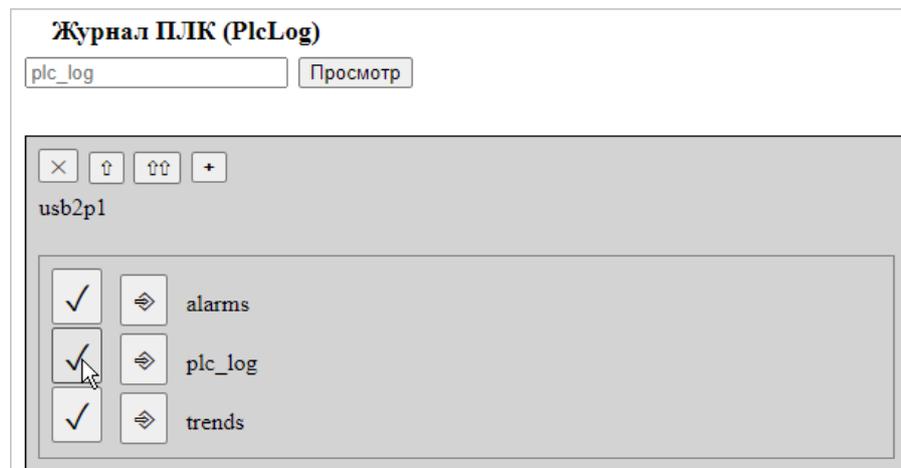


Рисунок 153 – Просмотр доступных каталогов относительно текущего каталога

Рисунок 154 – Каталоги на съемном дисковом накопителе *usb2p1*Рисунок 155 – Выбранный каталог *plc_log* на USB-накопителе, установленном в гнездо USB2

Для установки исходного пути для одной из категорий сохраняемых данных:

1. Нажмите кнопку **Просмотр** справа от текстового поля, содержащего текущий выбранный путь.
2. В появившейся области отображения подкаталогов текущего выбранного пути нажмите кнопку .
3. Нажмите кнопку  слева от исходного подкаталога в корневом каталоге пользователя *user*, как показано на рисунке 156.

Для журнала ПЛК следует выбрать подкаталог *plc_log* корневого каталога, для трендов – *trend*, для аварийных событий – *alarms*.

4. В веб-конфигураторе нажмите кнопку **Применить конфигурацию**. После перезапуска контроллера настройки путей вступят в силу.

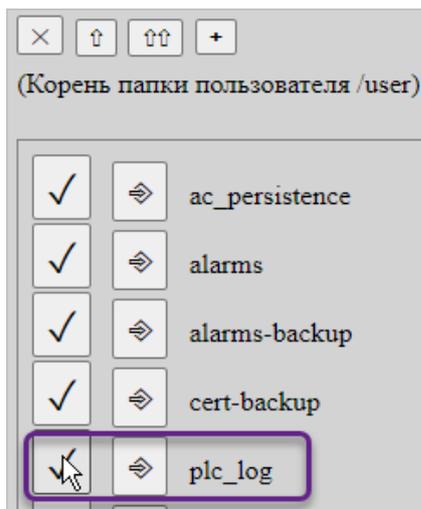


Рисунок 156 – Возврат исходного пути для сохранения файлов журнала ПЛК

5.18. Журнал ПЛК

Страница **Журнал ПЛК** веб-конфигуратора, показанная на рисунке 157, предназначена для настройки параметров журнала сообщений системы исполнения приложений МЭК 61131-3 контроллера, а также для включения сохранения журнала в файлах и просмотра сообщений системы исполнения, сохраненных в файлах.

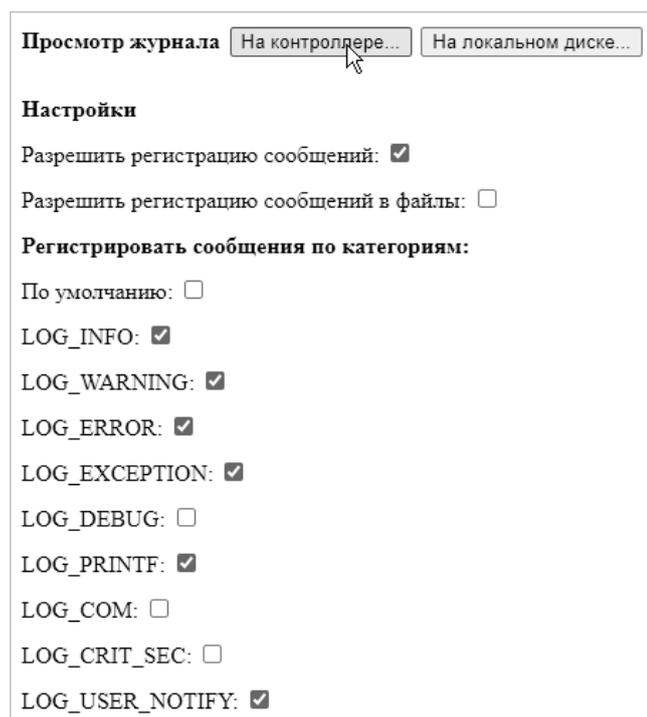


Рисунок 157 – Страница "Журнал ПЛК"

По умолчанию регистрация сообщений в журнале включена, что позволяет просматривать сообщения системы исполнения от момента последнего включения контроллера в IDE МЭК 61131-3 на вкладке **Журнал**, как показано на рисунке 158, при этом количество просматриваемых сообщений ограничено размером буфера в среде разработки, кроме того сообщения в журнале по умолчанию сохраняются до выключения питания контроллера.

Для сохранения сообщений журнала на встроенном или внешнем дисковом накопителе контроллера следует включить опцию **Разрешить регистрацию сообщений в файлы**.

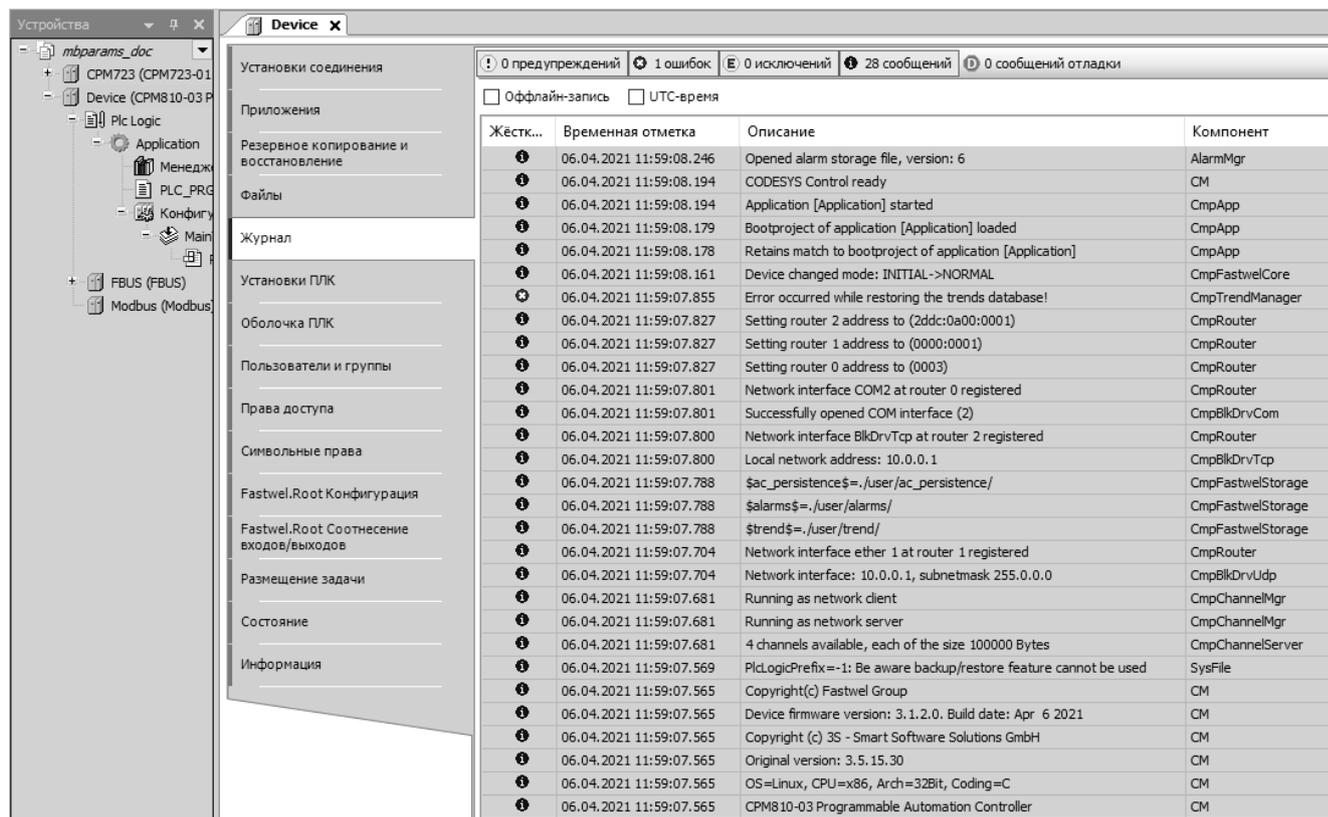
Количество и максимальный размер создаваемых файлов, образующих циклический буфер, определяются параметрами **Файлов, не более** и **Макс. размер файла, в байтах** соответственно.

Обратите внимание, что для уменьшения вероятности повреждения текущего файла журнала ПЛК при внезапном выключении питания рекомендуется максимальный размер ограничить несколькими сотнями килобайт.

По умолчанию для размещения файлов журнала ПЛК используется каталог *user/plc_log*. Для выбора другого каталога, в том числе на съемном дисковом накопителе, следуйте указаниям п. 5.17 настоящего руководства.

Группа параметров **Регистрировать сообщения по категориям** позволяет исключить регистрацию сообщений системы исполнения выбранных категорий в соответствии с таблицей 7.

Для просмотра записей журнала ПЛК, сохраненных в файлах, следует нажать кнопку **Просмотр журнала – На контроллере**, а затем нажать кнопку , расположенную слева от имени файла журнала с расширением csv, как показано на рисунке 159.



Жестк...	Временная отметка	Описание	Компонент
	06.04.2021 11:59:08.246	Opened alarm storage file, version: 6	AlarmMgr
	06.04.2021 11:59:08.194	CODESYS Control ready	CM
	06.04.2021 11:59:08.194	Application [Application] started	CmpApp
	06.04.2021 11:59:08.179	Bootproject of application [Application] loaded	CmpApp
	06.04.2021 11:59:08.178	Retains match to bootproject of application [Application]	CmpApp
	06.04.2021 11:59:08.161	Device changed mode: INITIAL->NORMAL	CmpFastwelCore
	06.04.2021 11:59:07.855	Error occurred while restoring the trends database!	CmpTrendManager
	06.04.2021 11:59:07.827	Setting router 2 address to (2ddc:0a00:0001)	CmpRouter
	06.04.2021 11:59:07.827	Setting router 1 address to (0000:0001)	CmpRouter
	06.04.2021 11:59:07.827	Setting router 0 address to (0003)	CmpRouter
	06.04.2021 11:59:07.801	Network interface COM2 at router 0 registered	CmpRouter
	06.04.2021 11:59:07.801	Successfully opened COM interface (2)	CmpBlkDrvCom
	06.04.2021 11:59:07.800	Network interface BlkDrvTcp at router 2 registered	CmpRouter
	06.04.2021 11:59:07.800	Local network address: 10.0.0.1	CmpBlkDrvTcp
	06.04.2021 11:59:07.788	\$ac_persistence\$=./user/ac_persistence/	CmpFastwelStorage
	06.04.2021 11:59:07.788	\$alarms\$=./user/alarms/	CmpFastwelStorage
	06.04.2021 11:59:07.788	\$trend\$=./user/trend/	CmpFastwelStorage
	06.04.2021 11:59:07.704	Network interface ether 1 at router 1 registered	CmpRouter
	06.04.2021 11:59:07.704	Network interface: 10.0.0.1, subnetmask 255.0.0.0	CmpBlkDrvUdp
	06.04.2021 11:59:07.681	Running as network client	CmpChannelMgr
	06.04.2021 11:59:07.681	Running as network server	CmpChannelMgr
	06.04.2021 11:59:07.681	4 channels available, each of the size 100000 Bytes	CmpChannelServer
	06.04.2021 11:59:07.569	PlcLogicPrefix=-1: Be aware backup/restore feature cannot be used	SysFile
	06.04.2021 11:59:07.565	Copyright(c) Fastwel Group	CM
	06.04.2021 11:59:07.565	Device firmware version: 3.1.2.0. Build date: Apr 6 2021	CM
	06.04.2021 11:59:07.565	Copyright (c) 3S - Smart Software Solutions GmbH	CM
	06.04.2021 11:59:07.565	Original version: 3.5.15.30	CM
	06.04.2021 11:59:07.565	OS=Linux, CPU=x86, Arch=32Bit, Coding=C	CM
	06.04.2021 11:59:07.565	CPM810-03 Programmable Automation Controller	CM

Рисунок 158 – Сообщения системы исполнения в журнале IDE М&К 61131-3

Записи журнала из выбранного файла будут отображены в дополнительном окне браузера, как показано на рисунке 161.

Каждый запуск контроллера сопровождается формированием в текущем файле журнала блока заголовка с метаданной о фрагменте журнала, который может быть отображен отметкой флажка **Метаинформация – Показать**, как показано на рисунке 160.

Таблица 7 – Категории сообщений в журнале ПЛК

Категория	Значение ClassId	Описание категории
LOG_INFO	1	Информационные сообщения
LOG_WARNING	2	Предупреждения
LOG_ERROR	4	Сообщения об ошибках
LOG_EXCEPTION	8	Сообщения о критических ошибках, как правило, переводящих контроллер в безопасный режим
LOG_DEBUG	16	Отладочные сообщения
LOG_PRINTF	32	Сообщения без метки времени (например, элементы разметки журнала)
LOG_COM	64	Отладочные сообщения коммуникационных сервисов взаимодействия со средой разработки (не используется)
LOG_CRIT_SEC	256	Не используется
LOG_USER_NOTIFY	65536	Сообщения системы исполнения, приводящие к выводу диалоговых окон сообщений в среде разработки, подключенной к контроллеру

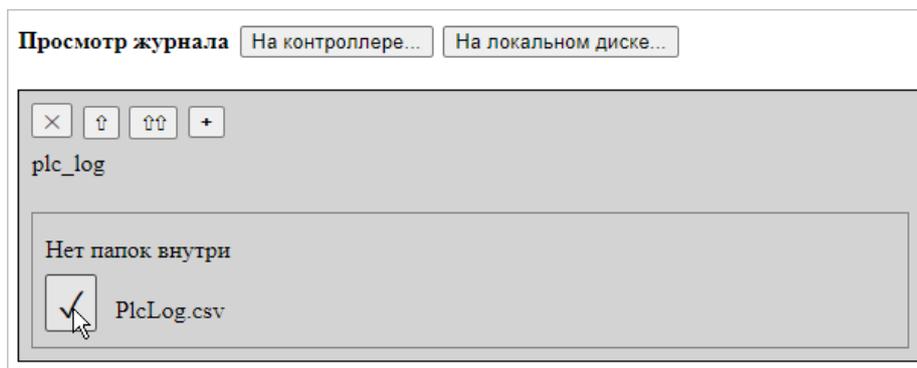


Рисунок 159 – Выбор файла сообщений журнала ПЛК для просмотра

```

Метаинформация  Показать
;*****
;loggername>PlcLog</loggername>
;logoptions>
;
;  <enable>1</enable>
;  <path>user/plc_log</path>
;  <type>normal</type>
;  <timestamp>rtc high resolution</timestamp>
;  <deactivatable>0</deactivatable>
;  <dump>always</dump>
;  <filter>0x0001002f</filter>
;  <maxentries>500</maxentries>
;  <maxfiles>3</maxfiles>
;  <maxfilesize>100000</maxfilesize>
;</logoptions>
;entries>
;Timestamp, CmpId, ClassId, ErrorId, InfoId, InfoText
;ClassId: LOG_INFO          =1
;ClassId: LOG_WARNING      =2
;ClassId: LOG_ERROR        =4
;ClassId: LOG_EXCEPTION    =8
;ClassId: LOG_DEBUG        =16
;ClassId: LOG_PRINTF       =32
;ClassId: LOG_COM          =64
;</entries>
;*****
© 2021-04-05T14:01:58.670Z 0x00000001 32 =====
© 2021-04-05T14:01:58.670Z 0x00000001 1 4 CPM810-03 Program

```

Рисунок 160 – Метаинформация из блока заголовка фрагмента сохраненного журнала ПЛК

В секции *logoptions* заголовка перечислены текущие параметры журнала ПЛК, а в секции *entries* – заголовки столбцов.

Сообщения системы исполнения выводятся в табличной форме, при этом наиболее существенна информация в столбцах с метками времени (*Timestamp*), классами сообщений (*ClassId*) и строками сообщений (*InfoText*).

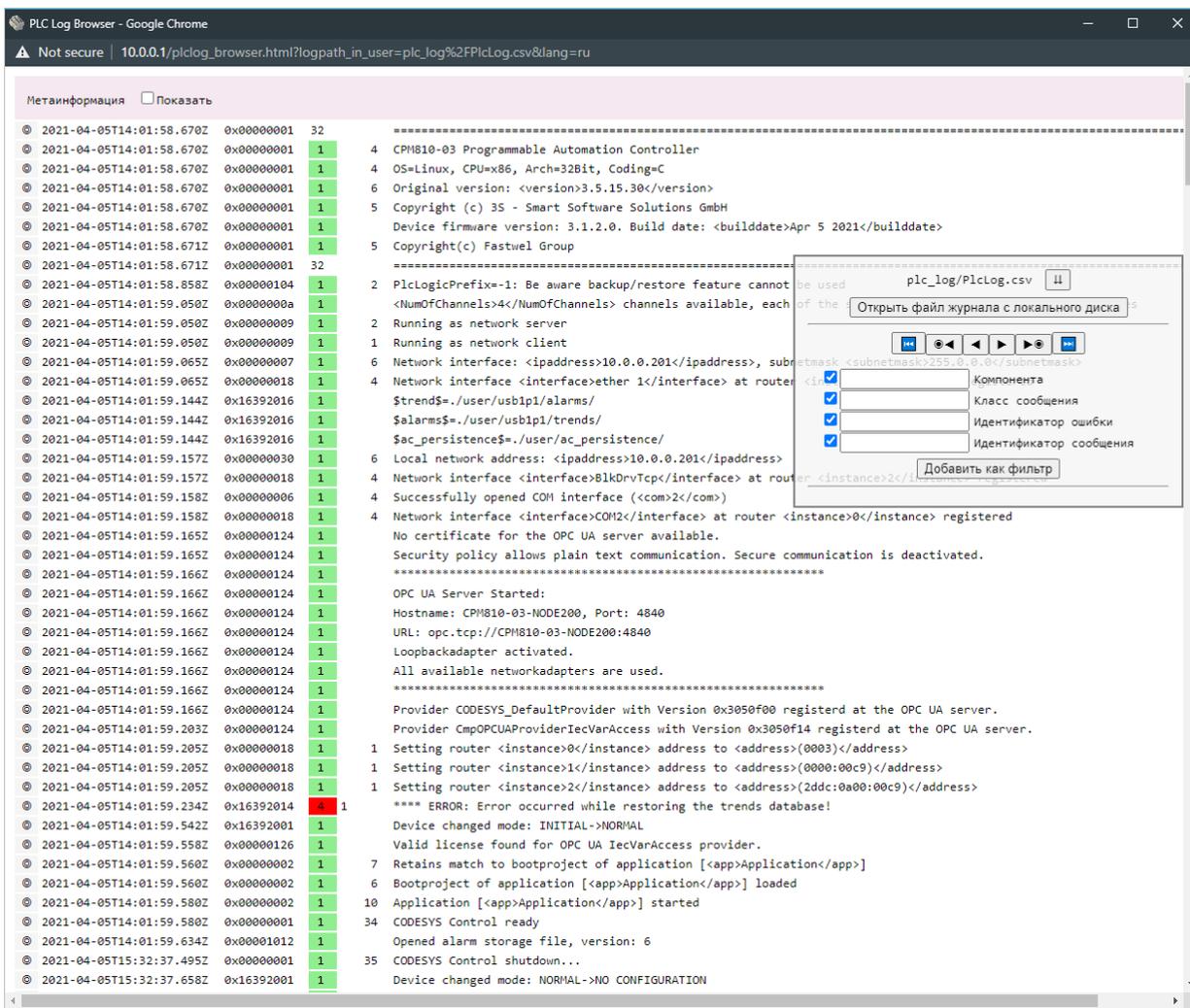


Рисунок 161 – Просмотр содержимого файла сообщений журнала ПЛК в браузере

Первый столбец (*Timestamp*) содержит метки времени сообщений с точностью до миллисекунд.

Третий столбец (*ClassId*) содержит числовые идентификаторы категорий сообщений в соответствии с таблицей 7, некоторые из которых выделены цветом.

Крайний правый столбец содержит текст сообщений системы исполнения.

Для навигации по сообщениям журнала может быть использована панель навигации, показанная на рисунке 162 и отображаемая поверх таблицы сообщений.

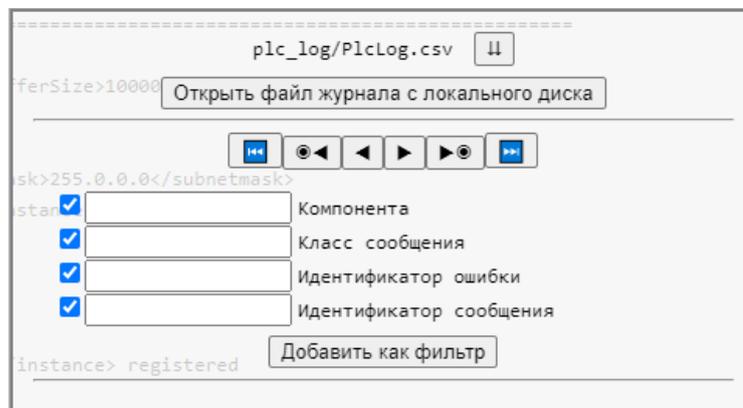


Рисунок 162 – Панель навигации по сообщениям журнала ПЛК

В верхней части панели навигации отображается путь и имя выбранного файла журнала ПЛК.

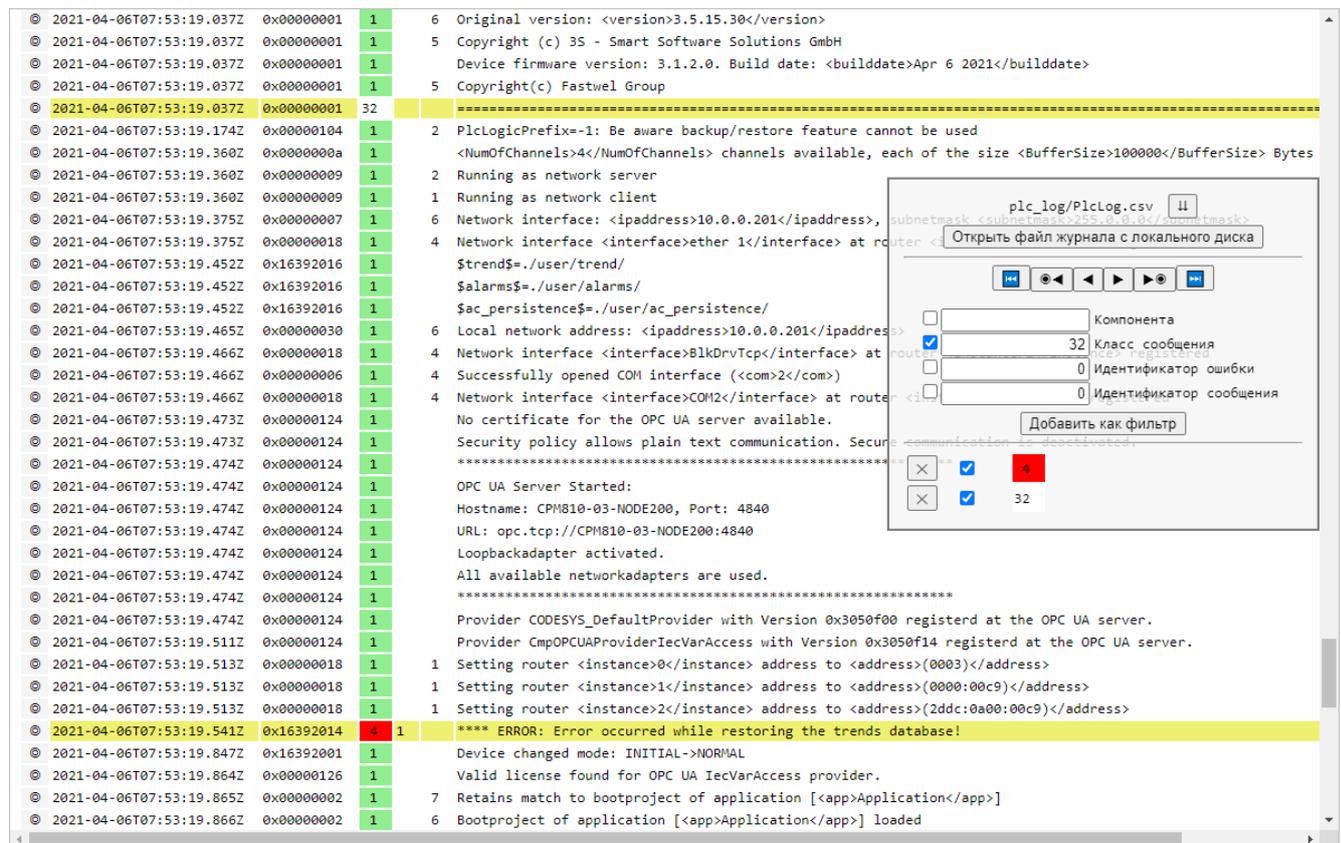
Кнопка  позволяет сохранить файл журнала на компьютере для последующего просмотра. Кнопка **Открыть файл журнала с локального диска** предназначена для открытия с целью просмотра ранее сохраненного файла журнала на компьютере.

Кнопки  и  служат для перехода и выбора самого первого ("старого") и последнего ("нового") сообщения журнала соответственно.

Кнопки  и  предназначены для перехода к предыдущему и следующему сообщениям журнала относительно текущего выбранного сообщения.

Для поиска и выделения цветом сообщений, соответствующих определенным условиям, возможно определить один или несколько фильтров. Для этого следует заполнить одно или несколько полей из доступных (**Компонента**, **Класс сообщения**, **Идентификатор ошибки** и **Идентификатор сообщения**) и нажать кнопку **Добавить как фильтр**. Таким образом возможно определить несколько условий для поиска и выделения сообщений, как показано на рисунке 163.

После определения всех фильтров для поиска сообщений следует нажать кнопку **Применить фильтры**, после чего сообщения, соответствующие заданным условиям по всем фильтрам будут выделены, как показано на рисунке 163. Кнопки  и  позволяют переходить к предыдущему и следующему сообщению среди сообщений, соответствующих условиям по текущим заданным фильтрам.



Timestamp	Hex ID	Severity	Count	Message
2021-04-06T07:53:19.037Z	0x00000001	1	6	Original version: <version>3.5.15.30</version>
2021-04-06T07:53:19.037Z	0x00000001	1	5	Copyright (c) 3S - Smart Software Solutions GmbH
2021-04-06T07:53:19.037Z	0x00000001	1		Device firmware version: 3.1.2.0. Build date: <builddate>Apr 6 2021</builddate>
2021-04-06T07:53:19.037Z	0x00000001	1	5	Copyright(c) Fastwel Group
2021-04-06T07:53:19.037Z	0x00000001	32		-----
2021-04-06T07:53:19.174Z	0x00000104	1	2	PlcLogicPrefix=-1: Be aware backup/restore feature cannot be used
2021-04-06T07:53:19.360Z	0x0000000a	1		<NumOfChannels>4</NumOfChannels> channels available, each of the size <BufferSize>100000</BufferSize> Bytes
2021-04-06T07:53:19.360Z	0x00000009	1	2	Running as network server
2021-04-06T07:53:19.360Z	0x00000009	1	1	Running as network client
2021-04-06T07:53:19.375Z	0x00000007	1	6	Network interface: <ipaddress>10.0.0.201</ipaddress>, subnetmask <subnetmask>255.255.255.0</subnetmask>
2021-04-06T07:53:19.375Z	0x00000018	1	4	Network interface <interface>ether 1</interface> at router <router>1</router>
2021-04-06T07:53:19.452Z	0x16392016	1		\$trend\$=./user/trend/
2021-04-06T07:53:19.452Z	0x16392016	1		\$alarms\$=./user/alarms/
2021-04-06T07:53:19.452Z	0x16392016	1		\$ac_persistence\$=./user/ac_persistence/
2021-04-06T07:53:19.465Z	0x00000030	1	6	Local network address: <ipaddress>10.0.0.201</ipaddress>
2021-04-06T07:53:19.466Z	0x00000018	1	4	Network interface <interface>BlkDrvTcp</interface> at router <router>1</router>
2021-04-06T07:53:19.466Z	0x00000006	1	4	Successfully opened COM interface (<com>2</com>)
2021-04-06T07:53:19.466Z	0x00000018	1	4	Network interface <interface>COM2</interface> at router <router>1</router>
2021-04-06T07:53:19.473Z	0x00000124	1		No certificate for the OPC UA server available.
2021-04-06T07:53:19.473Z	0x00000124	1		Security policy allows plain text communication. Secure communication is not available.
2021-04-06T07:53:19.474Z	0x00000124	1		OPC UA Server Started:
2021-04-06T07:53:19.474Z	0x00000124	1		Hostname: CPM810-03-NODE200, Port: 4840
2021-04-06T07:53:19.474Z	0x00000124	1		URL: opc.tcp://CPM810-03-NODE200:4840
2021-04-06T07:53:19.474Z	0x00000124	1		Loopbackadapter activated.
2021-04-06T07:53:19.474Z	0x00000124	1		All available networkadapters are used.
2021-04-06T07:53:19.474Z	0x00000124	1		-----
2021-04-06T07:53:19.474Z	0x00000124	1		Provider CODESYS_DefaultProvider with Version 0x3050f00 registered at the OPC UA server.
2021-04-06T07:53:19.511Z	0x00000124	1		Provider CmpOPCUAProviderIecVarAccess with Version 0x3050f14 registered at the OPC UA server.
2021-04-06T07:53:19.513Z	0x00000018	1	1	Setting router <instance>0</instance> address to <address>(0003)</address>
2021-04-06T07:53:19.513Z	0x00000018	1	1	Setting router <instance>1</instance> address to <address>(0000:00c9)</address>
2021-04-06T07:53:19.513Z	0x00000018	1	1	Setting router <instance>2</instance> address to <address>(2ddc:0a00:00c9)</address>
2021-04-06T07:53:19.541Z	0x16392014	4	1	**** ERROR: Error occurred while restoring the trends database!
2021-04-06T07:53:19.847Z	0x16392001	1		Device changed mode: INITIAL->NORMAL
2021-04-06T07:53:19.864Z	0x00000126	1		Valid license found for OPC UA IecVarAccess provider.
2021-04-06T07:53:19.865Z	0x00000002	1	7	Retains match to bootproject of application [<app>Application</app>]
2021-04-06T07:53:19.866Z	0x00000002	1	6	Bootproject of application [<app>Application</app>] loaded

Рисунок 163 – Два фильтра по категории (классу) сообщений в журнале ПЛК

6. Системные библиотеки

6.1. Общие сведения

Информация о системных библиотеках контроллеров Fastwel приведена в руководстве пользователя Fastwel PLC Application Toolkit. Настоящий раздел содержит информация о библиотеках, реализованных только в контроллере CPM810-03.

6.2. Библиотека доступа к порту дискретного ввода-вывода FastwelDIO

6.2.1. Общие сведения

Функции библиотеки FastwelDIO предназначены для программного управления 24-мя каналами дискретного ввода-вывода, выведенными на соединитель "DIO" на передней панели контроллера.

Каналы дискретного ввода-вывода сгруппированы в три порта по 8 каналов в каждом. Используется следующее соответствие между обозначениями портов дискретного ввода-вывода CPM810-03 и номерами портов, передаваемыми в качестве параметра *port* функциям *FwDIO_setDirection*, *FwDIO_getDirection*, *FwDIO_getPort*, *FwDIO_setPort*, *FwDIO_readPort*, *FwDIO_writePort*:

```
Port A 0
Port B 1
Port C 2
```

Используется следующее соответствие между обозначениями каналов дискретного ввода-вывода CPM810-03 и номерами каналов, передаваемыми в качестве параметра *line* функциям *FwDIO_getLineDirection*, *FwDIO_setLineDirection*, *FwDIO_getLine*, *FwDIO_setLine*, *FwDIO_readLine*, *FwDIO_writeLine*:

```
Port A Ch 0 0
Port A Ch 1 1
Port A Ch 2 2
Port A Ch 3 3
Port A Ch 4 4
Port A Ch 5 5
Port A Ch 6 6
Port A Ch 7 7
Port B Ch 0 8
Port B Ch 1 9
Port B Ch 2 10
Port B Ch 3 11
Port B Ch 4 12
Port B Ch 5 13
Port B Ch 6 14
Port B Ch 7 15
Port C Ch 0 16
Port C Ch 1 17
Port C Ch 2 18
Port C Ch 3 19
Port C Ch 4 20
Port C Ch 5 21
Port C Ch 6 22
Port C Ch 7 23
```

При запуске контроллера после включения питания все три порта дискретного ввода-вывода настроены на ввод, и, при отсутствии подключенных источников сигналов, чтение их состояния возвращает уровень логической 1 для всех каналов.

Перед началом работы с портами или каналами портов в приложении следует настроить каждый из трех портов на ввод или вывод путем вызова функции *FwDIO_setDirection*. При настройке порта на вывод возможно также определить начальное состояние выходных каналов порта. Обратите внимание, что функция *FwDIO_setDirection* может вызвана в любой момент времени выполнения приложения.

Для доступа к восьми каналам каждого порта или к отдельным каналам с использованием сквозной нумерации библиотека содержит наборы функций с префиксами *FwDIO_set* и *FwDIO_get*, а также *FwDIO_write* и *FwDIO_read*.

Функции с префиксом *FwDIO_set* и *FwDIO_get* возвращают статус выполнения операции, который представлен в библиотеке в виде набора констант:

```
VAR_GLOBAL CONSTANT
(* успех *)
FWDIO_RES_OK : INT := 0;
(* функция не поддерживается на данной платформе *)
FWDIO_RES_NOT_SUPPORTED : INT := -1;
(* неправильный номер порта *)
FWDIO_RES_INVALID_PORT : INT := -2;
(* неправильный номер канала *)
FWDIO_RES_INVALID_LINE : INT := FWDIO_RES_INVALID_PORT;
(* неправильный аргумент функции: нулевой указатель или код направления *)
FWDIO_RES_INVALID_ARG : INT := -4;
(* нет аппаратного доступа к данному порту или каналу *)
FWDIO_RES_ACCESS_ERROR : INT := -5;
END_VAR
```

Функции с префиксами *FwDIO_write* и *FwDIO_read* являются упрощенными вариантами аналогичных функций с префиксами *FwDIO_set* и *FwDIO_get* и предназначены для применения в случае, если при разработке приложения имеется полная уверенность в правильности передаваемых им параметров.

Все функции библиотеки FastwelDIO рассчитаны на работу в мультизадачном окружении, что обеспечивает возможность взаимоисключающего доступа к каналам и портам из разных задач приложения.

6.2.2. Функции

6.2.2.1. FwDIO_getPortsCount

Функция возвращает количество портов дискретного ввода-вывода, обслуживаемых библиотекой FastwelDIO на данной платформе.

```
FUNCTION FwDIO_getPortsCount : INT
VAR_INPUT
END_VAR
```

Входные параметры:

нет

Возвращаемый результат:

Функция возвращает:

3: на платформе CPM810-03 поддерживается три порта;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.2. FwDIO_getLinesCount

Функция возвращает количество каналов дискретного ввода-вывода у порта *port*.

```
FUNCTION FwDIO_getLinesCount : INT
VAR_INPUT
  port : INT;
END_VAR
```

Входные параметры:

port : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03 от 0 до 2.

Возвращаемый результат:

При правильном значении параметра *port* функция возвращает 8 на платформе CPM810-03.

Возвращаемые значения при неудачном завершении:

FWDIO_RES_INVALID_PORT (-2): значение параметра *port* менее 0 или более 2.

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.3. FwDIO_setDirection

Функция настраивает порт дискретного ввода-вывода *port* на ввод или на вывод.

```
FUNCTION FwDIO_setDirection : INT
VAR_INPUT
  port : INT;
  direction : INT;
  initial_value : BYTE;
END_VAR
```

Входные параметры:

port : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03: от 0 до 2.

direction : INT;

Направление каналов порта:

FWDIO_DIRECTION_INPUT (0) – порт настраивается на ввод;

FWDIO_DIRECTION_OUTPUT (1) – порт настраивается на вывод.

initial_value : BYTE

Данный параметр действителен только при настройке порта на вывод (*direction* := 1) и определяет начальное логическое состояние выходных каналов порта. Например, значение 16#00 позволяет установить состояние логического 0 на всех выходных каналах порта *port*. Меньшему номеру канала порта соответствует меньший номер бита значения *initial_value*.

Возвращаемый результат:

Возвращаемые значения:

FWDIO_RES_OK (0) : успех;

FWDIO_RES_INVALID_PORT (-2): значение параметра *port* менее 0 или более 2;

FWDIO_RES_INVALID_ARG (-4): значение параметра *direction* отлично от 0 и 1;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.4. FwDIO_getDirection

Функция возвращает текущий режим работы порта дискретного ввода-вывода *port*: ввод или вывод.

```
FUNCTION FwDIO_getDirection : INT
VAR_INPUT
  port : INT;
END_VAR
```

Входные параметры:

port : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03: от 0 до 2.

Возвращаемый результат:

Возвращаемые значения:

FWDIO_DIRECTION_INPUT (0): – порт настроен на ввод;

FWDIO_DIRECTION_OUTPUT (1): порт настроен на вывод.

FWDIO_RES_INVALID_PORT (-2): значение параметра *port* менее 0 или более 2;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.5. FwDIO_setLineDirection

Данная функция не поддерживается на платформе CPM810-03 и возвращает FWDIO_RES_NOT_SUPPORTED.

6.2.2.6. FwDIO_getLineDirection

Функция возвращает текущий режим работы канала дискретного ввода-вывода *line*: ввод или вывод.

```
FUNCTION FwDIO_getLineDirection : INT
VAR_INPUT
    line : INT;
END_VAR
```

Входные параметры:

line : INT

Номер канала дискретного ввода-вывода. На платформе CPM810-03: от 0 до 23.

Возвращаемый результат:

Возвращаемые значения:

FWDIO_DIRECTION_INPUT (0): – порт, которому принадлежит канал *line*, настроен на ввод;

FWDIO_DIRECTION_OUTPUT (1): порт, которому принадлежит канал *line*, настроен на вывод.

FWDIO_RES_INVALID_LINE (-3): значение параметра *line* менее 0 или более 23;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.7. FwDIO_getPort

При успешном завершении функция записывает текущее логическое состояние каналов порта дискретного ввода-вывода *port* по адресу *pValue*.

```
FUNCTION FwDIO_getPort : INT
VAR_INPUT
    port : INT;
    pValue : POINTER TO BYTE;
END_VAR
```

Входные параметры:

port : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03: от 0 до 2.

pValue : POINTER TO BYTE

Указатель на переменную типа BYTE, в которую будет записано текущее логическое состояние каналов порта дискретного ввода-вывода *port*. Меньшему номеру канала порта соответствует меньший номер бита значения, записанного по адресу *pValue*.

Возвращаемый результат:

FWDIO_RES_OK (0) : успех, логическое состояние каналов порта *port* записано по адресу *pValue*;

FWDIO_RES_INVALID_PORT (-2): значение параметра *port* менее 0 или более 2;

FWDIO_RES_INVALID_ARG (-4): *pValue* содержит нулевое значение;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.8. FwDIO_readPort

Функция возвращает текущее логическое состояние каналов порта дискретного ввода-вывода *port*.

```
FUNCTION FwDIO_readPort : BYTE
VAR_INPUT
    port : INT;
END_VAR
```

Входные параметры:**port** : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03: от 0 до 2. Меньшему номеру канала порта соответствует меньший номер бита возвращаемого значения.

Возвращаемый результат:

Если функции передано значение *port* от 0 до 2 включительно и имеется аппаратный доступ к портам дискретного ввода-вывода, функция возвращает логическое состояние восьми каналов порта *port*. Меньшему номеру канала порта соответствует меньший номер бита возвращаемого значения.

Если функции передано значение *port* от 0 до 2 включительно и нет доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS, функция возвращает 0.

Если функции передано значение *port* менее 0 или более 2, то функция возвращает 0.

6.2.2.9. FwDIO_setPort

При успешном завершении функция устанавливает логическое состояние *value* на каналах порта *port*, настроенного на вывод.

```
FUNCTION FwDIO_setPort : INT
VAR_INPUT
  port : INT;
  value : BYTE;
END_VAR
```

Входные параметры:**port** : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03: от 0 до 2.

value : BYTE

Логическое состояние, которое должно быть установлено на каналах порта дискретного ввода-вывода *port*, настроенного на вывод. Меньшему номеру канала порта соответствует меньший номер бита передаваемого значения *value*.

Возвращаемый результат:

FWDIO_RES_OK (0) : успех, логическое состояние каналов порта *port* записано по адресу *pValue*;

FWDIO_RES_INVALID_PORT (-2): значение параметра *port* менее 0 или более 2;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.10. FwDIO_writePort

Функция устанавливает логическое состояние *value* на каналах порта *port*, настроенного на вывод.

```
FUNCTION FwDIO_writePort : INT
VAR_INPUT
  port : INT;
  value : BYTE;
END_VAR
```

Входные параметры:**port** : INT

Номер порта дискретного ввода-вывода. На платформе CPM810-03: от 0 до 2.

value : BYTE

Логическое состояние, которое должно быть установлено на каналах порта дискретного ввода-вывода *port*, настроенного на вывод. Меньшему номеру канала порта соответствует меньший номер бита передаваемого значения *value*.

Возвращаемый результат:

Возвращает значение параметра *value*.

6.2.2.11. FwDIO_getLine

При успешном завершении функция записывает текущее логическое состояние канала дискретного ввода-вывода *line* по адресу *pValue*.

```
FUNCTION FwDIO_getLine : INT
VAR_INPUT
  line : INT;
  pValue : POINTER TO BOOL;
END_VAR
```

Входные параметры:

line : INT

Номер канала дискретного ввода-вывода. На платформе CPM810-03: от 0 до 23.

pValue : POINTER TO BOOL

Указатель на переменную типа BOOL, в которую будет записано текущее логическое состояние канала дискретного ввода-вывода *line*. Значению TRUE соответствует уровень логической 1, значению FALSE – уровень логического 0.

Возвращаемый результат:

FWDIO_RES_OK (0) : успех, логическое состояние канала *line* записано по адресу *pValue*;

FWDIO_RES_INVALID_LINE (-3): значение параметра *line* менее 0 или более 23;

FWDIO_RES_INVALID_ARG (-4): *pValue* содержит нулевое значение;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.12. FwDIO_readLine

Функция возвращает текущее логическое состояние канала дискретного ввода-вывода *line*.

```
FUNCTION FwDIO_readLine : BYTE
VAR_INPUT
  line : INT;
END_VAR
```

Входные параметры:

line : INT

Номер канала дискретного ввода-вывода. На платформе CPM810-03: от 0 до 23.

Возвращаемый результат:

Если функции передано значение *line* от 0 до 23 включительно и имеется аппаратный доступ к портам дискретного ввода-вывода, функция возвращает TRUE, если канал *line* находится в состоянии логической 1, и FALSE – если канал *line* находится в состоянии логического 0.

Если функции передано значение *line* от 0 до 23 включительно и нет доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS, функция возвращает FALSE.

Если функции передано значение *port* менее 0 или более 2, то функция возвращает FALSE.

6.2.2.13. FwDIO_setLine

При успешном завершении функция устанавливает логическое состояние *value* на канале дискретного ввода-вывода *line*, ранее настроенном на вывод. Логическое состояние остальных каналов порта дискретного ввода-вывода, которому принадлежит *line*, остается неизменным.

```
FUNCTION FwDIO_setLine : INT
VAR_INPUT
  line : INT;
  value : BOOL;
END_VAR
```

Входные параметры:

line : INT

Номер канала дискретного ввода-вывода. На платформе CPM810-03: от 0 до 23.

value : BOOL

Новое логическое состояние, которое будет установлено на канале дискретного ввода-вывода *line*. Значению TRUE соответствует уровень логической 1, значению FALSE – уровень логического 0.

Возвращаемый результат:

FWDIO_RES_OK (0) : успех, на канале *line* установлено логическое состояние *value*;

FWDIO_RES_INVALID_LINE (-3): значение параметра *line* менее 0 или более 23;

FWDIO_RES_ACCESS_ERROR (-5): нет аппаратного доступа к портам дискретного ввода-вывода из-за неправильной настройки BIOS.

6.2.2.14. FwDIO_writeLine

Функция устанавливает логическое состояние *value* на канале *line* порта дискретного ввода-вывода, ранее настроенного на вывод. Логическое состояние остальных каналов порта дискретного ввода-вывода, которому принадлежит *line*, остается неизменным.

```
FUNCTION FwDIO_writeLine : INT
VAR_INPUT
  line : INT;
  value : BOOL;
END_VAR
```

Входные параметры:

line : INT

Номер канала дискретного ввода-вывода. На платформе CPM810-03: от 0 до 23.

value : BOOL

Новое логическое состояние, которое будет установлено на канале дискретного ввода-вывода *line*. Значению TRUE соответствует уровень логической 1, значению FALSE – уровень логического 0.

Возвращаемый результат:

Возвращает значение параметра *value*.

ПРИЛОЖЕНИЕ 1. ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Версия	Дата	Ссылка	Статус	Примечания
0.1	23.12.2019	Документ	создан	Предварительный выпуск
0.2	28.01.2020	Документ	изменен	Предварительный выпуск, заданы реквизиты
1.0	14.08.2020	п. 2.4.4	изменен	Уточнены характеристики производительности системы исполнения и размеры сегментов памяти приложения
		п. 5.4.1.6	изменен	Добавлена дополнительная информация об особенностях жизненного цикла энергонезависимых переменных
		п. 5.4.7	создан	Добавлено описание сервера OPC Unified Architecture (начиная с версии 3.0.x.x системного ПО контроллера и версии 3.255.0.0 описания целевой платформы)
		п. 6.4	изменен	Добавлена информация о включении и отключении поддержки взаимодействия контроллера со средой разработки через порты COM1 или COM2 на странице Маршрутизация CODESYS .
		п. 6.10	изменен	Добавлено описание страницы Параметры OPC UA (начиная с версии 3.0.x.x системного ПО контроллера)
		п. 6.11	изменен	Добавлено описание страницы Сертификаты безопасности (начиная с версии 3.0.x.x системного ПО контроллера).
		п. 6.12	создан	Добавлено описание страницы Видео .
		п. 6.16.2	создан	Добавлено описание операции включения и выключения поддержки устройств USB.
		п. 6.16.4	изменен	Добавлена информация о состоянии цифровых сертификатов после восстановления системных параметров контроллера
		п. 7.7.2 –7.7.3	изменены	Уточнено значение параметра ulTimeout функции SysComRead и структуры COM_Settings при работе со встроенными последовательными портами.
п. 8.5.2.1	изменен	Добавлено описание параметра cert=all для команды savearr (начиная с версии 3.0.x.x системного ПО контроллера).		
1.1	05.07.2021	Документ	изменен	Устранены дефекты оформления
		п. 3.5.2, п. 3.5.3.3	изменен	Добавлена информация о правиле формирования исходного имени контроллера, произведенного начиная с 2021 г.
		п. 5.2.4	создан	Добавлена информация о чтении текущего режима контроллера по сети (начиная с версии системного ПО 3.2.3.0).
		п. 5.4.7.5	изменен	Добавлена информация о чтении текущего режима контроллера через OPC UA (начиная с версии системного ПО 3.2.3.0).
		п. 5.6.2.4	изменен	Добавлена информация о назначении адреса шлюза по умолчанию (Default Gateway)
		п. 6.10	изменен	Добавлено описание параметра Имя хоста в качестве имени сервера на странице Параметры OPC UA
		п. 6.17	создан	Добавлено описание страницы веб-конфигуратора Настройка путей (начиная с версии системного ПО 3.1.x.x)
		п. 6.18	создан	Добавлено описание страницы веб-конфигуратора Журнал ПЛК (начиная с версии системного ПО 3.2.x.x)
		п. 7.8	создан	Добавлено описание системной библиотеки FastwelRemovableMedia (с версии 3.2.x.x системного ПО контроллера, с версии 1.0.7.0 пакета адаптации CODESYS V3 для контроллеров Fastwel).
		п. 8.2	создан	Добавлено описание команд disks и eject для просмотра информации и отключения USB-накопителей.
1.2	07.04.2022	п. 2.5.2	изменен	Расширен состав пакета адаптации CODESYS V3 для контроллеров Fastwel.
		п. 3.4	изменен	Заменен рисунок 9.
		п. 5.4.7.2	изменен	Добавлена информация о поле Имя сервера (начиная с версии системного ПО 3.3.4.0).

Версия	Дата	Ссылка	Статус	Примечания
		п. 5.5.2.1, п. 5.5.3, п. 5.5.4.1, п. 5.5.4.2, п. 5.5.5.1, п. 5.5.5.2	изменен	Добавлено описание параметров мастера шины FBUS Дополнительные параметры – Допустимое количество ошибок и Дополнительные параметры – Период обслуживания сети, мс. (начиная с версии 1.0.7.0 пакета адаптации CODESYS V3 для контроллеров Fastwel
		п. 6.10	изменен	Добавлена информация о поле Имя сервера (начиная с версии системного ПО 3.3.4.0).
		п. 8.7	создан	Добавлена информация об оценке загрузки процессора при помощи команды topservice оболочки ПЛК (начиная с версии системного ПО 3.3.4.0).
2.0	22.09.2023	Документ	изменен	Информация о программной модели, системных библиотеках и сервисных операциях перенесена в документ <i>ИМЕС.00390-03 33 01. Пакет инструментальных средств ПЛК Fastwel. Fastwel PLC Application Toolkit. Руководство пользователя.</i> Добавлена информация о поддержке протоколов HTTPS, FTPS, SFTP.
		п. 5.12	изменен	Добавлена информация о способе переключения языка ввода с английского на русский и наоборот при использовании клавиатуры