

Программное обеспечение «Просуп»

Руководство программиста

Содержание

1. Введение	5
2. Подготовка к работе с изделием	6
2.1. Документация	6
3. Настройка и диагностика модулей CPU МК-505-120 и МК-545-010	7
3.1. Общие сведения о работе процессорных модулей изделия	7
3.2. Инструментальное программное обеспечение Fallover	8
3.2.1. Системные требования	8
3.3. Подключение к модулю CPU с помощью Fallover	8
3.4. Настройка и обновление модулей	10
3.4.1. Сетевые настройки	11
3.4.2. Настройки времени	14
3.4.3. Приложения	15
3.4.4. Логирование во внешнее приложение	15
3.4.5. Программа пользователя	17
3.4.6. Настройки безопасности	18
3.4.7. Настройки шифрования	20
3.4.8. Подключение и обновление модуля CN (МК-545-010) с помощью Fallover	22
3.4.9. Обновление модуля CPU с помощью Fallover	22
3.4.10. Обновление модуля CPU с SD-карты	23
4. Работа в среде разработки Procyon IDE	24
4.1. Системные требования	24
4.2. Установка и настройка среды разработки Procyon IDE	24
4.1.1. Установка среды разработки Procyon IDE	24
4.1.2. Установка пакета поддержки контроллера MKLogic-500	24
4.2. Создание проекта MKLogic-500 в Procyon IDE	25
4.3. Структура проекта (Дерево устройств)	26
4.3.1. Дерево устройств	26
4.3.2. Правила и механизмы расположения и конфигурации объектов в дереве устройств	27
4.3.3. Дочерние объекты «Приложения»	31
4.3.4. Дерево устройств в режиме онлайн	31
4.4. Генерация прикладного кода	32
4.4.1. Установка среды разработки Procyon IDE	32
4.4.2. Сообщения при генерации прикладного кода	32
4.5. Загрузка проекта в контроллер	33
4.5.1. Настройка соединения с контроллером	33
4.5.2. Загрузка прикладного кода — Логин и запуск ПЛК	34
5. Работа с модулями MKLogic-500	35
5.1. Общие принципы работы с устройствами модулей ввода-вывода	35
5.1.1. Канал диагностической информации модулей	37
5.2. Общие принципы работы с дополнительными устройствами	38
5.3. Модули питания МК-550-024	38
5.4. Модули центрального процессора МК-504-120	39
5.4.1. Коммуникационный модуль МК-546-010	41
5.4.2. Коммуникационный модуль МК-544-040	41
5.5. Модули аналогового ввода МК-513-016 и МК-513-016А	42
5.6. Модули аналогового вывода МК-514-008 и МК-514-008А	43
5.7. Модули аналогового ввода МК-516-008 и МК-516-008А	44
5.8. Модули дискретного ввода МК-521-032 и МК-521-032А	44
5.9. Модули дискретного вывода МК-531-032 и МК-531-032А	45
5.10. Модули дискретного ввода МК-523-032А	46
5.11. Модули дискретного вывода МК-532-032А	47
5.12. Коммуникационные модули МК-541-002	48
5.12.1. Инициализация и передача команд в модуль МК-541-002	51
5.12.2. Особенности работы Modbus-функций 1, 2, 3 и 4 в модуле МК-541-002	52
5.12.3. Особенности работы Modbus-функций 1, 2, 5 и 15 в модуле МК-541-002	52
5.13. Коммуникационные модули МК-545-010	52
5.14. Модули аналогового вывода МК-574-008А с протоколом HART	52
5.15. Модули аналогового ввода МК-576-008А с протоколом HART	54
5.16. Модули аналогового ввода МК-576-016А с протоколом HART	55
5.17. Дистанционные стойки, подключенные по протоколу POWERLINK	55
6. Реализация резервирования в MKLogic-500	57
6.1. Термины и определения	57

6.1.1. Роли Primary и Secondary	57
6.1.2. Роли Active и Passive	57
6.1.3. Состояние Standalone	57
6.1.4. Области синхронизируемых данных	57
6.2. Ограничения при использовании резервирования	57
6.3. Использование резервирования	57
6.4. Библиотека NftRedundancy	58
6.4.1. Функция AreaRegister	58
6.4.2. Функция SwapActiveCPU	59
6.5. Настройка областей синхронизации резервирования	59
6.6. Использование скрипта redundancyAreas для автоматизации настройки резервирования	61
7. Работа с коммуникационными протоколами MKLogic-500	62
7.1. Протокол Modbus	62
7.1.1. Режим “ведомого” в протоколе Modbus	62
7.1.2. Режим “ведущий” в протоколе Modbus	67
7.2. Протокол IEC 60870-5-104 (сервер)	73
7.2.1. Режим ведомого в протоколе IEC 60870-5-104	74
8. Использование дополнительных библиотек MKLogic-500	79
8.1. Библиотека NftSys	79
8.1.1. Функция FTPEnable	79
8.1.2. Функция StoreRetainsInFile	79
8.1.3. Функция CreateHistoricalDataFile	79
8.1.4. Функция StoreHistoricalFile	80
8.1.5. Функция ReadRealFromFile	80
8.1.6. Функция ReadReal2DimArray	81
8.1.7. Функция WriteRealToFile	81
8.1.8. Функция WriteReal2DimArray	82
8.2. Библиотека NftSpecial	82
8.2.1. Функция UdpMessage	82
8.3. Библиотека DiagMK-500	83
8.3.1. Функциональный блок для диагностики модулей без горячего резервирования ПЛК	83
8.3.2. Функциональный блок для диагностики модулей с горячим резервированием ПЛК	94
8.3.3. Работа с библиотекой в проекте	97

Ни одна из частей данного документа не может быть воспроизведена или передана по каналам связи в любой форме или любыми средствами без предварительного письменного согласия АО «НЕФТЕАВТОМАТИКА».

Товарный знак является зарегистрированным товарным знаком АО «НЕФТЕАВТОМАТИКА».

 **ПРИМЕЧАНИЕ**

АО «НЕФТЕАВТОМАТИКА» сохраняет за собой право вносить изменения или обновлять данные и технические характеристики в данном руководстве без предварительного уведомления и каких-либо обязательств при условии, что такие изменения не ухудшают технические характеристики изделия, а являются результатом работ по усовершенствованию его конструкции или технологии производства.

 **Импортные и экспортные правил**

Заказчики АО «НЕФТЕАВТОМАТИКА» должны соблюдать все законы и инструкции по экспорту и импорту. Они должны заранее получить необходимые разрешения и лицензии, касающиеся экспорта, реэкспорта или импорта на указанное в данном руководстве изделие, включая программные продукты и технические данные в них.

Адрес: 450005, Россия, Уфа, 50-летия Октября, 24

Тел.: +7 (347) 279-88-99, 8-800-700-78-68

Факс: 8-800-700-78-68

Веб-сайт: <http://www.nefteavtomatika.ru>

Эл. почта: nefteavtomatika@nefteavtomatika.ru

1. Введение

Настоящее руководство по программированию (далее – РП) содержит сведения, необходимые для конфигурирования и программирования изделия Контроллер программируемый логический МКLogic-500 (далее – изделие) на языках МЭК-61131-3 специалистами АСУТП в среде Procyon IDE.

В РП приведены сведения об установке и настройке среды разработки, конфигурировании изделия и особенностях работы с модулями ввода-вывода и коммуникационными модулями при написании технологических программ в среде разработки.

Настоящее РП распространяется на изделие и применимо при использовании модуля центрального процессора МК-505-120.

Конфигурирование и программирование изделия должно осуществляться специально обученным и изучившим настоящее РП обслуживающим персоналом.

В РП приняты следующие условные обозначения:

Таблица 1.1 – Условные обозначения

Обозначение	Комментарий
 ПРИМЕЧАНИЕ	Дополнительные сведения и указания
 ВНИМАНИЕ	Информация, на которую следует обратить особое внимание
Специальный шрифт	Названия функций, параметров, переменных, папок, документов, листинги примеров
Выделение цветом	Ссылки на разделы руководства, рисунки и таблицы
Текст курсивом	Названия пунктов меню, кнопок

2. Подготовка к работе с изделием

2.1. Документация

Перед началом работы с изделием следует обязательно ознакомиться со следующими документами:

- КДСА.426471.004 РЭ_2.0_00 Контроллер программируемый логический МКLogic-500 Руководство по эксплуатации;
- Спецификация прикладного уровня и коммуникационного профиля CANopen CiA 301 (CANopen_CiA_301_spec_rus);
- Procyon IDE (Procyon IDE.chm);

Все вышеперечисленные документы поставляются вместе со средой разработки Procyon IDE и располагаются на диске в папке Online Help (здесь и далее, все пути и имена файлов приводятся относительно папки с установочным комплектом Procyon IDE).

3. Настройка и диагностика модулей CPU МК-505-120 и МК-545-010

3.1. Общие сведения о работе процессорных модулей изделия

Модули центрального процессора изделия (далее модули CPU) в первую очередь предназначены для работы в парах, с поддержкой горячего резервирования.

При работе в режиме резервирования один из модулей CPU всегда является активным (находится в режиме Primary), второй же является резервным (находится в режиме Secondary). В режиме резервирования модули CPU кроме исполнения программы пользователя выполняют синхронизацию настроек при старте, синхронизацию данных в ходе работы, автоматическое переключение на резервный модуль CPU при отказе активного.

Необходимым условием корректной работы модулей CPU в паре является связь модулей CPU и по шине CAN, и по шине Ethernet (через сетевые интерфейсы в роли Datalink IP). Отсутствие связи по шине CAN приводит к раздельному запуску модулей CPU, как будто они оба запущены без резервирования. Отсутствие связи по шине Ethernet приводит к запуску модулей CPU в паре, но без синхронизации настроек и данных при старте и в ходе работы.

! **ВНИМАНИЕ** Работа модулей CPU в паре, но без связи и по шине CAN и по шине Ethernet – недопустима.

Переключение на резервный модуль CPU происходит при наступлении следующих событий:

- выход из строя (выключение) активного модуля CPU либо его программного обеспечения;
- одновременное пропадание/отключение на активном модуле CPU интерфейсов CAN, и Ethernet на сетевом интерфейсе в роли Datalink IP;
- пропадание связи на сетевом интерфейсе в роли Uplink IP активного CPU при наличии связи на соответствующем сетевом интерфейсе резервного CPU;
- полное пропадание связи с остальными устройствами на шине Powerlink активного модуля CPU при наличии связи с устройствами Powerlink у резервного модуля CPU.

! **ВНИМАНИЕ** При переключении между активным и резервным модулями CPU смены IP-адресов не происходит.

При старте модуль CPU с включённым режимом резервирования ищет второй модуль CPU по шине CAN. Если найти второй модуль CPU не удаётся, дальнейшая работа выполняется с сетевыми настройками предыдущего сеанса и в роли Primary. При этом запуск программы пользователя сопровождается двухсекундным миганием всей индикации модуля CPU.

! **ВНИМАНИЕ** Сетевые настройки модуль CPU применяет строго на основании своего положения на шине CAN. Если модуль CPU находится сразу после блока питания с CAN-адресом 1, он считается модулем CPU PR и использует сетевые настройки для варианта PR, во всех остальных случаях модуль CPU использует сетевые настройки для варианта SE.

Если запускаемый модуль CPU при старте обнаруживает второй модуль CPU, выполняются следующие операции:

- модули CPU обмениваются текущими состояниями. Если обнаруженный модуль CPU уже в работе (запущена программа пользователя), то запускаемый модуль назначает себя на время старта ведомым. Если обнаруженный модуль ещё не в работе, то ведомым признаётся модуль CPU с большим адресом на шине CAN. Настройки ведущего модуля CPU имеют приоритет над настройками ведомого модуля;
- синхронизируются роли сетевых интерфейсов модулей CPU, от ведущего модуля CPU к ведомому;

— после синхронизации ролей сетевых интерфейсов синхронизируются сетевые настройки модулей, от ведущего модуля CPU к ведомому. Запускаемый модуль CPU применяет к себе сетевые настройки согласно своему положению на шине CAN (PR или SE), независимо от того ведущий он или ведомый. Также ведомый модуль применяет к себе настройки NTP-сервера ведущего модуля CPU;

— после синхронизации сетевых интерфейсов выполняется выбор программы пользователя. Принцип выбора: если программа пользователя есть на обоих модулях CPU, ведомый модуль CPU удаляет свой вариант. Если программа пользователя есть только на ведомом CPU, в этом случае ведущий копирует себе программу пользователя. Такое поведение защищает от ситуации, когда модуль с программой пользователя случайно запускается после запуска резервного модуля без программы пользователя;

— также после синхронизации сетевых интерфейсов копируется содержимое папки FTP ведущего модуля CPU в папку FTP ведомого. Не копируются файлы обновлений, файлы журналов обновлений и диагностические журналы;

— также после синхронизации сетевых интерфейсов от ведущего модуля CPU к ведомому копируется текущая конфигурация сети Powerlink (для модулей CPU с поддержкой работы с Powerlink) и ключи шифрования OPC UA;

— запускаемый модуль CPU передаёт управление Procyon IDE и переходит в штатный режим работы.

Таким образом, рекомендованная последовательность запуска модулей CPU такая:

— модули настроены одинаково или имеют заводские настройки, то произвольная последовательность;

— один модуль настроен корректно и несёт в себе актуальную программу пользователя, а другой модуль взят из ЗИПа – первым следует запустить модуль с корректными настройками/программой.

3.2. Инструментальное программное обеспечение Fallover

Модули CPU выпускаются с предустановленными сетевыми настройками. Для установки необходимых сетевых настроек, а также для настройки временных параметров и для выполнения диагностики следует использовать бесплатно распространяемое приложение Fallover.exe. Его можно скачать с сайта разработчиков АО “Нефтеавтоматика”.

В функции программы входит:

- сбор диагностических данных;
- доступ и обновление по FTP;
- настройка сетевых интерфейсов и маршрутов;
- настройка параметров времени и NTP-сервера;
- диагностика работы системного ПО и настройка параметров логирования;
- работа с журналом сообщений среды исполнения Procyon IDE и удаление программы пользователя;
- настройка шифрованного обмена по протоколу OPC UA;
- управление параметрами авторизации для работы с функциями настройки, диагностики, а также доступа и обновления по FTP.

3.2.1. Системные требования

Для работы программы требуется установка следующих компонентов:

- платформа **.NET Framework** версии 4.7.2 и выше;
- библиотека **NPcap** (версии 1.6.0 и выше) либо **WinPcap** (версии 4.1.3 и выше).

3.3. Подключение к модулю CPU с помощью Fallover

Для начала настройки модуля CPU следует выполнить следующие операции:

1) С помощью сетевого кабеля подключить порт ETH1 настраиваемого модуля CPU непосредственно к порту рабочего ПК, либо к общему с рабочим ПК сетевому коммутатору. Расположение портов CPU можно посмотреть в разделе [Модули центрального процессора МК-505-120](#)(см. Рис. 5.9, 5.10).

2) В окне приложения Fallover, выбрать IP-адрес сетевого интерфейса, к которому подключён настраиваемый модуль CPU, затем нажать кнопку «Поиск устройства». В течение 10 секунд будет выполняться сканирование сети с целью найти и опознать все доступные модули центрального процессора изделия. По окончании поиска таблица будет заполнена перечнем модулей CPU, доступных для настройки и диагностики (Рис. 3.1). В таблицу выводится следующая информация для каждого обнаруженного модуля CPU:

- наименование типа модуля CPU;
- IP-адрес+шлюз для сетевого интерфейса в роли IP;
- IP-адрес+шлюз для сетевого интерфейса в роли Datalink IP;
- серийный номер модуля CPU;
- имя ресурса выполняемой в модуле CPU программы пользователя;
- поддержка модулем CPU режима резервирования;
- Поддержка модулем CPU протокола Powerlink.

Устройство	IP адрес	Datalink адрес	Серийный №	Ресурс	Резерв	Powerlink
MKLogic500 CPU 504	192.168.202.1 0.0.0.0	192.168.111.215 0.0.0.0	2200015	Application	Да	Да

Рис. 3.1 - Окно приложения Fallover с найденными модулями CPU

3) Если поиск не дал результатов, следует провести поиск модуля по IP-адресу. Для этого в окне приложения следует нажать кнопку «Поиск по IP», в открывшемся окне ввести IP-адрес порта ETН1 желаемого модуля и нажать кнопку «ОК» (Рис. 3.2). Все IP-адреса модуля CPU бегущей строкой показываются на панели индикации. По окончании поиска в таблице появится модуль (будет подсвечен серым) доступный для настройки и диагностики (Рис. 3.3).

Рис. 3.2 - Окна ввода IP-адреса модуля CPU

Устройство	IP адрес	Datalink адрес	Серийный №	Ресурс	Резерв	Powerlink
MKLogic500 CPU 504	192.168.202.1 0.0.0.0	192.168.111.215 0.0.0.0	2200015	Application	Да	Да

Рис. 3.3 - Окно приложения Fallover с найденным модулем CPU

4) При выборе в таблице строки модуля и нажатии на кнопку «Подсветить» модуль в течение 5 секунд мигает всеми индикаторами лицевой панели. Это позволяет идентифицировать модуль в случае, когда сетевые настройки модулей CPU не позволяют этого сделать.

5) При необходимости записать в модуль файлы настройки либо файлы обновлений следует выбрать в таблице строку с модулем и нажать кнопку «Доступ по FTP». В открывшееся окно проводника можно скопировать необходимые файлы (Рис. 3.4).

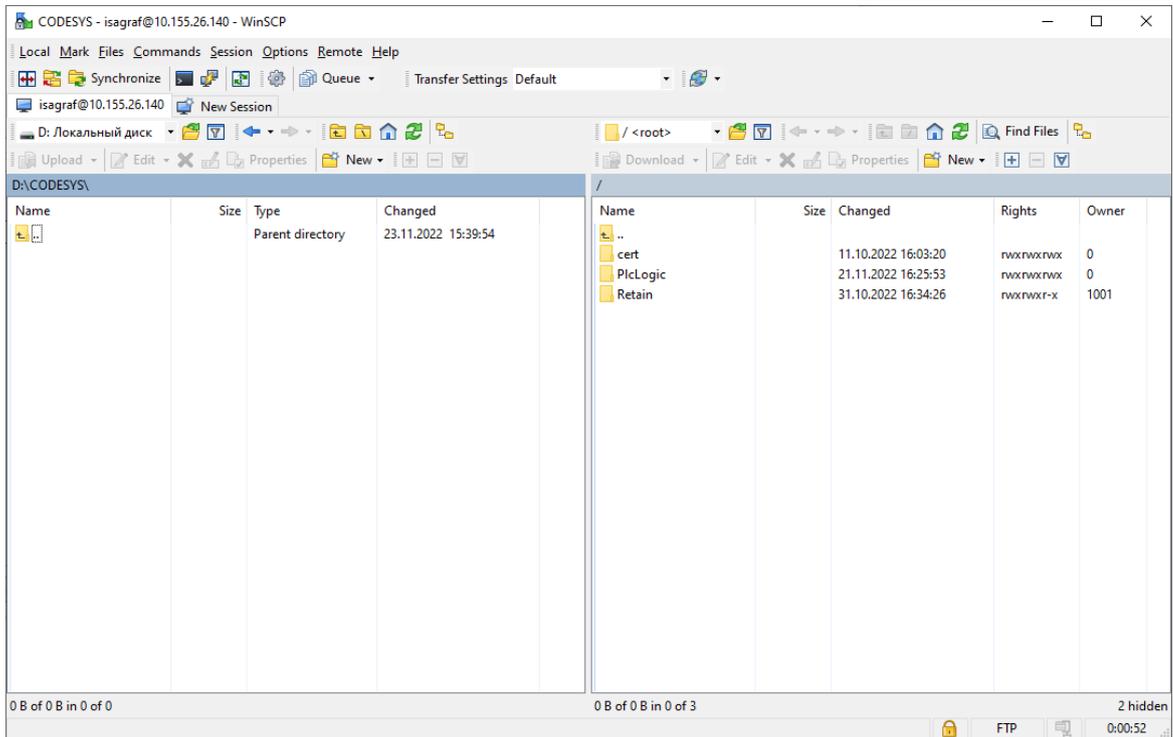


Рис. 3.4 - Окно проводника в режиме доступа по FTP к модулю CPU

6) Для выполнения непосредственно настройки и/или диагностики модуля CPU следует выбрать в таблице строку с модулем и нажать кнопку «Настройки», либо дважды кликнуть на этой строке.

7) Для формирования пакета логов, передаваемых непосредственно производителю, выбрать в таблице строку с модулем и нажать кнопку «Диагностика». Спустя некоторое время сформированный пакет логов будет помещен в каталог “/result”. Далее этот пакет передается в службу технической поддержки производителя оборудования.

3.4. Настройка и обновление модулей

Для выполнения непосредственно настройки и/или диагностики модуля CPU следует выбрать в таблице строку с модулем и нажать кнопку «Настройки» либо дважды кликнуть на этой строке. Открывшееся окно настроек модуля CPU имеет 5 разделов, открывающихся при нажатии соответствующего пункта списка в левой части окна (Рис. 3.5). Общими для всех разделов являются кнопки «Прочитать из CPU» и «Закрыть» в нижнем правом углу окна. Кнопку «Прочитать из CPU» следует использовать для обновления значений в текущем разделе окна настроек. Заголовок окна содержит тип CPU и его серийный номер.

Раздел «Сетевые настройки» предназначен для настройки сетевых интерфейсов и маршрутов модуля CPU для режимов работы Primary и Secondary, а также для назначения ролей сетевым интерфейсам.

Раздел «Настройки времени» предназначен для настройки времени и часового пояса часов реального времени модуля CPU, а также для настройки NTP-сервера модуля CPU.

Раздел «Приложения» предназначен для диагностики работы системного программного обеспечения модуля CPU, для перезапуска системного программного обеспечения модуля CPU в случае непредвиденного отказа, для запуска диагностического web-сервера в модуле CPU и для управления диагностическими сообщениями системного программного обеспечения.

Раздел «Программа пользователя» предназначен для работы с журналом сообщений системы исполнения Prosuon IDE (данная функция зарезервирована, но пока не реализована), а также для удаления программы пользователя с модуля CPU.

Раздел «Настройки шифрования» предназначен для настройки шифрованного обмена по протоколу OPC UA и поддерживается только на модулях CPU МК- 504- 120 (пока не поддерживается).

Раздел «Настройки безопасности» предназначен для настройки парольного доступа модулей CPU.

Далее по тексту будут более подробно разобраны особенности работы во всех пяти разделах.

3.4.1. Сетевые настройки

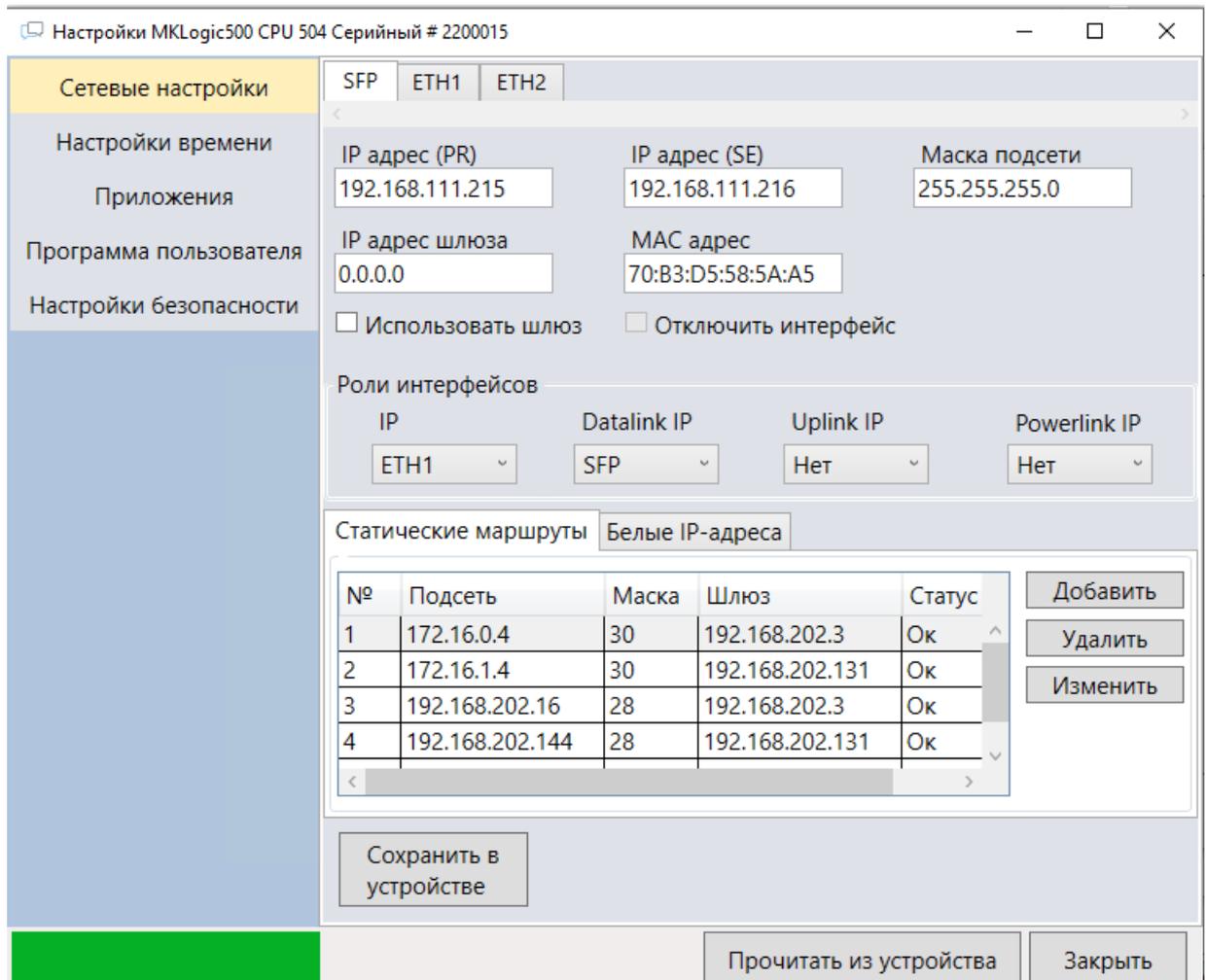


Рис. 3.5 - Окно сетевых настроек модуля CPU

В верхней части окна настроек модуля CPU в разделе «Сетевые настройки» (Рис. 3.5) расположены вкладки с параметрами сетевых интерфейсов модуля CPU. Каждый интерфейс настраивается в двух вариантах – для модуля CPU в позиции PR (сразу после блока питания с CAN-адресом 1) и для модуля CPU во всех остальных позициях (SE). Два варианта настроек нужны для того, чтобы модуль CPU мог передать корректные сетевые настройки резервному модулю CPU без его предварительной настройки.

При необходимости, можно отключить сетевой(-ые) интерфейс(-ы). Для этого нужно:

- в окне «Сетевые настройки» выбрать интересующий интерфейс;
- нажать кнопку «Отключить интерфейс»;

— далее произойдет перезапуск устройства CPU (при этом необходимо учитывать, что на резервированной сборке, после перезапуска, устройство CPU примет настройки от текущего активного устройства CPU).

ВНИМАНИЕ Отключенному (-ым) интерфейсу (-ам) невозможно назначить «Роль интерфейса».

ВНИМАНИЕ Все сетевые интерфейсы модуля CPU должны находиться в разных подсетях. Нарушение этой рекомендации при работе с разными интерфейсами в одной подсети приводит к потерям пакетов и к постоянным обрывам связи.

Маска подсети (обязательно) и шлюз (при необходимости) настраиваются одинаково для обоих вариантов (PR и SE) интерфейса.

ВНИМАНИЕ Допускается использование шлюза только на одном сетевом интерфейсе. Нарушение этого требования приводит к потерям пакетов и невозможности устойчиво работать с другими подсетями. Для более сложных сетевых топологий следует использовать статические маршруты.

В средней части окна расположены выпадающие списки для задания ролей сетевых интерфейсов модуля CPU. Необходимо выбрать интерфейсы для роли IP-интерфейса (для связи со средой разработки Procyon IDE) и для роли Datalink IP-интерфейса (связь с резервным модулем CPU в режиме резервирования).

Роль Uplink IP назначается на сетевой интерфейс, который должен постоянно быть на связи (например, если через этот интерфейс выполняется обмен данными с модулем CPU по Modbus TCP). При потере связи Uplink-интерфейсом модуля CPU в режиме Primary - происходит переключение на Secondary CPU (при условии, что Uplink-интерфейс Secondary CPU на связи). Роль Uplink IP является опциональной.

Роль Powerlink IP интерфейса реализована для CPU МК-504-120 и МК-545-010 (у МК-545-010 роль Powerlink IP настроена по умолчанию). Назначение сетевого интерфейса на роль Powerlink IP необходимо для корректной работы Powerlink.

Под настройками ролей расположено окно настройки статических маршрутов и белых IP-адресов модуля CPU. Статические маршруты настраиваются сразу для всех сетевых интерфейсов модуля. При нажатии кнопки «Добавить» появляется окно настроек параметров маршрута (Рис. 3.6).

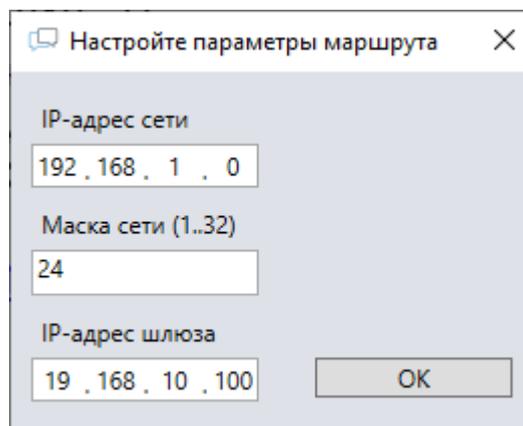


Рис. 3.6 - Окно настроек параметров маршрута модуля CPU

Конечная точка определяет конечную точку маршрута. Конечной точкой может быть сетевой IP-адрес (где разряды узла в сетевом адресе имеют значение 0), IP-адрес маршрута к узлу, или значение 0.0.0.0 для маршрута по умолчанию. mask маска_сети Указывает маску сети (также известной как маска подсети) в соответствии с точкой назначения. Маска сети может быть маской подсети соответствующей сетевому IP-адресу, например 255.255.255.255 для маршрута к узлу или 0.0.0.0. для маршрута по умолчанию. Если данный параметр пропущен, используется маска подсети 255.255.255.255. Новые маршруты запишутся в модуль CPU при сохранении всех сетевых настроек. Кнопка «Сохранить в устройстве» служит для применения введенных настроек сетевых интерфейсов, ролей сетевых интерфейсов и маршрутов в модуле CPU.

ВНИМАНИЕ Нажатие кнопки «Сохранить в устройстве» приводит к перезапуску системного ПО модуля CPU с остановкой и перезапуском программы пользователя, и требует подтверждения в диалоговом окне. При работе модуля CPU в паре рестарт системного ПО приведёт к копированию сетевых настроек из второго модуля CPU (см. раздел «Общие сведения о работе процессорных модулей изделия»). Поэтому для успешного применения сетевых настроек следует отключить второй модуль CPU.

Белые IP-адреса настраиваются отдельно для каждого интерфейса модуля (Рис. 3.7).

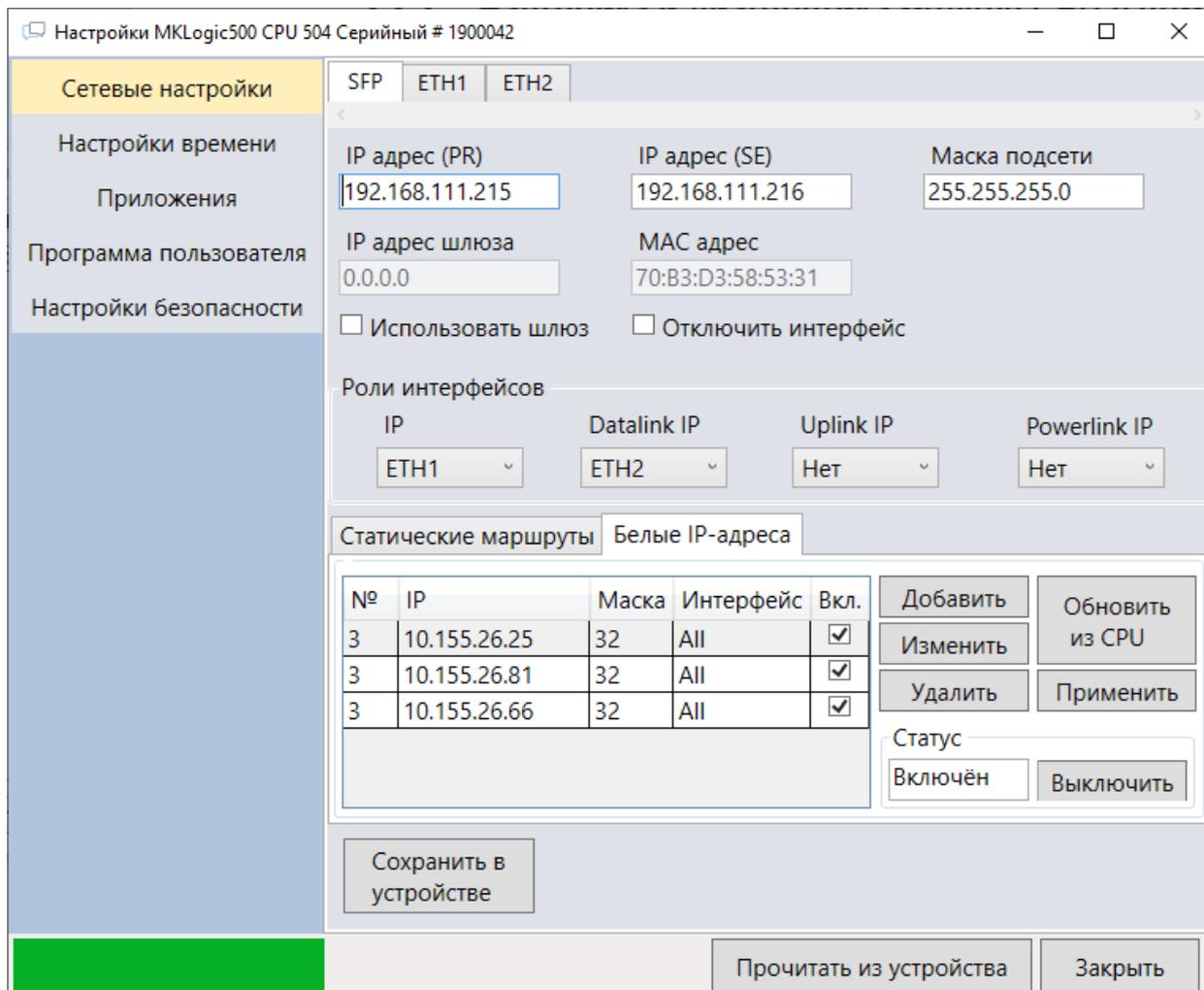


Рис. 3.7 - Окно сетевых настроек модуля CPU с белыми IP-адресами

При нажатии кнопки «Добавить» появляется окно настроек белого IP-адреса (Рис. 3.8).

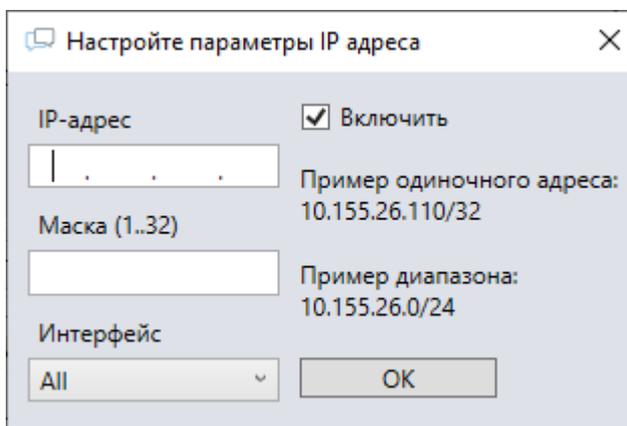


Рис. 3.8 - Окно настроек белых IP-адресов модуля CPU

Белые IP-адреса запишутся в модуль CPU при нажатии кнопки «Применить». Также имеется возможность изменить «Статус» работы «Белых IP-адресов» нажав на кнопку «Включить» или «Выключить» (в зависимости от статуса текст кнопки меняется).

Примечание В список «Белых IP-адресов» следует вносить только IP-адреса с которых непосредственно планируется подключаться к модулю CPU (не следует вносить IP-адреса сетевых интерфейсов модуля).

3.4.2. Настройки времени

В верхней части окна настроек модуля CPU в разделе «Настройки времени» (Рис. 3.9) расположены поля ввода и кнопки для настройки временных параметров модуля CPU.

Рис. 3.9 - Окно настроек времени модуля CPU

Текущие время и часовой пояс часов реального времени модуля CPU могут быть обновлены по нажатию кнопки «Время CPU». Под ними находятся поля ввода времени и часового пояса. Нажатием кнопки «Время PC» их значения можно синхронизировать со значениями ПК, на котором запущен Fallover. По нажатию кнопки «Применить к CPU» эти значения применяются к модулю CPU.

ВНИМАНИЕ При работе модулей CPU в паре обновление настроек NTP будет работать только для ведущего модуля CPU, в ведомый модуль CPU настройки будут скопированы автоматически. При попытке обновить настройки NTP ведомого модуля CPU изменения будут проигнорированы.

В нижней части окна расположены поля ввода для настройки параметров NTP-сервера модуля CPU и кнопки для управления NTP-сервером модуля CPU. Текущие значения адресов NTP-серверов и статус NTP-сервера могут быть обновлены нажатием кнопки «Текущие NTP», кнопка «Применить NTP» служит для записи значений адресов NTP-серверов в модуль CPU.

3.4.3. Приложения

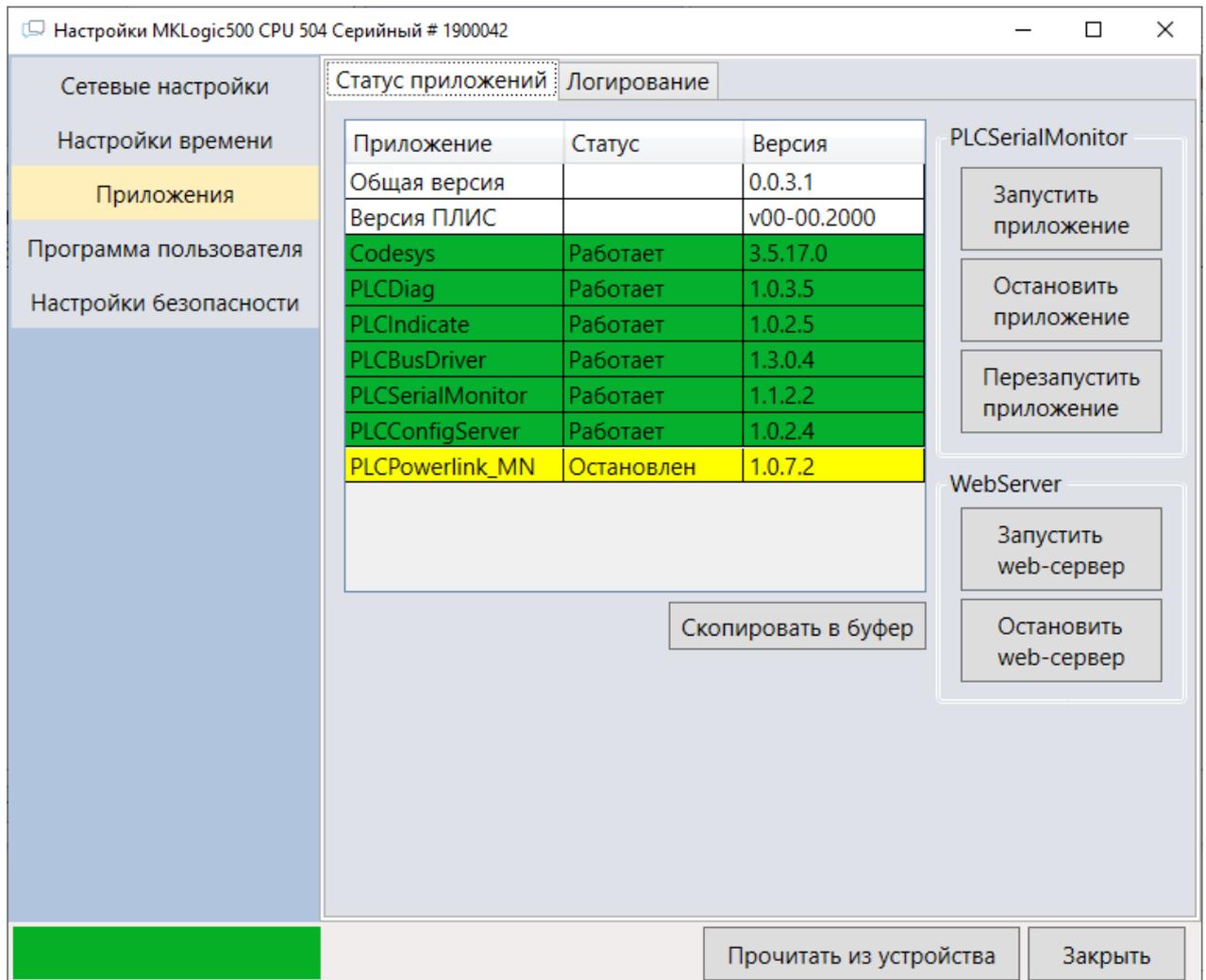


Рис. 3.10 - Окно статуса приложения модуля CPU

В верхней части окна настроек модуля CPU в разделе «Статус приложений» (Рис. 3.10) расположен список объектов системного программного обеспечения модуля CPU с указанием их имени, номера версии и текущего статуса. Также в этом списке указывается общая версия системного ПО модуля CPU. Для модулей МК-504-120 и МК-545-010 в этом списке также указываются версии прошивок микросхем ПЛИС.

Данная информация служит для диагностических целей и должна по запросу передаваться сервисным службам организации-производителя.

В правой части окна расположены кнопки управления службой PLCSerialMonitor из пакета системного программного обеспечения модуля CPU.

! **ВНИМАНИЕ** Остановка или перезапуск службы PLCSerialMonitor влечёт остановку программы пользователя.

Ручное управление работой службой PLCSerialMonitor следует использовать исключительно по согласованию с сервисными службами организации-производителя изделия.

Также в правой части окна расположены кнопки управления диагностическим web-сервером модуля CPU. При запуске модуля CPU диагностический web-сервер выключен. При его запуске (нажатием кнопки «Запустить web-сервер») к модулю CPU можно подключиться из браузера по IP-адресу сетевого интерфейса ETH1 к порту 8080 для получения диагностической информации о шинах CAN и Powerlink (для модулей CPU с поддержкой работы с Powerlink).

3.4.4. Логирование во внешнее приложение

Модулями CPU поддерживается режим перенаправления системных сообщений (логирования) во внешние приложения по протоколу syslog.

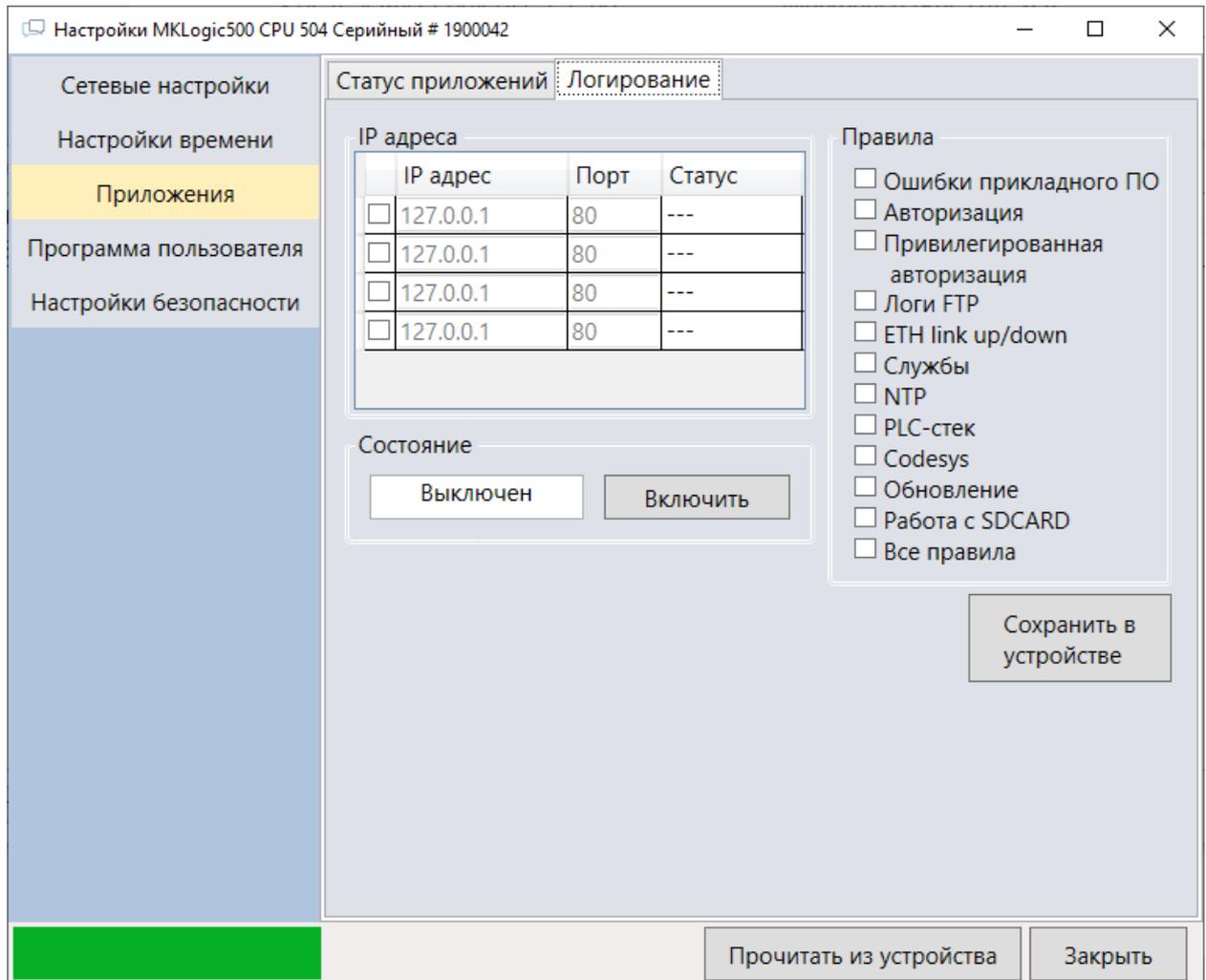


Рис. 3.11 - Окно настройки логирования системных сообщений

По умолчанию режим логирования выключен.

В верхней части вкладки «Логирование» в разделе «Приложения» (Рис. 3.11) расположен список IP-адресов и портов серверов приёма syslog-сообщений. Поддерживается отправка не более чем на 4 сервера. Там же расположен баннер статуса службы логирования и кнопка запуска/остановки службы.

В правой части вкладки расположен список правил, согласно которым отправляются syslog-сообщения. Там же расположена кнопка для сохранения правил в модуле CPU.

Для приёма и просмотра системных сообщений можно использовать любое ПО, поддерживающее получение сообщений по протоколу syslog, например, Visual Syslog Server.

3.4.5. Программа пользователя

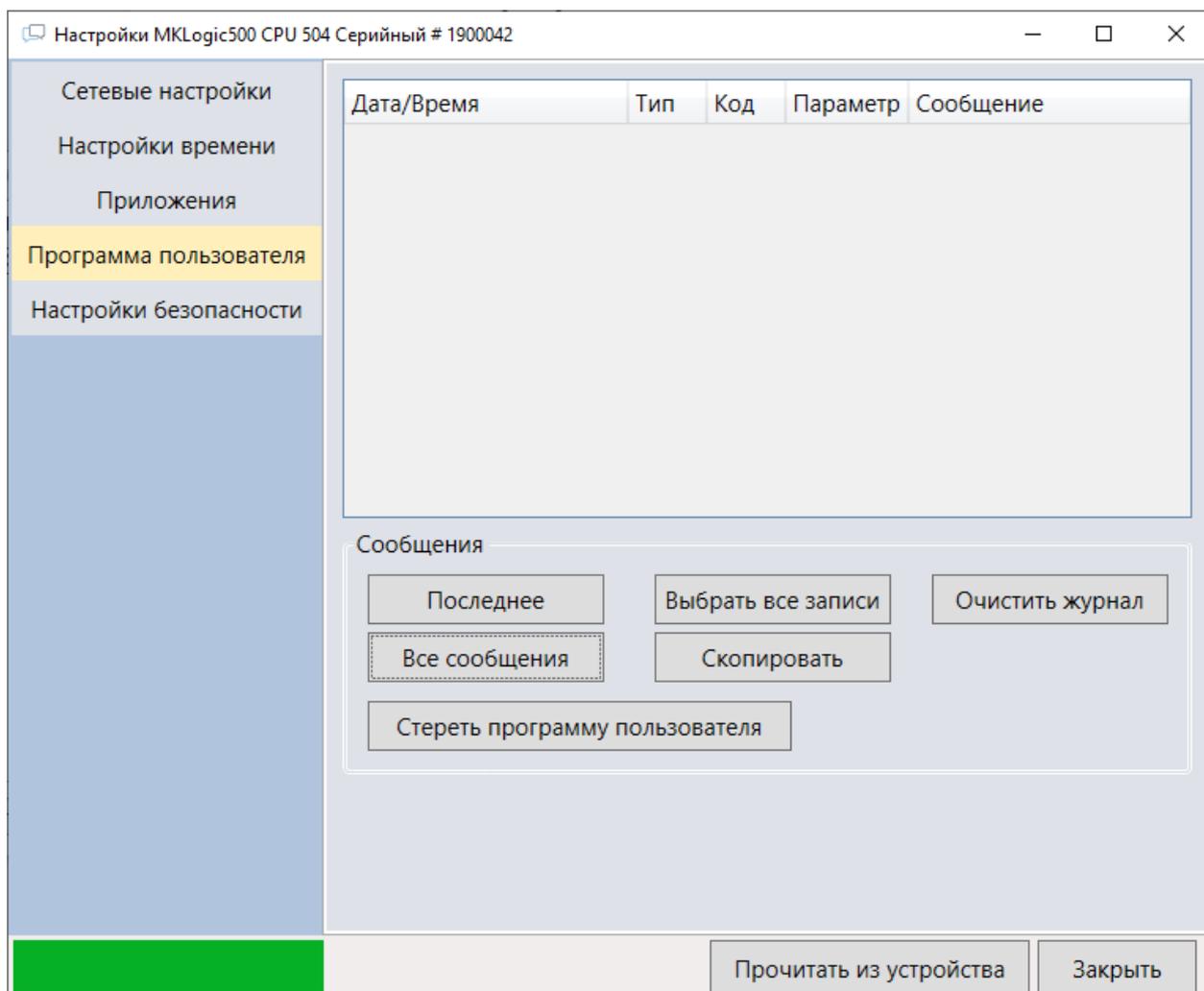


Рис. 3.12 - Окно программы пользователя модуля CPU

В верхней части окна настроек модуля CPU в разделе «Программа пользователя» (Рис. 3.12) расположен журнал сообщений программы пользователя. По умолчанию при открытии окна показывается только самое последнее сообщение журнала.

В нижней части окна расположены кнопки управления работой журнала, а также кнопка стирания программы пользователя.

Кнопка стирания программы пользователя предназначена либо для очистки модуля CPU перед его передачей третьим лицам, либо для восстановления нормальной работоспособности модуля CPU при сбое в ходе загрузки проекта в модуль CPU.

3.4.6. Настройки безопасности

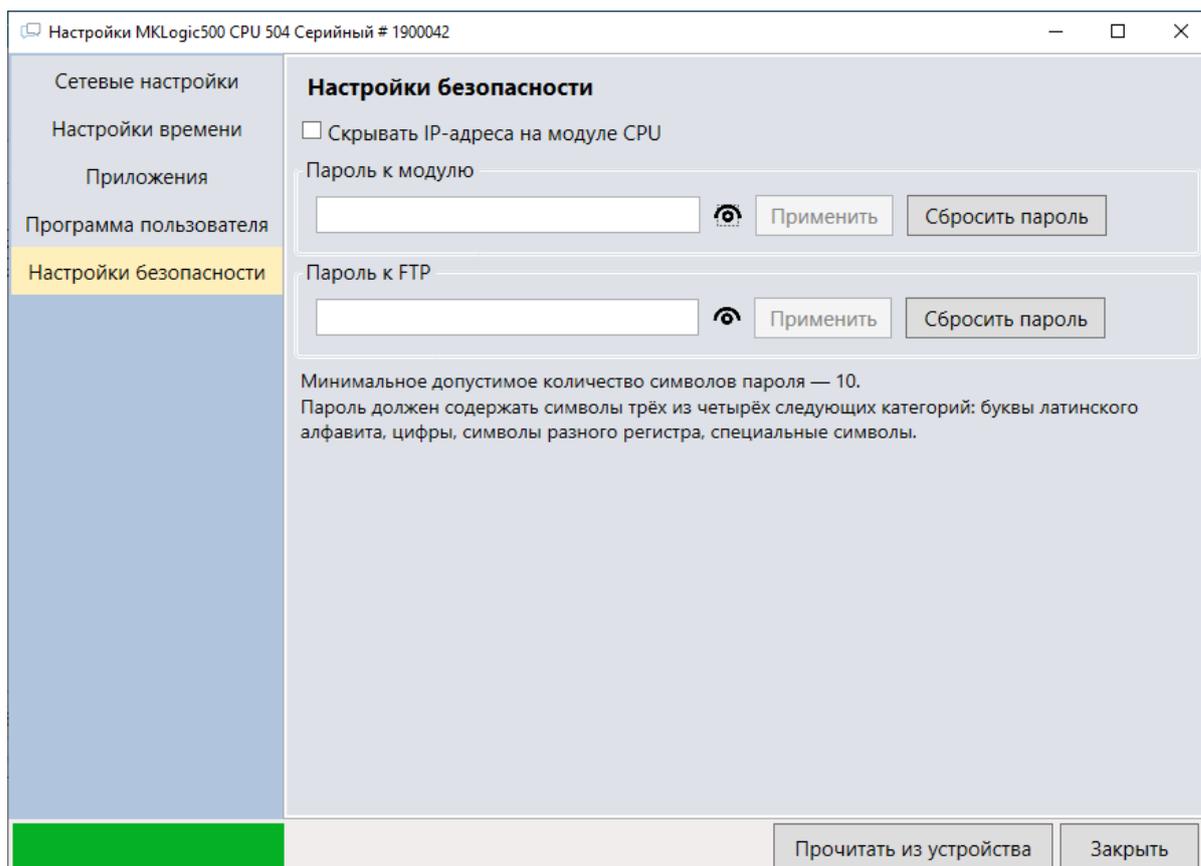


Рис. 3.13 - Окно настроек безопасности модуля CPU

Модулями CPU поддерживается парольная защита функций настройки, диагностики, а также доступа и обновления по FTP. Управление паролем выполняется в разделе «Настройки безопасности» (Рис. 3.13).

Требования к паролю описаны в центральной части рабочего окна. Кнопка «Применить к CPU» станет доступна после ввода корректного пароля. Пиктограмма в форме глаза служит для отображения вводимых символов.

Для снятия пароля с модуля CPU следует нажать кнопку «Сбросить пароль». При следующем подключении модуль будет доступен без проверки авторизации.

По нажатию на кнопку «Применить к CPU» новый пароль будет сохранён в модуле CPU. При следующем подключении к CPU при попытке доступа к защищённым функциям, потребуется пройти процедуру авторизации (Рис. 3.14). Число попыток ввода пароля не ограничено.

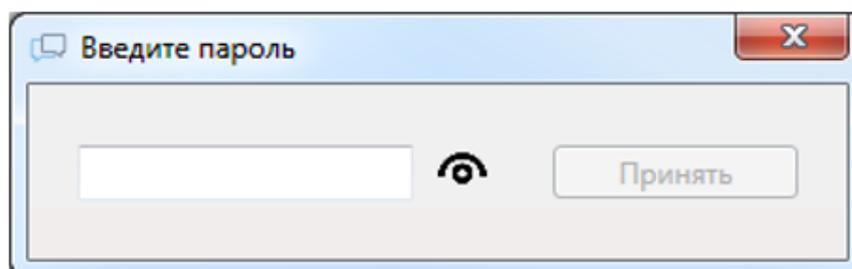


Рис. 3.14 - Окно авторизации

При необходимости можно скрыть бегущую строку с IP-адресами на панели индикации модуля CPU. Для этого нужно включить настройку «Скрывать IP-адреса на модуле CPU».

! **ВНИМАНИЕ** Скрытие IP-адреса произойдет после полного прохода бегущей строки, т.е. после нажатия настройки «Скрывать IP-адреса на модуле CPU», сначала бегущая строка отобразит все IP модуля CPU, и только при следующем проходе IP будут скрыты.

Исходно модули CPU поставляются без пароля.

Процесс сброса пароля модуля CPU с помощью Fallover

Для сброса пароля CPU потребуется с помощью Fallover:

- 1) Произвести поиск устройства CPU.
- 2) Выбрать интересующий CPU и нажать кнопку «Доступ по FTP».
- 3) Скопировать в папку файлы обновления. В папке FTP модуля CPU должны находиться следующие файлы (Рис. 3.16), скачать которые можно с официального сайта АО «Нефтеавтоматика»:

- service-504-resetPassword.enc;
- service-504-resetPassword.sign.

При наличии других файлов с расширением .enc и .sign их необходимо удалить, иначе обновление будет проигнорировано.

- 4) Выбрать в окне Fallover-а интересующий CPU и нажать кнопку «Обновить из FTP», дождаться окончания выполнения.

Далее произойдет сброса пароля (в процессе обновления стандартная индикация будет погашена). После окончания выполнения индикация вернется к стандартному виду.

Процесс сброса пароля модуля CPU с помощью SD-карты

Для сброса пароля CPU потребуется:

- 1) SD-карта с файловой системой FAT или FAT32. В корневой директории SD-карты должны находиться следующие файлы (скачать которые можно с официального сайта АО «Нефтеавтоматика»):

- service-504-resetPassword.enc;
- service-504-resetPassword.sign.

При наличии других файлов с расширением .enc и .sign их необходимо удалить, иначе обновление будет проигнорировано

- 2) Вставить подготовленную SD карту в разъем.

Далее произойдет сброс пароля (в процессе обновления стандартная индикация будет погашена). После окончания выполнения индикация вернется к стандартному виду.

При выполнении сброса пароля посредством обновления произойдет следующее:

- сбросится пароль доступа к модулю CPU;
- сотрется программа пользователя;
- сбросятся статические маршруты;
- отключенные интерфейсы перейдут в состояние «Включен»;
- сбросится белый список.

3.4.7. Настройки шифрования

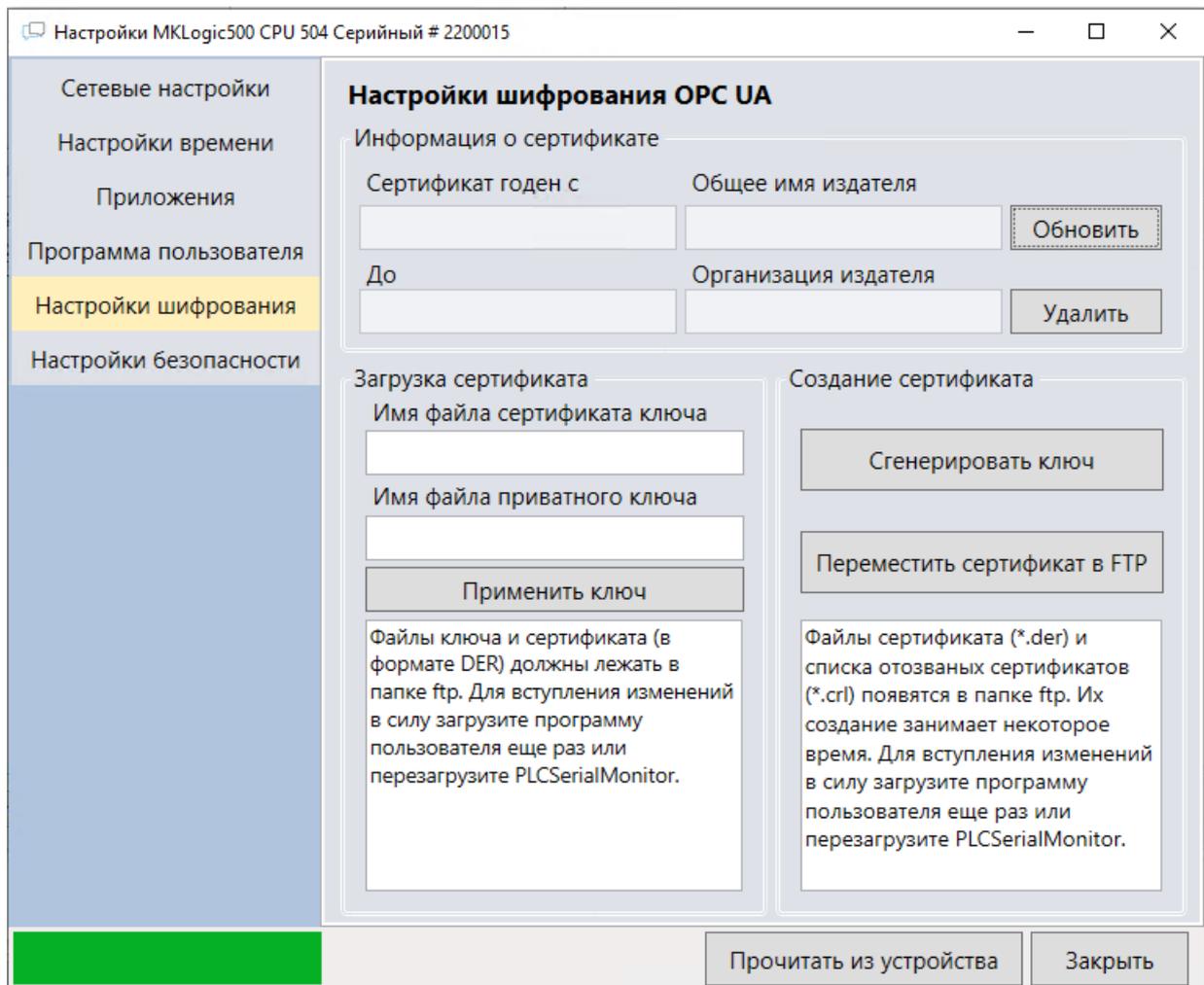


Рис. 3.15 - Раздел «Настройки шифрования»

Модули CPU 504 (МК-504-120) поддерживают шифрование обмена по протоколу OPC UA по стандарту X.509. Настроить параметры шифрования OPC UA можно в разделе «Настройки шифрования» (Рис. 3.15).

В верхней части окна настроек модуля CPU в разделе «Настройки шифрования» выводится информация о текущих параметрах сертификата шифрования X.509. Текущий сертификат можно удалить из модуля CPU нажатием кнопки «Удалить» (удаление не затрагивает папку FTP). Кнопка «Обновить» служит для вывода актуальной информации о параметрах сертификата после его создания либо загрузки.

! **ВНИМАНИЕ** Загрузка пользовательского сертификата и ключа должна производиться в активный модуль CPU.

В левой части окна расположены органы управления для загрузки пользовательского сертификата ключа. Для загрузки пользовательского сертификата в модуль CPU файлы сертификата ключа и приватного ключа (оба в формате DER) должны быть скопированы по FTP в модуль CPU, их имена следует ввести в соответствующие поля ввода, после чего следует нажать кнопку «Применить ключ».

Если загрузка сертификата произошла успешно, то файлы сертификата и ключа будут удалены из папки FTP и перемещены в модуль CPU. Также сертификат будет автоматически загружен в резервный модуль CPU. Для применения загруженного сертификата следует перезапустить программу пользователя с запущенным устройством `opcua_server`.

В правой части окна расположены кнопки для генерации самоподписанного сертификата в модуле CPU. Для генерации ключа следует нажать кнопку «Сгенерировать ключ». В корне FTP модуля CPU должны появиться файл сертификата с именем «`nft.crt.der`» и файл со списком

отозванных сертификатов «nft.der.crl», которые нужны для подключения к OPC UA клиента к серверу. При нажатии кнопки «Переместить сертификат в ftp» эти файлы снова будут скопированы в корень FTP.

Сгенерированный сертификат будет автоматически загружен в резервный модуль CPU. Для применения сгенерированного сертификата следует перезапустить программу пользователя с запущенным устройством orcsua_server.

! **ВНИМАНИЕ** Создание пользовательского сертификата должно производиться на активном модуле CPU*.

Пример сценария bash для создания файлов самоподписанного сертификата и приватного ключа X.509 в Linux, с помощью библиотеки OpenSSL (см.Листинг 3.1).

```
# Генерируем самоподписанный сертификат X.509 в папке ca
# Create directory to store CA's files
mkdir ca
# Create CA key
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048

-out ca/example.key
# Create self-signed CA cert
openssl req -new -x509 -days 3600 -key ca/example.key

-subj "/CN=EXAMPLE CA/O=EXAMPLE Organization" -out ca/example.crt
# Convert cert to der format
openssl x509 -in ca/example.crt -inform pem -out ca/example.crt.der

-outform der
# Create cert revocation list CRL file
# NOTE : might need to create in relative path
# - File './demoCA/index.txt' (Empty)
# - File './demoCA/crlnumber' with contents '1000'
openssl ca -crldays 3600 -keyfile ca/example.key -cert ca/example.crt

-gencrl -out ca/example.crl
# Convert CRL to der format
openssl crl -in ca/example.crl -inform pem -out ca/example.der.crl

-outform der
# Генерируем файлы ключа и сертификата в папке OPC_UA
# Create directory to store server's files
mkdir OPC_UA
# Create server key
openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048

-out OPC_UA/server.key
# Convert server key to der format
openssl rsa -in OPC_UA/server.key -inform pem -out OPC_UA/server.key.der

-outform der
# Create server cert sign request
openssl req -new -sha256 -key OPC_UA/server.key
-subj "/C=ES/ST=MAD/O=MyServer/CN=localhost" -out OPC_UA/server.csr
# Sign cert sign request (NOTE: must provide exts.txt)
openssl x509 -days 3600 -req -in OPC_UA/server.csr -extensions v3_ca

-extfile OPC_UA/exts.txt -Ccreateserial -CA ca/example.crt

-Cakey ca/example.key -out OPC_UA/server.crt
# Convert cert to der format
openssl x509 -in OPC_UA/server.crt -inform pem -out OPC_UA/server.crt.der

-outform der
```

Листинг 3.1– Сценарий создания файлов сертификата и ключа X.509

После выполнения этого сценария в папке OPC-UA будут созданы пара файлов server.crt.der и server.key.der.

3.4.8. Подключение и обновление модуля CN (МК-545-010) с помощью Fallover

Для начала настройки модуля CN следует выполнить следующие операции:

- 1) Разорвать кольцо/линию POWERLINK.
- 2) Перевести переключатели адреса модуля CN в запрещенное положение (Табл. 3.1).

Таблица 3.1 – Разрешённые положения переключателей адреса

Положение	Address High	Address Low
1	+	+
2	+	+
3	+	+
4	+	+
5	запрещено	+
6	запрещено	+
7	запрещено	+
8	запрещено	+
9	запрещено	+
A	запрещено	+
B	запрещено	+
C	запрещено	+
D	запрещено	+
E	запрещено	+
F	запрещено	+

3) С помощью сетевого кабеля подключить порт «PORT1.1» настраиваемого модуля CPU непосредственно к порту рабочего ПК, либо к общему с рабочим ПК сетевому коммутатору.

4) В окне приложения Fallover выбрать IP-адрес сетевого интерфейса, к которому подключён настраиваемый модуль CN; затем нажать кнопку «Поиск устройства»/«Поиск по IP». По окончании поиска таблица будет заполнена перечнем модулей CPU/CN, доступных для настройки и диагностики.

5) Выбрать строку с интересующим модулем и нажать кнопку «Доступ по FTP» (для доступа может потребоваться настроить подсеть рабочего ПК в соответствии с IP-адресом на панели индикации) и скопировать в папку файлы обновления. В папке FTP модуля CN должны находиться следующие файлы (скачать которые можно с официального сайта АО «Нефтеавтоматика»):

- patch-545-####-##-##_####.enc;
- patch-545-####-##-##_####.sign.

6) Выбрать в окне Fallover-а интересующий CN и нажать кнопку «Обновить из FTP», дождаться окончания выполнения (в процессе обновления стандартная индикация будет погашена и будут отображаться стадии обновления от 1 до 7). После окончания выполнения обновления индикация вернется к стандартному виду.

! **ВНИМАНИЕ** После завершения обновления рекомендуется удалить файлы с расширением .enc и .sign из модуля CN во избежание случайного повторного обновления.

После окончания выполнения индикация вернется к стандартному виду, а так же в модуле CN будет создан файл журнала вида logpatch-XXX-YYYY-mm-dd-HHMMSS.txt (где XXX – тип модуля).

! **ВНИМАНИЕ** Во избежание некорректной работы модуля не рекомендуется на модуль со «свежим» обновлением устанавливать обновления предшествующие ему.

3.4.9. Обновление модуля CPU с помощью Fallover

Для обновления модуля CPU с помощью Fallover потребуется:

- 1) Произвести поиск устройства CPU.потребуется

2) Выбрать интересующий CPU и нажать кнопку «Доступ по FTP».

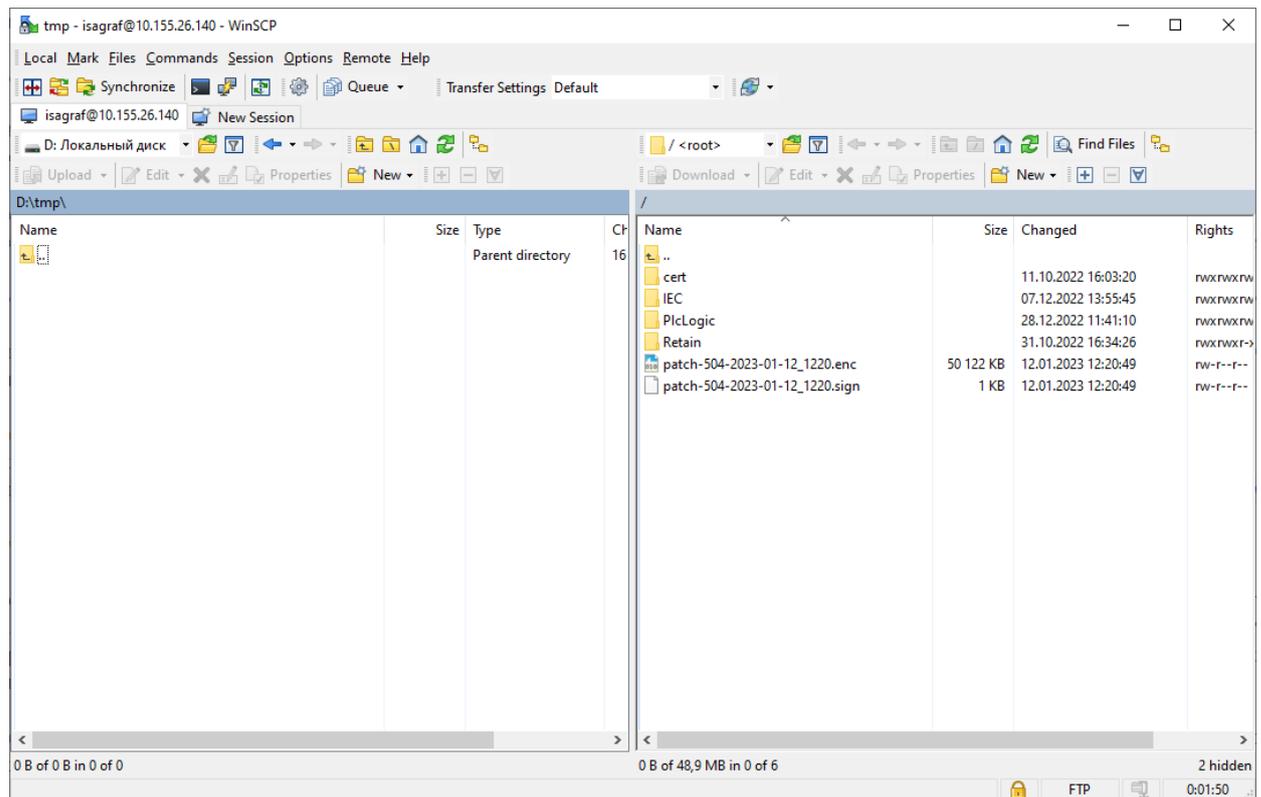


Рис. 3.16 - Окно проводника в режиме доступа по FTP к модулю CPU с файлами обновления

3) Скопировать в папку файлы обновления. В папке FTP модуля CPU должны находиться следующие файлы (Рис. 3.16), скачать которые можно с официального сайта АО «Нефтеавтоматика»:

- patch-504-####-##-##_####.enc;
- patch-504-####-##-##_####.sign.

При наличии других файлов с расширением .enc и .sign их необходимо удалить, иначе обновление будет проигнорировано.

4) Выбрать в окне Fallover интересующий CPU и нажать кнопку «Обновить из FTP», дождаться окончания выполнения.

Далее произойдет обновление (в процессе обновления стандартная индикация будет погашена). После окончания выполнения индикация вернется к стандартному виду.

3.4.10. Обновление модуля CPU с SD-карты

Для обновления модуля CPU потребуется с помощью SD-карты:

1) SD-карта с файловой системой FAT или FAT32. В корневой директории SD-карты должны находиться следующие файлы, скачать которые можно с официального сайта АО «Нефтеавтоматика»:

- patch-504-####-##-##_####.enc;
- patch-504-####-##-##_####.sign.

При наличии других файлов с расширением .enc и .sign их необходимо удалить, иначе обновление будет проигнорировано.

2) Вставить подготовленную SD карту в разъём.

Далее произойдет обновление (в процессе обновления стандартная индикация будет погашена). После окончания выполнения индикация вернется к стандартному виду.

4. Работа в среде разработки Procyon IDE

4.1. Системные требования

Требования к техническим средствам (компонентам), необходимые для установки и исполнения среды Procyon IDE, приведены в Табл. 3.1.

Табл. 3.1 – Технические требования для работы среды Procyon IDE

Характеристики технических средств	Требования
Тактовая частота процессора ПК	1000 МГц (32 бита/ 64 бит)
Оперативная память ПК	1 Гб
Операционная система	Astra Linux 1.7.5+
Монитор	не менее 17
Минимальное разрешение экрана	1600x900
Свободное место на жестком диске	1 Гб

4.2. Установка и настройка среды разработки Procyon IDE

4.1.1. Установка среды разработки Procyon IDE

Перед установкой скачайте дистрибутив Procyon IDE с официального сайта [АО «Нефтеавтоматика»](#). Необходимо выбрать архитектуру вашей системы (32 или 64 бит). Для установки среды разработки Procyon IDE следует запустить скачанный файл и следовать указаниям мастера установки. При появлении предупреждающего окна, разрешите внесение изменений на данном компьютере, нажав кнопку «Да».

4.1.2. Установка пакета поддержки контроллера MKLogic-500

Скачайте с сайта [АО «Нефтеавтоматика»](#) пакет поддержки контроллера MKLogic-500 последней версии. В среде Procyon IDE выберите меню «Инструменты»→«Менеджер пакетов». В открывшемся окне нажмите кнопку «Установить...» и укажите файл скачанного пакета.

В появившемся окне «Проверить подписи пакетов» выберите «MKLogic-500 Procyon IDE Package», установите галочку «Не подписанные и самоподписанные пакеты» и нажмите «ОК». Выберите типовую установку и нажмите «Далее >>»(Рис. 4.1).

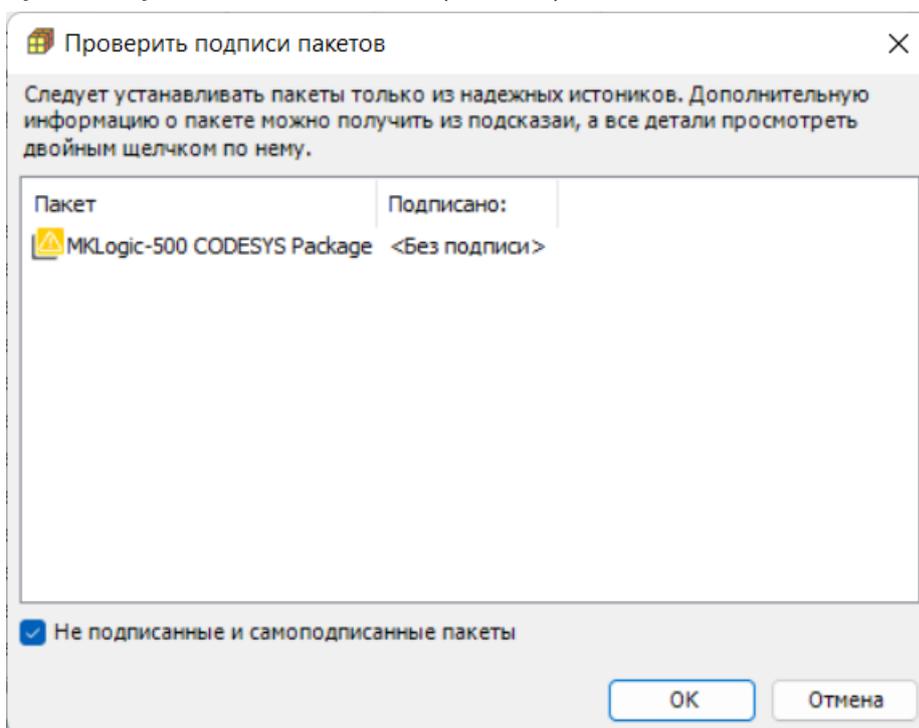


Рис. 4.1 - Установка пакета поддержки MKLogic-500

При появлении предупреждающего окна, разрешите внесение изменений на данном компьютере, нажав кнопку «Да». Завершите установку пакета нажатием кнопки «Завершить».

В данном руководстве не рассматривается подробно процесс программирования в среде Procyon IDE. Более подробную информацию можно получить на сайте [справочной системы Procyon IDE](#).

4.2. Создание проекта MKLogic-500 в Procyon IDE

Создание проекта происходит так же как описано в [справочной системы Procyon IDE](#) в разделе «Создание и конфигурация проектов». Рекомендуется создавать проект с помощью диалога Стандартный проект.

Нажмите «Файл»→«Новый проект» и затем шаблон «Проекты»→«Стандартный проект». Введите имя (например: myProject) и расположение в файловой системе.

В результате откроется диалог Стандартный проект.

Далее выберите устройство Nefteavtomatika MKLogic-500(504) из списка «Устройство» и Структурированный текст (ST) из списка PLC_PRG. Затем нажмите ОК.

В результате будет создан проект со стандартной структурой. Имя проекта myProject показано на панели заголовка главного окна системы разработки. В дереве устройств (окно Устройства) будет создана следующая структура (Рис. 4.2).

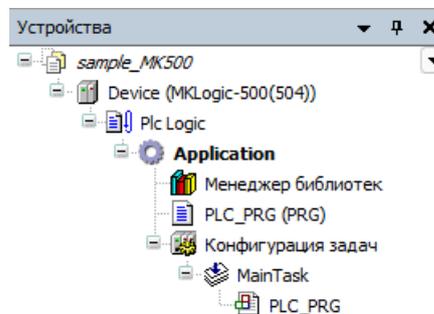


Рис. 4.2 - Стандартный вид проекта

! **ВНИМАНИЕ** Для полноценной поддержки резервирования в контроллерах MKLogic-500 необходима генерация загрузочных приложений при загрузке и при онлайн-замене. Для включения этих опций необходимо нажать правую кнопку на приложении и выбрать пункт меню «Свойства...»→«Установки загрузочного приложения...» и поставить соответствующие галочки (Рис. 4.3 и Рис. 4.4)

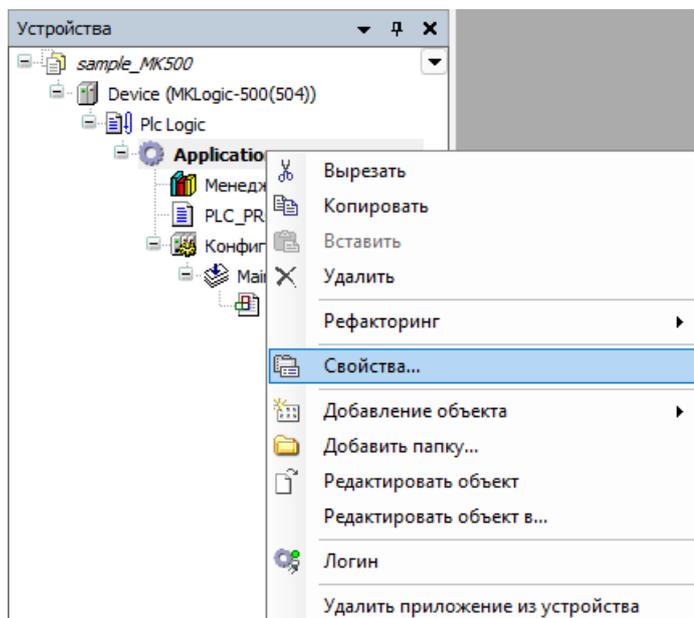


Рис. 4.3 - Свойства приложения

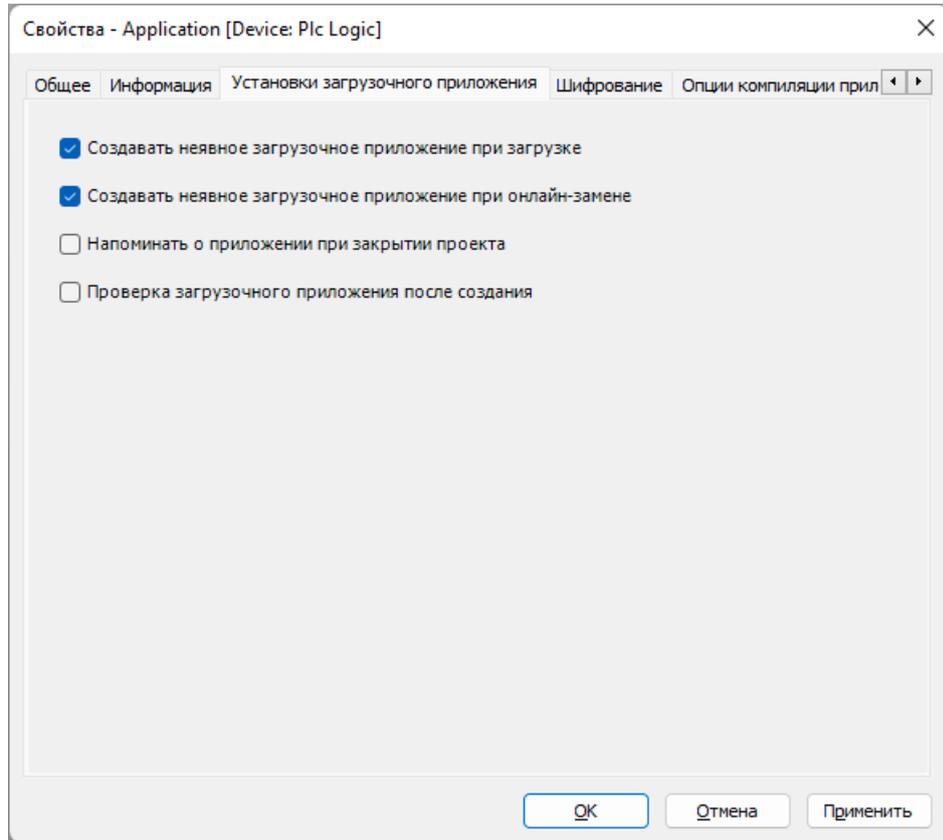


Рис. 4.4 - Свойства приложения

4.3. Структура проекта (Дерево устройств)

4.3.1. Дерево устройств

В окне «Устройства» (или дереве устройств) в соответствии с целевым устройством располагаются ваши объекты дерева устройств.

Корневым узлом дерева устройств является символьный элемент: «».

Под этим узлом вставляются объекты, соответствующие ПЛК, которые также называются целевыми системами и являющиеся программируемыми устройствами. Для ПЛК МКLogic-500 таким объектом является устройство «MKLogic- 500 (504)». Программируемые устройства автоматически снабжаются дополнительным узлом «PLC logic» под объектом устройства исключительно для организации. Под этим узлом вы можете вставлять объекты «Приложение» (значок ).

В программируемое устройство в качестве дочернего устройства могут быть вставлены параметризируемые устройства, такие как специфические аппаратные средства, fieldbus-системы, и др.

При добавлении объектов вам автоматически предлагаются все возможные устройства из локального репозитория устройств. Каждое устройство определяется описанием устройства и должно быть установлено в локальной системе для того, чтобы его можно было вставить в дерево устройств. Файлы описания всех доступных устройств ПЛК МКLogic-500 поставляются в пакете поддержки.

 **Примечание** Окно POU содержит объекты, которые используются во всем проекте. Объекты программирования, предназначенные для определенного приложения, следует вставлять под объектом этого приложения в окне Устройства (дерево устройств).

По умолчанию в системе разработки предоставляется возможность запуска активного приложения на эмулируемом устройстве. В данный момент опция эмуляции доступна для целевой системы Procyon IDE Control Win V3. В режиме эмуляции вы также можете протестировать онлайн-функциональность приложения без использования аппаратного обеспечения. Режим эмуляции включается командой меню «Онлайн»→«Эмуляция».

Обратите внимание на возможность подключения к устройству с помощью команды Режим онлайн-конфигурации для приложения без предварительной загрузки. Это может быть полезно при первичном вводе в эксплуатацию I/O- системы, поскольку вы можете протестировать входы/выходы в конфигурации ПЛК еще до программирования и загрузки самого приложения.

Пример дерева устройств (Рис. 4.5):

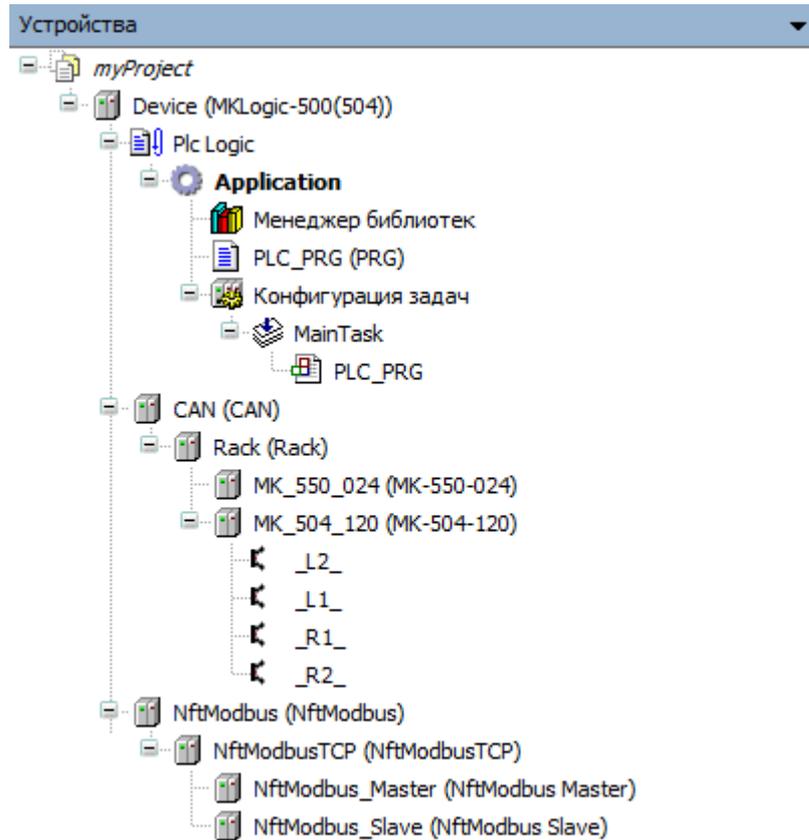


Рис. 4.5 - Пример дерева устройств

Элемент устройства в дереве устройств включает в себя значок, символьное имя устройства, которое можно редактировать в дереве, и тип устройства (имя устройства, заданное в описании устройства).

Установки связи устройства, параметры и соотношения входов/выходов задаются в диалогах редактора устройств. Он открывается двойным щелчком мыши по объекту устройства.

4.3.2. Правила и механизмы расположения и конфигурации объектов в дереве устройств

— Вставка объектов: Чтобы вставить объект устройства, нажмите «Добавить устройство» или «Вставить устройство» в контекстном меню дерева устройств (см. Рис. 4.6 и Рис. 4.7). Для других объектов используется команда «Добавить объект» (см. Рис. 4.8). Просуоп IDE всегда предлагает только те объекты, которые могут быть вставлены на выбранной позиции. Пример: Модули ввода-вывода МКLogic-500 могут быть вставлены только под объектом Rack, а приложения вставляются только под программируемыми устройствами. Доступность объектов устройств также зависит от того, какие устройства установлены в репозитории устройств.

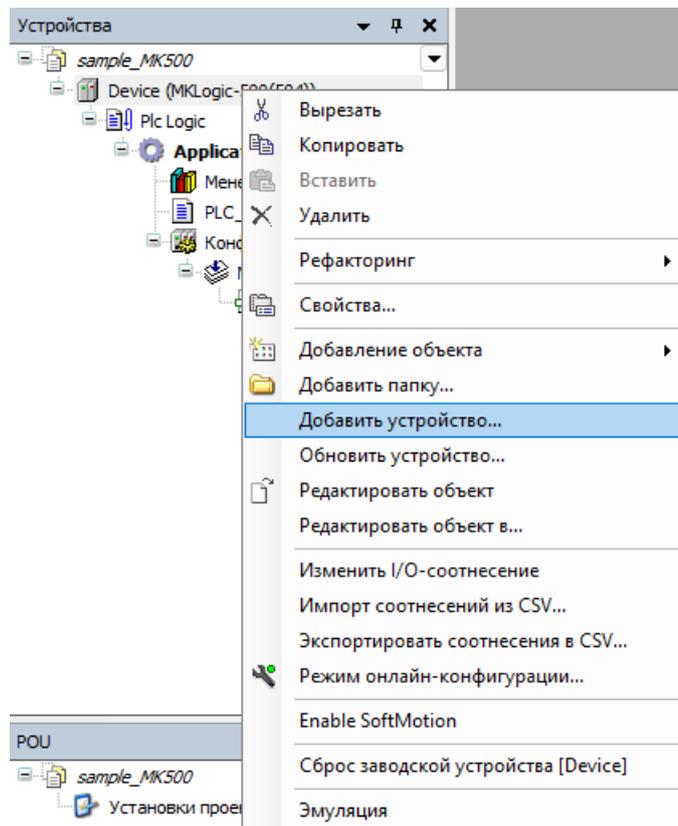


Рис. 4.6 - Добавление устройства в проект

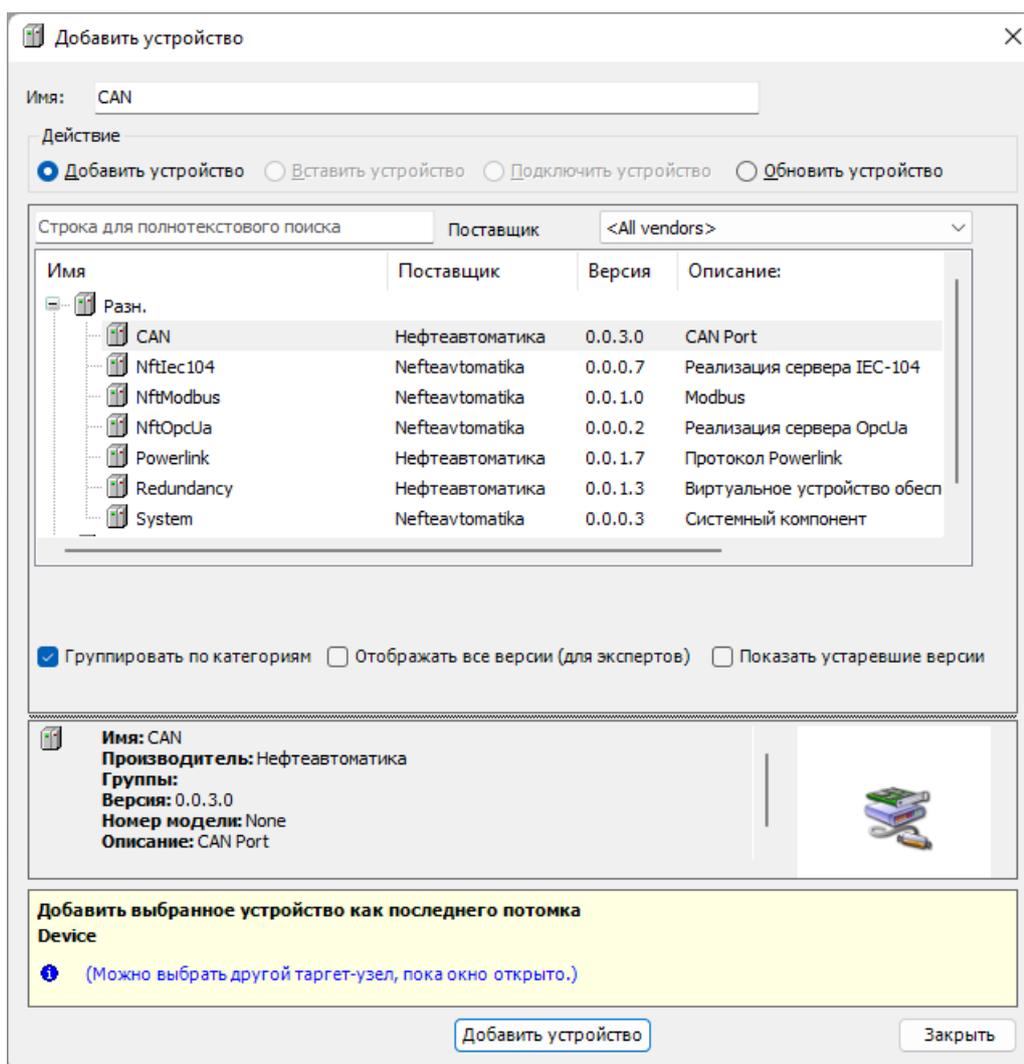


Рис. 4.7 - Выбор устройства для добавления в проект

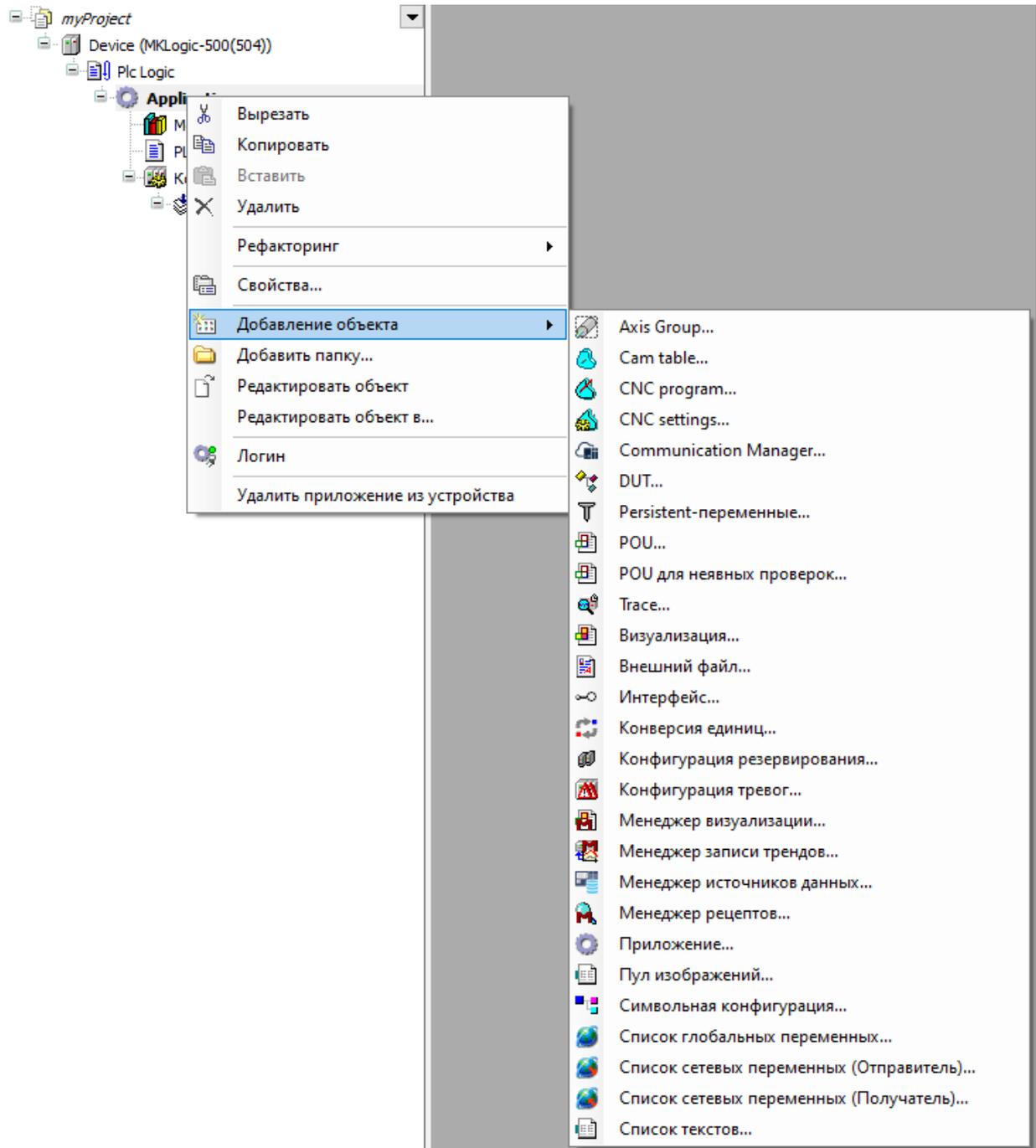


Рис. 4.8 - Добавление объекта в проект

— **«На уровне сразу под корневым узлом»:** Можно вставлять только объекты программируемых устройств. Если вы выберете объект другого типа, например, список текстов, то Prosupn IDE автоматически вставит его в окно «POU» (глобальный пул).

— **«Вставка приложений»:**

— объект Приложение может быть вставлен только под узлом «PLC logic» (программируемое устройство). Все приложения должны иметь уникальные имена для каждого устройства. Под каждым приложением вы можете вставлять дополнительные объекты, необходимые для программирования, такие как POU, DUT, GVL и визуализации (см. раздел «Дочерние объекты «Приложения»»);

— под каждым приложением вы должны вставить конфигурацию задач и определить соответствующие вызовы программы (POU для конкретных приложений или экземпляры POU из окна POU);

— ели непосредственно под устройством расположено несколько приложений, необходимо задать параметры обработки входов/выходов устройства. Это относится к переменным приложения, которые Procyon IDE должен использовать для связи с целевой системой. Эти параметры задаются на вкладке «Установки ПЛК» редактора устройств;

— иерархическое расположение приложений, области: Приложение может быть также вставлено под другим приложением. Такая структура имеет вид «Родительское приложение» - «Дочернее приложение». В этом случае действуют следующие правила: Дочернее приложение может обращаться к объектам родительского приложения, но не наоборот. Это сделано для того, чтобы дочернее приложение можно было удалить или изменить, не затрагивая родительское приложение.

! **ВНИМАНИЕ** После того, как родительское приложение было изменено, ПЛК удаляет дочернее приложение при выполнении онлайн-изменения.

— **«Вставка устройств»:** Procyon IDE вставляет объект устройства в дерево в виде узла. Если узлы заданы в описании устройства, они вставляются автоматически. Программируемое устройство может быть также представлено под узлом. Порядок объектов устройств в дереве (сверху вниз): На каждом уровне сначала представлено программируемое устройство (PLC Logic), после которого в алфавитном порядке перечислены остальные типы.

— **«Обновление устройств»:** Устройство, добавленное в дерево устройств, может быть заменено на другую версию того же устройства или на устройство другого типа («Обновить устройство»). При наличии под устройством доступно дерево конфигурации.

— **«Перемещение и удаление объектов»:** Вы можете использовать стандартные команды над объектами, такие как «Вырезать», «Копировать», «Вставить» и «Удалить» или просто перетащить объект на другое место. Когда вы копируете объект, новый объект получает то же имя, к которому добавляется порядковый номер.

Использование скрипта MK500Options.py для автоматизации обновления устройств

Для больших проектов обновление всех устройств дерева может занимать большое количество времени. Для упрощения этой процедуры предусмотрен скрипт MK500Options.py, написанный на языке Python (интерпретатор Python версии 2.7 встроен в среду Procyon IDE). Среда Procyon IDE поддерживает выполнение скриптов на языке Python (Рис. 4.9).

Скрипт MK500Options.py (входит в стандартный пакет поддержки МК-500) проходит по всему дереву устройств и обновляет их. После установки пакета поддержки МК-500 скрипт MK500Options.py располагается по стандартному пути "C:\LibrariesNefteavtomatika".

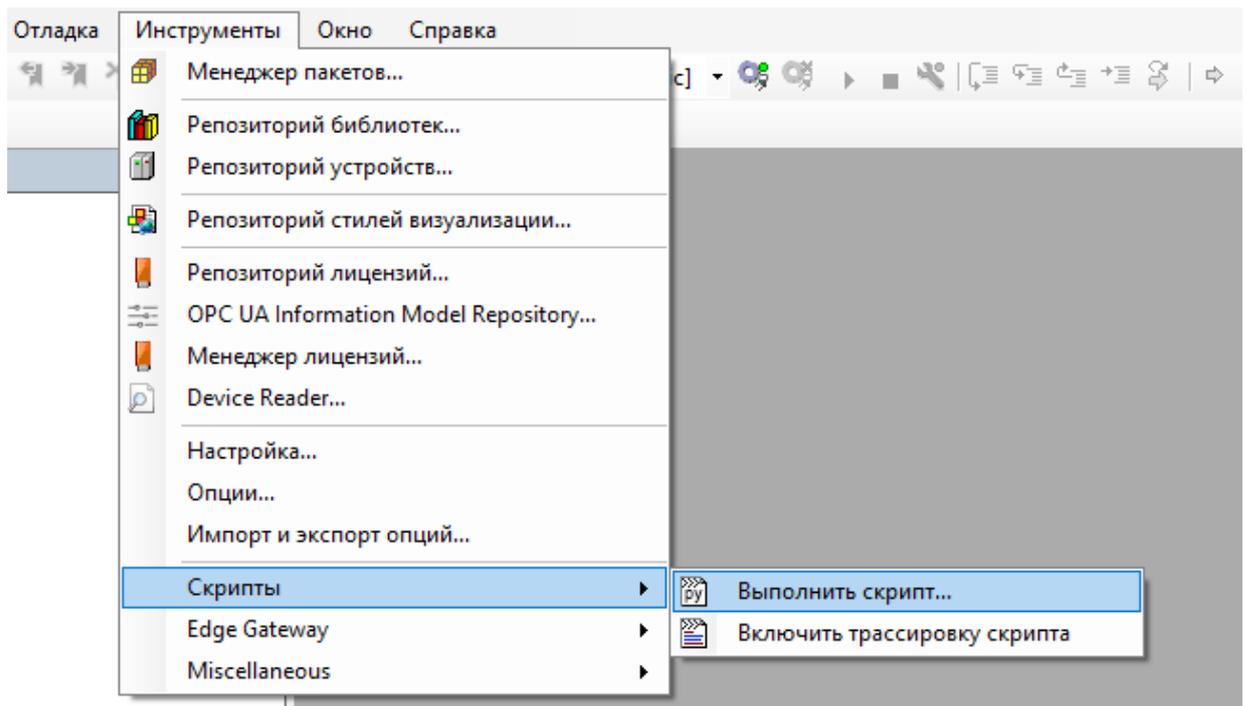


Рис. 4.9 - Запуск скрипта MK500Options.py

4.3.3. Дочерние объекты «Приложения»

Добавление дочерних объектов в объект «Приложение» производится путем вызова контекстного меню, нажатием правой кнопки мыши на узле объекта «Приложение» и выбором меню «Добавление объекта».

В объект «Приложение» можно добавить следующие объекты:

— объект «DUT»  (Data Unit Type) определяет пользовательский тип данных. Выбрать можно из следующих типов данных: Структура, Перечисление, Псевдоним, Объединение;

— объект «Внешний файл» - это любой файл который вы хотите добавить в проект. Файл добавленный в глобальную область видимости (Окно «POU») не загружается в контроллер. Файл добавленный в приложение (окно «Устройства») всегда загружается в контроллер при загрузке или онлайн замене;

— объект «GVL»  (Global Variable List) - список глобальных переменных, служит для объявления, редактирования и отображения глобальных переменных. Если объект «GVL» вставлен как дочерний объект объекта «Приложение», то его переменные доступны в рамках приложения, если же «GVL» добавлен в окно «POU», переменные доступны в рамках всего проекта;

— объект Persistent-переменные  - служит для объявления Persistent-переменных, эти переменные хранятся в специальной энергонезависимой памяти;

— объект «Менеджер библиотек»  содержит список всех библиотек которые были интегрированы в проект для создания приложения;

— объект «POU»  (Program Organization Unit) - объект для создания исходного кода программы. Существует три типа объектов «POU»: «Программа», «Функциональный блок», «Функция»;

— объект «POU для неявных проверок»  - специальные функция которые используются для неявного мониторинга;

— объект «Конфигурация задач»  используется для определения и отображения основных установок для конфигурации задач;

— объект «Символьная конфигурация» - используется для создания символьного описания переменных проекта.

4.3.4. Дерево устройств в режиме онлайн

В режиме онлайн значок рядом с объектом устройства указывает на его статус:

 : ПЛК подключен, приложение запущено, устройство работает, обмен данными выполняется. Опция «Обновлять IO при остановке» на вкладке «Установки ПЛК» может быть включена или отключена;

 : ПЛК подключен и находится в состоянии «СТОП»; опция «Обновлять IO при остановке» на вкладке «Установки ПЛК» отключена;

 : ПЛК подключен, приложение запущено. Доступна диагностическая информация;

 : Устройство находится в предрабочем режиме и пока не запущено. Доступна диагностическая информация;

 : Обмен данными с устройством не выполняется; ошибка шины, отсутствует конфигурация или режим эмуляции;

 : Устройство работает в демо-режиме 30 минут. По истечении этого времени демо-режим будет завершен, и обмен данными прекратится;

 : Устройство настроено, но не в полностью рабочем состоянии. Обмен данными не выполняется. Например: CANopen-устройства при запуске в предрабочем режиме;

 : Включен режим резервирования. Fieldbus-мастер не отправляет никаких данных, поскольку активен другой мастер;

 : Описание устройства не найдено в репозитории устройств.

Имена всех подключенных устройств и приложений подсвечиваются зеленым (Рис. 4.10).

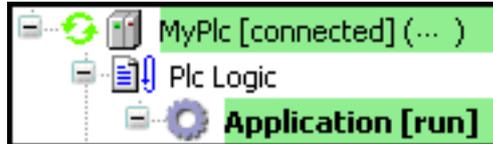


Рис. 4.10 - Отображение подключенных устройств

Имена устройств, работающих в режиме эмуляции, выделяются курсивом (Рис. 4.11):

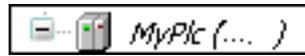


Рис. 4.11 - Отображение устройств в режиме эмуляции

Дополнительная диагностическая информация доступна на вкладке Состояние соответствующего редактора устройств.

 **Примечание** Если вы попытаетесь выполнить логин в то время, как описание устройства на целевом устройстве будет новее, чем в проекте, появится предупреждение, с помощью которого вы сможете отменить действие.

4.4. Генерация прикладного кода

Прикладной код — это машинный код, который выполняется контроллером при запуске приложения.

Procyon IDE автоматически выполнит генерацию прикладного кода из исходного кода, введенного в системе разработки. Это происходит автоматически перед загрузкой приложения в ПЛК. Перед генерацией прикладного кода проверяются все присваивания, типы данных и доступность библиотек. Кроме этого во время генерации кода происходит распределение адресов памяти.

Эту команду можно выполнить явно, нажав «Компиляция»→«Генерировать код». Это может быть полезно для выявления ошибок в исходном коде еще до подключения ПЛК. Ошибки выводятся в окне сообщений (категория «Компиляция»).

4.4.1. Установка среды разработки Procyon IDE

Перед установкой скачайте дистрибутив Procyon IDE с официального сайта [АО «Нефтеавтоматика»](#). Необходимо выбрать архитектуру вашей системы (32 или 64 бит). Для установки среды разработки Procyon IDE следует запустить скачанный файл и следовать указаниям мастера установки. При появлении предупреждающего окна, разрешите внесение изменений на данном компьютере, нажав кнопку «Да».

4.4.2. Сообщения при генерации прикладного кода

Когда вы генерируете прикладной код, Procyon IDE выводит информацию о распределении памяти в окне сообщений. В памяти возникают промежутки, потому что перераспределение подразумевается только для новых и измененных блоков и переменных из-за инкрементной компиляции памяти. Онлайн-замена имеет тот же эффект. Подобная фрагментация уменьшает количество доступной памяти. Однако с помощью команды «Компиляция»→«Очистить» можно выполнить полное перераспределение памяти и, таким образом, увеличить количество свободной памяти.

Дополнительная информация о сообщениях при генерации кода: Ошибки синтаксиса и ошибки, которые Procyon IDE выявляет во время генерации кода и распределения памяти, выводятся в окне сообщений в категории «Компиляция»(Рис. 4.12). При каждой генерации кода вы получаете дополнительную информацию о размере кода и данных (в байтах), содержании распределенных областей памяти и наивысшем используемом адресе (байт). От ПЛК зависит, какие данные и код в каких областях памяти хранятся. В МКLogic-500(504) код и данные хранятся в одной области. Под адреса %I, %M и %Q всегда зарезервирована память, даже если им не присвоены переменные. После очистки приложения память полностью перераспределяется. В этом случае могут появиться небольшие пробелы в памяти в связи с заданным выравниваем (как правило, 8 байт). Более крупные промежутки могут возникать в результате изменения даты без очистки, например, с увеличением диапазона массива. В этом случае перекомпилируются только затронутые ROU.



Рис. 4.12 - Окно с результатами компиляции

Онлайн- замена имеет тот же эффект. Полученные промежутки по возможности используются для других изменений. Поскольку малые промежутки больше не могут использоваться в некоторых случаях, размер “наибольшего согласованного промежутка памяти” затронутой области памяти (в байтах) является значимой информацией для генерации текущего кода так же, как и его процентное соотношение с общей памятью.

4.5. Загрузка проекта в контроллер

4.5.1. Настройка соединения с контроллером

Контроллер подключается через gateway, который может быть вашим компьютером или сетевым компьютером, подключенным к контроллеру. Для определения пути соединения можно использовать диалог Установки соединения. Этот диалог открывается автоматически при попытке выполнения логина, если настройки связи еще не определены.

В открывшемся диалоге необходимо выполнить следующие шаги:

- 1) Выберите контроллер в дереве устройств и выполните команду «Проект»→«Редактировать объект». В результате контроллер откроется в редакторе;
- 2) Перейдите на вкладку «Установки соединения»;
- 3) В строке меню нажмите кнопку «Сканировать сеть». Откроется диалог «Выбор устройства». Слева будут показаны все устройства, доступные в сети (Рис. 4.13);
- 4) Выберите нужное устройство и нажмите «ОК». Будет задан путь соединения к контроллеру.

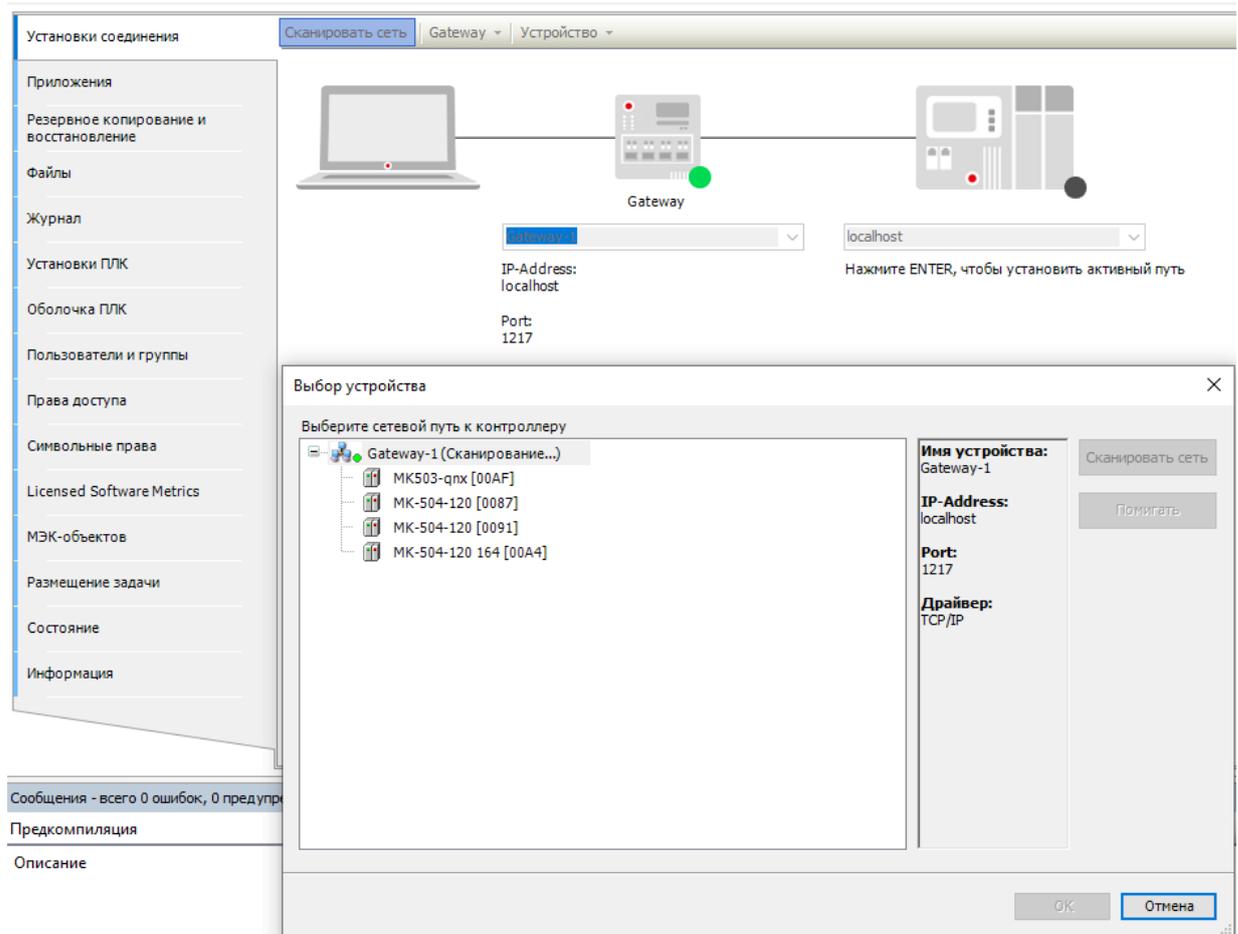


Рис. 4.13 - Окно выбора устройств

4.5.2. Загрузка прикладного кода — Логин и запуск ПЛК

Чтобы загрузить исходный код вашего приложения в контроллер, вы должны войти (выполнить логин) в контроллер с приложением. Если в проекте есть несколько приложений, необходимо сначала переключиться на правильное приложение (Рис. 4.14).

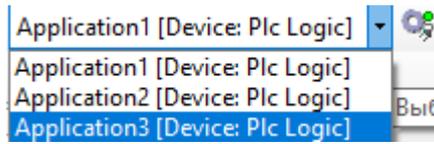


Рис. 4.14 - Окно выбора приложения

Когда вы загружаете приложение в контроллер, Procyon IDE выполняет следующие проверки:

- список приложений на контроллере сравнивается с приложениями, доступными в проекте. Если они не совпадают, вам будет предложено загрузить приложение, которого еще нет на контроллере, или удалить существующие приложения;

- для блоков с внешней реализацией в загружаемом приложении Procyon IDE проверяет их доступность на контроллере. Если они недоступны, то в диалоговом окне и в окне сообщений выдается сообщение о «неразрешенных ссылках». Затем Procyon IDE сравнивает параметры (переменные) блоков в загружаемом приложении и параметры одноименных блоков в приложении, которое существует на контроллере (проверка подписи). Если есть различия, то в диалоговом окне и в окне сообщений выдается сообщение о «недействительных подписях»;

- если в свойствах приложения выбран флажок «*Загрузить информацию о приложении*», в ПЛК загружается дополнительная информация о содержимом приложения;

- если для одного устройства существует несколько приложений, обратите внимание, что диалоговое окно «*Соотнесение входов/выходов*» содержит определение приложения для соотнесения входов/выходов используемого устройства.

Перенос приложений и запуск программ

Условие: Приложение не содержит ошибок, а параметры соединения контроллера определены корректно. Приложения пока не существует на контроллере.

- 1) Выберите нужное приложение в дереве устройств. Если у вас есть только одно приложение, переходите к шагу 3;

- 2) Нажмите «*Установить активное приложение*». Имя приложения будет выделено жирным шрифтом;

- 3) Выполните команду «*Онлайн*»→«*Логин*». Появится диалог, в котором нужно будет подтвердить создание приложения в контроллере;

- 4) Нажмите «*Да*». Приложение будет загружено в контроллер;

- 5) Выполните команду «*Отладка*»→«*Старт*» или нажмите F5. Приложение будет запущено на контроллере.

5. Работа с модулями MKLogic-500

5.1. Общие принципы работы с устройствами модулей ввода-вывода

Все модули ввода-вывода MKLogic-500 в дереве устройств как дочерние устройства Rack, который в свою очередь является дочерним к CAN. Такая структура является максимально приближенной к реальному расположению модулей в корзинах (см. Рис. 5.1).

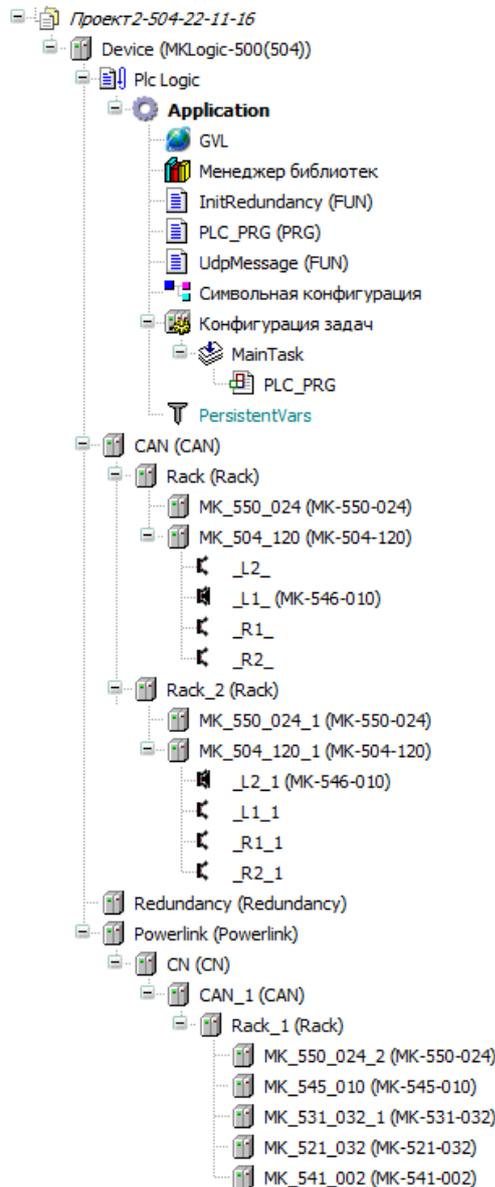


Рис. 5.1 - Пример структуры проекта

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Rack Offset	USINT(1..8)	1	1		Адрес корзины

Рис. 5.2 - Параметр смещение

Каждое устройство Rack имеет параметр Rack Offset в диапазоне 1..8 (см. Рис. 5.2). Это значение должно соответствовать положению многопозиционного переключателя ADDRESS на модуле питания данной корзины.

Смещение применяется ко всем дочерним модулям устройства.

Каждый модуль имеет два общих информационных параметра (только на чтение) Common и Powerlink multiplicity (см. Рис. 5.3). Оба параметра получают актуальные значения только в онлайн режиме.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
POWERLINK Id	USINT	240	240		Идентификатор POWERLINK
Rack Offset	USINT	1	1		Адрес корзины
Slot	USINT	1	1		Позиция в корзине
CAN Id	USINT	1	1		CANOpen-адрес
Powerlink multiplicity					Кратность PDO, применяется в Powerlink
TPDO_Mul	USINT	1	1		Коэффициент дробления при передаче через POWERLINK (к модулю)
RPDO_Mul	USINT	1	1		Коэффициент дробления при передаче через POWERLINK (от модуля)

Рис. 5.3 - Общие информационные параметры модулей

Описание структуры Common:

— POWERLINK Id идентификатор на шине Powerlink. Для модулей ввода-вывода, находящихся на одной шине CAN с модулями CPU, будет равен 240. Для остальных определяется настройками Powerlink;

— RackOffset номер корзины модуля (начиная с 1), дублирует параметр Rack Offset родительского Rack;

— Slot номер слота модуля в корзине (начиная с 1), следует учитывать что первые два слота зарезервированы под модули питания (даже если второй модуль питания отсутствует в корзине), нумерация всех остальных модулей начинается с 3. Нумерация автоматическая в соответствии с расположением устройства в дереве устройств сверху вниз;

— CAN Id результирующий адрес на шине CAN, вычисляется по формуле $(RackOffset - 1) * 16 + Slot$.

Параметр Powerlink multiplicity задает кратности при передаче через POWERLINK. Подробнее о кратностях см. в разделе «Дистанционные стойки подключенные по протоколу POWERLINK»

— TPDO_Mul Кратность при передаче через POWERLINK (данные от CPU к модулю);

— RPDO_Mul Кратность при передаче через POWERLINK (данные от модуля к CPU).

В Табл. 5.1 приведен список поддерживаемых модулей ввода-вывода.

Таблица 5.1 – Список поддерживаемых модулей

Имя устройства в дереве устройств	Наименование модуля	Назначение
МК-550-024	МК-550-024	Модуль питания напряжения постоянного тока с интерфейсом CAN
МК-505-120	МК-505-120	Модуль центрального процессора с 1 оптоволоконным интерфейсом резервирования, 2 интерфейсами Ethernet 100/1000 Base-T
МК-545-010	МК-545-010	Коммуникационный модуль с 1 портом Ethernet 100/1000 Base-T с поддержкой Powerlink с двумя хаб-выходами
МК-516-008A	МК-516-008A	Модуль аналогового ввода с 8 изолированными аналоговыми входами 0-20 (4 - 20) мА
МК-576-008A	МК-576-008A	Модуль аналогового ввода с 8 изолированными аналоговыми входами 0 - 20 (4 - 20) мА с поддержкой HART
МК-576-016A	МК-576-016A	Модуль аналогового ввода с 16 изолированными аналоговыми входами 0 - 20 (4 - 20) мА с поддержкой HART
МК-513-016A	МК-513-016A	Модуль аналогового ввода с 16 аналоговыми входами 0 - 20(4 - 20) мА
МК-514-008A	МК-514-008A	Модуль аналогового вывода с 8 аналоговыми
МК-574-008A	МК-574-008A	Модуль аналогового вывода с 8 аналоговыми выходами 0 - 20 (4 - 20) мА с поддержкой HART
МК-521-032	МК-521-032	Модуль дискретного ввода напряжения постоянного тока с 32 дискретными входами
МК-521-032A	МК-521-032A	Модуль дискретного ввода напряжения постоянного тока с 32 дискретными входами
МК-523-032A	МК-523-032A	Модуль дискретного ввода Namur с 32 дискретными входами

Имя устройства в дереве устройств	Наименование модуля	Назначение
МК-531-032	МК-531-032	Модуль дискретного вывода напряжения постоянного тока с 32 дискретными выходами
МК-531-032А	МК-531-032А	Модуль дискретного вывода напряжения постоянного тока с 32 дискретными выходами
МК-532-032А	МК-532-032А	Модуль дискретного вывода Namur с 32 дискретными выходами
МК-541-002	МК-541-002	Коммуникационный модуль с 2 интерфейсами RS-485

5.1.1. Канал диагностической информации модулей

Все модули ввода-вывода имеют диагностический входной канал Diagnostics. Диагностический канал предназначен для получения идентификационной информации о модуле, состоянии его CAN-шин и о том, совместим ли реально подключенный модуль с модулем, добавленным в проект. Диагностический канал представляет собой переменную-структуру типа IOModuleDiag. Поля структуры приведены на Рис. 5.4.

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%iD0			CAN диагностика
		VendorID	%iD0	DWORD		Код производителя
		ProductCode	%iD1	DWORD		Код продукта
		RevisionNumber	%iD2	DWORD		Номер ревизии
		SerialNumber	%iD3	DWORD		Серийный номер
		DeviceType	%iD4	DWORD		Тип устройства
		ManufactureStatus	%iD5	DWORD		Статус производителя
		CRC32	%iD6	DWORD		CRC32
		DeviceName	%iB28	ARRAY [1..33] OF SINT		Имя устройства
		SwVersion	%iB61	ARRAY [1..17] OF SINT		Версия программного обеспечения
		HwVersion	%iB78	ARRAY [1..17] OF SINT		Версия аппаратного обеспечения
		CurrentCANBus	%iB95	USINT		Номер текущей шины CAN
		CAN1heartbeat	%iB96	USINT		Последний Heartbeat модуля на шине CAN1
		CAN2heartbeat	%iB97	USINT		Последний Heartbeat модуля на шине CAN2
		State	%iB98	USINT		Код состояния модуля
		IsCompatible	%iB99	USINT		1 - если модуль совместим с требуемым
		IsPresent	%iB100	USINT		1 - если модуль присутствует на шине

Рис. 5.4 - Диагностический канал модулей

Пояснения к полям структуры IOModuleDiag:

- поля с VendorID по HwVersion служат для идентификации модуля;
- поле CurrentCANbus принимает значение 0, если рабочая CAN-шина не определена, 1 для текущей шины CAN1 и 2 для текущей шины CAN2;
- поля CAN1heartbeat и CAN2heartbeat принимают значения 127, если модуль изделия не в рабочем состоянии по данной шине, 5 если модуль изделия инициализован и в рабочем состоянии и 0, если модуль отсутствует на шине;
- поле IsCompatible принимает значение TRUE, если тип реально установленного в данной стойке и позиции модуля изделия совпадает с заданным в конфигурации;
- поле State содержит код, соответствующий состоянию светодиодной индикации модуля Табл. 5.2;
- поле Crc32 содержит контрольную сумму метрологически значимой части ПО модуля (для модулей аналогового ввода-вывода).

Таблица 5.2 – Расшифровка кодов состояния модуля ввода-вывода

Код	Состояние светодиода Error	Состояние модуля ввода-вывода
0	Не горит	Нет ошибок
1	Горит	Отсутствует внешнее напряжение (только для модулей питания в конфигурации с двумя модулями питания)
2	Мигает	Положения переключателей адреса и скорости модулей питания корректные, но не совпадают (только для модулей питания в конфигурации с двумя модулями питания)
3	Мигает	Отсутствует связь с модулем питания слева (только для модулей питания)
4	Мигает	Недопустимые положения переключателей адреса или скорости модулей питания (только для модулей питания)
5	Мигает	Отсутствует связь с модулем ввода-вывода слева (только для модулей ввода-вывода)

Код	Состояние светодиода Error	Состояние модуля ввода-вывода
6	Мигает	Недопустимые положения переключателей адреса или скорости модулей питания (для всех модулей)
7	Горит	Внутренний отказ модуля.

5.2. Общие принципы работы с дополнительными устройствами

К дополнительным устройствам в данном документе относятся устройства Procyon IDE, доступные к добавлению в дерево устройств и не относящиеся к модулям изделия (Табл. 5.1). Их общая черта – отсутствие привязки к физическому устройству в составе корзины. Дополнительные устройства ввода-вывода можно поделить на несколько групп:

- устройства, реализующие поддержку протокола Modbus RTU/TCP в модулях CPU;
- устройства, реализующие поддержку протокола IEC 60870-5-104 в модулях CPU;
- устройства, реализующие поддержку протокола OPC UA;
- устройства, реализующие поддержку протокола Powerlink;
- устройства, реализующие поддержку резервирования модулей CPU.

Таблица 5.3 – Дополнительные устройства

Наименование дополнительного устройства	Функциональное назначение устройства
NftModbus	Поддержка протокола Modbus
NftModbusTCP	Поддержка протокола Modbus TCP
NftModbusSlave	Реализация серверной части Modbus
NftModbusMaster	Реализация клиентской части Modbus
Powerlink	Поддержка протокола Powerlink
CN	Реализация роли CN протокола Powerlink
Redundancy	Поддержка резервирования
МК-544-040	Поддержка коммуникационного модуля МК-544-040 (4 порта Ethernet)
МК-546-010	Поддержка коммуникационного модуля МК-546-010 (1 порт Powerlink)

В Табл. 5.3 приведён перечень доступных дополнительных устройств и их функциональное назначение.

5.3. Модули питания МК-550-024

В каждом устройстве Rack должен присутствовать один или два модуля питания. При добавлении корзины один модуль питания добавляется автоматически и его нельзя удалить. При необходимости в корзину можно добавить резервный модуль питания. Модули питания всегда занимают первое и второе положение в корзине и их невозможно передвинуть в другое положение.

Кроме переменной диагностики, устройство МК-550-024 имеет в своем составе переменную с диагностикой модуля питания PSU diagnostics (Рис. 5.5).

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%ID0			CAN диагностика
		PSU diagnostics	%ID26			
		Supply voltage	%ID26	REAL	В	Входное напряжение
		Self ID	%IW54	UINT		ID модуля
		Error code	%IW55	UINT		Код ошибки работы модуля
		Can Bus Speed	%IW56	UINT	кБит/с	Скорость CAN-шины
		Can1 Error Counter	%IW57	UINT		Счетчик ошибок шины CAN1
		Can2 Error Counter	%IW58	UINT		Счетчик ошибок шины CAN2

Рис. 5.5 - Описание структуры данных PSU Diagnostics

В поле errorCode модуль возвращает битную маску ошибок в своей работе. Расшифровка кодов ошибок работы модуля приведена в Табл. 5.4.

Таблица 5.4 – Расшифровка кодов ошибок errorCode структуры PSU diagnostics

Наименование дополнительного устройства	Функциональное назначение устройства
0	1 – пониженное напряжение 5В
1	1 – переключатель адреса CAN (ADDRESS) в запрещённом положении
2	1 – переключатель скорости CAN (BITRATE) в запрещённом положении
3.15	Не используются

5.4. Модули центрального процессора МК-504-120

Кроме диагностического канала, устройство МК-504-120 имеют в своём составе следующие каналы:

- Канал с диагностикой модуля центрального процессора CPU diagnostics (Рис. 5.6);
- Канал с расширенной диагностикой модуля центрального процессора Ext diagnostics (Рис. 5.7);
- Канал с диагностикой резервирования модуля центрального процессора Redundancy diagnostics (Рис. 5.8).

Переменная	Соотнесе...	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%ID30			CAN диагностика
		CPU diagnostics	%ID56			Диагностика CPU diagnostics
		CpuLoad	%ID56	REAL	%	Загрузка CPU
		MemoryFree	%IW114	WORD	МБ	Свободная память
		EthernetPorts	%IB230			Диагностика портов Ethernet
		Values	%IB230	ARRAY [1..20] OF EthernetPortDiag		
		Values[1]	%IB230			
		Present	%IB230	USINT		0 - Порт отсутствует, 1 - Порт присутствует
		Link	%IB231	USINT		0 - не подключен, 1 - подключен, порт активен
		Values[2]	%IB232			
		Values[3]	%IB234			
		Values[4]	%IB236			
		Values[5]	%IB238			
		Values[6]	%IB240			
		Values[7]	%IB242			
		Values[8]	%IB244			
		Values[9]	%IB246			
		Values[10]	%IB248			
		Values[11]	%IB250			
		Values[12]	%IB252			
		Values[13]	%IB254			
		Values[14]	%IB256			
		Values[15]	%IB258			
		Values[16]	%IB260			
		Values[17]	%IB262			
		Values[18]	%IB264			
		Values[19]	%IB266			
		Values[20]	%IB268			
		UplinkPortIndex	%IW135	INT		Индекс порта с ролью Uplink в ethernetPots; 0 - нет порта с ролью Uplink
		DatalinkPortIndex	%IW136	INT		Индекс порта с ролью Datalink в ethernetPots; 0 - нет порта с ролью Datalink

Рис. 5.6 - Описание структуры данных CPU diagnostics

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%ID30			CAN диагностика
		CPU diagnostics	%ID56			Диагностика CPU diagnostics
		Ext diagnostics	%IB276			Расширенная диагностика
		SFP	%IB276			Статус SFP
		Present	%IX276.0	BOOL		Статус наличия SFP
		Dist	%IB277	SINT		Характеристика расстояния SFP
		Version	%IB278	ARRAY [1..17] OF SINT		Версия SFP
		Redundancy diagnostics	%IB295			Диагностика резервирования

Рис. 5.7 - Описание структуры данных Ext diagnostics

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%ID30			CAN диагностика
		CPU diagnostics	%ID56			Диагностика CPU diagnostics
		Ext diagnostics	%IB276			Расширенная диагностика
		Redundancy diagnostics	%IB295			Диагностика резервирования
		IsActive	%IB295	USINT		Состояние текущего контроллера по резервированию
		IsValid	%IB296	USINT		Состояние валидности текущего контроллера по резервированию
		IsPrimary	%IB297	USINT		Роль текущего контроллера по положению в корзине

Рис. 5.8 - Описание структуры данных Redundancy diagnostics

Всем Ethernet портам модуля центрального процессора МК-504-120 (включая порты на модулях расширения МК-544-040 и МК-546-010) в среде программирования Procyon IDE присвоено условное обозначение: Eth1..Eth20.

В Табл. 5.5 и на Рис. 5.9, 5.10 отображено соответствие значений идентификатора порта физическим портам модулей CPU.

Таблица 5.5 – Таблица соответствия аппаратных портов Ethernet наименованиям портов в Procyon IDE

Слот PCI-шины	Аппаратный порт	Наименование порта в Procyon IDE
Центральный слот (CPU)	Порт SFP интерфейса Ethernet Порт ETH1 интерфейса Ethernet Порт ETH2 интерфейса Ethernet	Eth1 Eth2 Eth3
Слот расширения L2	Порт ETH1 интерфейса Ethernet / Порт Powerlink модуля МК-546-010 Порт ETH2 интерфейса Ethernet Порт ETH3 интерфейса Ethernet Порт ETH4 интерфейса Ethernet	Eth5 Eth6 Eth7 Eth8
Слот расширения L1	Порт ETH1 интерфейса Ethernet / Порт Powerlink модуля МК-546-010 Порт ETH2 интерфейса Ethernet Порт ETH3 интерфейса Ethernet Порт ETH4 интерфейса Ethernet	Eth9 Eth10 Eth11 Eth12
Слот расширения R1	Порт ETH1 интерфейса Ethernet / Порт Powerlink модуля МК-546-010 Порт ETH2 интерфейса Ethernet Порт ETH3 интерфейса Ethernet Порт ETH4 интерфейса Ethernet	Eth13 Eth14 Eth15 Eth16
Слот расширения R2	Порт ETH1 интерфейса Ethernet / Порт Powerlink модуля МК-546-010 Порт ETH2 интерфейса Ethernet Порт ETH3 интерфейса Ethernet Порт ETH4 интерфейса Ethernet	Eth17 Eth18 Eth18 Eth20

В случае МК-504-120 следует отметить, что слоты расширения имеют нумерацию от центра, чтобы наименования слотов расширения не менялось для задних шин МК-5-BUSe3 (2 слота расширения) и МК-5-BUSe5 (4 слота расширения). При этом нумерация слотов расширения для устройств МК-544-040 и МК-546-010 – слева направо Рис. 5.9, 5.10.

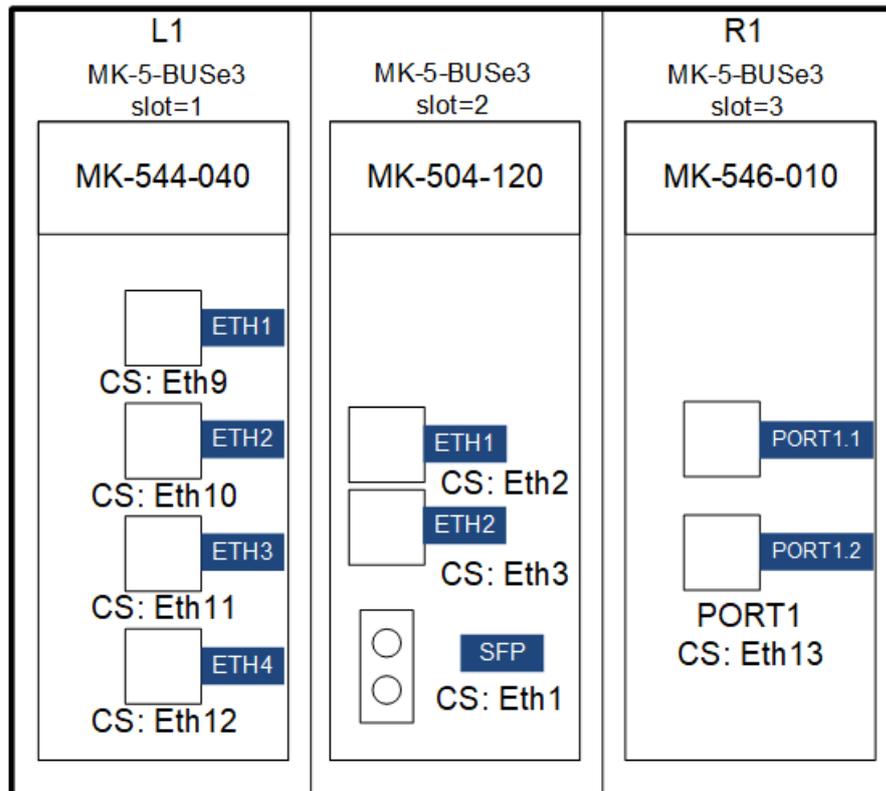


Рис. 5.9 - Пример нумерации слотов и портов на шине МК-5-BUSe3

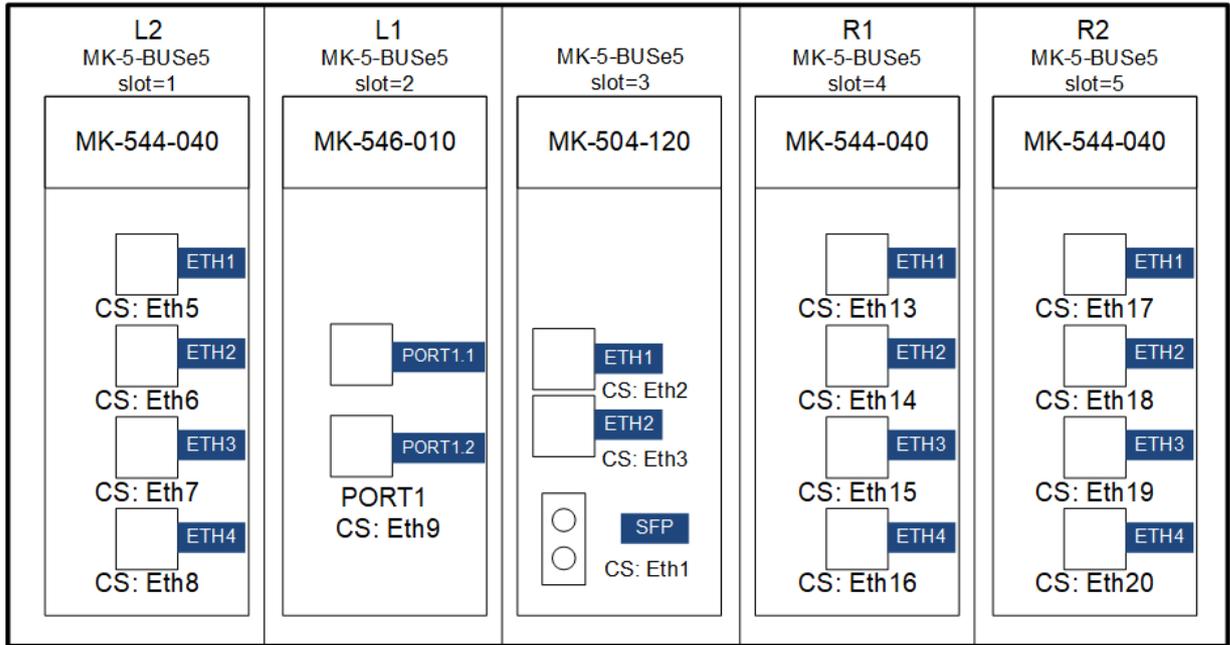


Рис. 5.10 - Пример нумерации слотов и портов на шине MK-5-BUSe5

Расположение модулей MK-544-040 и MK-546-010 в слотах дано условно, ограничений на размещение в слотах расширений нет. Единственное ограничение – поддерживается только один модуль MK-546-010.

5.4.1. Коммуникационный модуль MK-546-010

Устройство MK-546-010 предназначено для получения диагностической информации по модулю MK-546-010. Модуль имеет один канал диагностической информации Powerlink diagnostics (Рис. 5.11).

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Powerlink diagnostics	%IB298			Powerlink diagnostics
		Ports	%IB298	ARRAY [1..2] OF PowerlinkPortDiag		Диагностика Powerlink-портов
		Ports[1]	%IB298			Диагностика Powerlink-портов
		Linkup	%IB298	USINT		Наличие физического подключения на порту
		Full Duplex	%IB299	USINT		Режим работы: 1 - полный дуплекс, 0 - полудуплекс
		Speed	%IB300	USINT	МБит/с	Скорость Ethernet: 0 - 10, 1 - 100, 2 - 1000
		Port Blocking	%IB301	USINT		Статус блокировки
		Remote RX Failure	%IB302	USINT		Отсутствует прием на удаленном порту
		RX Failure	%IB303	USINT		Отсутствует прием на порту
		Loopback	%IB304	USINT		Наличие на порту петли
		Multiple Errors	%IB305	USINT		Множественные ошибки на порту
		Diag Errors Counter	%IB306	USINT		Счетчик ошибок диагностического канала
		Ports[2]	%IB307			Диагностика Powerlink-портов
		Leds	%IB316			Состояние светодиодов передней панели модуля
		Line	%IB316	USINT		Светодиод Line включен, Powerlink в режиме Line
		Ring	%IB317	USINT		Светодиод Ring включен, Powerlink в режиме Ring
		Eth	%IB318	USINT		Светодиод Eth включен, порт в режиме Ethernet
		Pwl	%IB319	USINT		Светодиод Pwl включен, порт в режиме Powerlink
		Version	%IB320	ARRAY [1..17] OF SINT		Аппаратная версия Powerlink

Рис. 5.11 - Описание структуры данных Powerlink diagnostics

Поля PortBlocking, RemoteRXFailure, RXFailure, Loopback, MultipleErrors и DiagErrorsCounter в структуре PowerlinkPortDiag предназначены для использования в будущем и не реализованы; их значения следует игнорировать.

5.4.2. Коммуникационный модуль MK-544-040

Устройство MK-544-040 предназначено для получения диагностической информации по модулю MK-544-040. Модуль имеет один канал диагностической информации MK-544-040 diagnostics (Рис. 5.12).

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		MK-544-040 diagnostics	%IB337			MK-544-040 диагностика
		MK-544-040 Ethernet Ports	%IB337	ARRAY [1..4] OF EthernetPortDiag		MK-544-040 Ethernet-порты
		MK-544-040 Ethernet Ports[1]	%IB337			MK-544-040 Ethernet-порты
		Present	%IB337	USINT		0 - Порт отсутствует, 1 - Порт присутствует
		Link	%IB338	USINT		0 - не подключен, 1 - подключен, порт активен
		MK-544-040 Ethernet Ports[2]	%IB339			MK-544-040 Ethernet-порты
		MK-544-040 Ethernet Ports[3]	%IB341			MK-544-040 Ethernet-порты
		MK-544-040 Ethernet Ports[4]	%IB343			MK-544-040 Ethernet-порты
		Version	%IB345	ARRAY [1..17] OF SINT		MK-544-040 аппаратная версия

Рис. 5.12 - Описание структуры данных МК-544-040 diagnostics

5.5. Модули аналогового ввода МК-513-016 и МК-513-016А

Примечание Модули дискретного ввода МК-513-016 и МК-513-016А отличаются только конструктивно и совершенно не отличаются в работе.

Кроме диагностического канала, модули МК-513-016 и МК-513-016А имеют в своём составе входной канал AI данных значений каналов аналогового ввода модуля, представляющий массив из 16 элементов типа UINT.

Значение каждого элемента массива устройства МК-513-016 соответствует коду АЦП соответствующего входа модуля. Код АЦП канала изменяется линейно, изменение на 1 мА соответствует изменению кода АЦП на 660 единиц. Расшифровка кодов АЦП модуля аналогового ввода приводится в Табл. 5.6.

Таблица 5.6 – Расшифровка значений входов устройства МК-513-016

Значения кода АЦП канала	Величина входного тока канала, мА
0	0,0
2640	4,0
13200	20,0
16383	24,82

Также имеется конфигурационный канал для настройки параметров каналов аналогового ввода модуля. Он представляется собой массив из 16 элементов структурного типа AIChannelParameters Рис. 5.13.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
AI Config	ARRAY [1..16] OF AIChannelParameters				Конфигурация каналов аналогового ввода
AI Config[1]					Конфигурация каналов аналогового ввода
Mode	Enumeration of USINT	4-20 mA	4-20 mA		Режим входа: 0 - 4..20mA, 1 - 0..20mA
Enable	BOOL	TRUE	TRUE		Разрешение входа
LPFilterEnable	BOOL	FALSE	FALSE		Разрешение ФНЧ
LPFilterTimeConstant	DWORD	0	0	мс	Постоянная времени ФНЧ
AI Config[2]					Конфигурация каналов аналогового ввода
AI Config[3]					Конфигурация каналов аналогового ввода
AI Config[4]					Конфигурация каналов аналогового ввода
AI Config[5]					Конфигурация каналов аналогового ввода
AI Config[6]					Конфигурация каналов аналогового ввода
AI Config[7]					Конфигурация каналов аналогового ввода
AI Config[8]					Конфигурация каналов аналогового ввода
AI Config[9]					Конфигурация каналов аналогового ввода
AI Config[10]					Конфигурация каналов аналогового ввода
AI Config[11]					Конфигурация каналов аналогового ввода
AI Config[12]					Конфигурация каналов аналогового ввода
AI Config[13]					Конфигурация каналов аналогового ввода
AI Config[14]					Конфигурация каналов аналогового ввода
AI Config[15]					Конфигурация каналов аналогового ввода
AI Config[16]					Конфигурация каналов аналогового ввода
Powerlink multiplicity					Кратность PDO, применяется в Powerlink

Рис. 5.13 - Конфигурация каналов аналогового ввода модуля МК-513-016

Параметры каналов устройства МК-513-016 (структура AIChannelParameters):

— Mode – режим работы подключаемого к аналоговому входу датчику. 0 – 4-20 мА, 1 – 0-20 мА. Влияет только на порог срабатывания индикации обрыва на передней панели модуля аналогового ввода МК-513-016, измерение текущего значения тока при обоих режимах работы производится от 0 мА. По умолчанию все каналы работают в режиме 4-20 мА;

- Enabled – разрешение работы канала. FALSE – запрещён, TRUE – разрешён. Запрещённый канал постоянно возвращает код ЦАП равный 0. По умолчанию все каналы разрешены;
- LPFilterEnable – разрешение работы фильтра низких частот на канале;
- LPFilterTimeConstant – постоянная времени фильтра низких частот, от 3 до 10000 мс. Имеет значение только при LPFilterEnable=TRUE. По умолчанию постоянная времени на всех каналах равна 20 мс.

5.6. Модули аналогового вывода МК-514-008 и МК-514-008А

 **Примечание** Модули дискретного ввода МК-514-008 и МК-514-008А отличаются только конструктивно и совершенно не отличаются в работе.

Кроме диагностического канала, модули МК-514-008 и МК-514-008А имеют в своём составе:

- выходной канал АО данных значений каналов аналогового вывода модуля, представляющий массив из 8 элементов типа UINT;
- входной канал АО Statuses статусов каналов аналогового вывода модуля, представляющий массив из 8 элементов типа USINT.

Значение каждого элемента массива устройства МК-514-008 соответствует коду ЦАП соответствующего выхода модуля. Код ЦАП канала изменяется линейно, изменение на 1 мА соответствует изменению кода ЦАП на 2730,625 единиц. Расшифровка кодов ЦАП модуля аналогового вывода приводится в Табл. 5.7.

Таблица 5.7 – Расшифровка значений выходов устройства МК-514-008

Значения кода ЦАП канала	Величина входного тока канала, мА
0	0,0
10922	4,0
54613	20,0

Также имеется конфигурационный канал для настройки параметров каналов аналогового вывода модуля. Он представляется собой массив из 8 элементов структурного типа AOChannelParameters Рис. 5.14.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
AO Config	ARRAY [1..8] OF AOChannelParameters				Конфигурация каналов аналогового вывода
AO Config[1]					Конфигурация каналов аналогового вывода
Mode	Enumeration of USINT	4-20 mA	4-20 mA		Режим входа: 0 - 4..20mA, 1 - 0..20mA
Enable	BOOL	TRUE	TRUE		Разрешение выхода
ErrorMode	BOOL	FALSE	FALSE		Поведение при остановке/отказе CPU: 0 - удерживать предыдущее значение, 1 - применить значение 'ErrorValue'
ErrorValue	WORD	0	0		Подстановочное значение при 'ErrorMode' = 1
AO Config[2]					Конфигурация каналов аналогового вывода
AO Config[3]					Конфигурация каналов аналогового вывода
AO Config[4]					Конфигурация каналов аналогового вывода
AO Config[5]					Конфигурация каналов аналогового вывода
AO Config[6]					Конфигурация каналов аналогового вывода
AO Config[7]					Конфигурация каналов аналогового вывода
AO Config[8]					Конфигурация каналов аналогового вывода
Powerlink multiplicity					Кратность PDO, применяется в Powerlink

Рис. 5.14 - Конфигурация каналов аналогового вывода модуля МК-514-008

Параметры каналов устройства МК-514-008 (структура AOChannelParameters):

- Mode – режим работы аналогового выхода. 0 – 4-20 мА, 1 – 0-20 мА. В режиме 4-20 мА при задании кода ЦАП ниже значения 10922 (4 мА) выходной ток канала всегда составляет 4 мА, в режиме 0-20 мА выходной ток канала строго соответствует заданному значению. По умолчанию все каналы работают в режиме 4-20 мА;
- Enabled – разрешение работы канала. FALSE – запрещён, TRUE – разрешён. Выходной ток запрещённого канала равен последнему заданному значению для уже работающего и 0 мА для не инициализированного модуля аналогового вывода. По умолчанию все каналы разрешены;
- ErrorMode – режим работы канала при потере модулем аналогового вывода связи с модулем CPU. FALSE – при потере связи с модулем CPU фиксировать значение тока канала, TRUE – присваивать коду ЦАП модуля значение параметра ErrorValue. По умолчанию все каналы имеют ErrorMode=FALSE;
- ErrorValue – значение кода ЦАП канала в режиме работы ErrorMode=TRUE при потере модулем аналогового вывода связи с CPU. По умолчанию ErrorValue всех каналов равны 0.

Значения канала AO Statuses соответствуют состоянию электрических цепей соответствующих каналов модуля аналогового вывода. Код состояния канала имеет тип USINT (значение по умолчанию 0).

Таблица 5.8 – Расшифровка статусов каналов аналогового выхода

Код ошибки статуса	Состояние канала аналогового выхода
0	нет ошибок
1	разрыв цепи
2	нет внешнего питания

5.7. Модули аналогового ввода МК-516-008 и МК-516-008А

 **Примечание** Модули дискретного ввода МК-516-008 и МК-516-008А отличаются только конструктивно и совершенно не отличаются в работе.

Кроме диагностического канала, модули МК-516-008 и МК-516-008А имеют в своём составе входной канал AI данных значений каналов аналогового ввода модуля, представляющий массив из 8 элементов типа UINT.

Значение каждого элемента массива канала AI МК-513-016 соответствует коду АЦП соответствующего входа модуля. Код АЦП канала изменяется линейно, изменение на 1 мА соответствует изменению кода АЦП на 2621,4 единиц. Расшифровка кодов АЦП модуля аналогового ввода приводится в Табл. 5.9.

Таблица 5.9 – Расшифровка значений входов устройства МК-516-008

Значения кода АЦП канала	Величина входного тока канала, мА
0	0,0
10486	4,0
52428	20,0
65535	25,0

Также имеется конфигурационный канал для настройки параметров каналов аналогового ввода модуля. Он представляется собой массив из 8 элементов структурного типа AIChannelParameters Рис. 5.15.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
AI Config	ARRAY [1..8] OF AIChannelParameters				Конфигурация каналов аналогового ввода
AI Config[1]					Конфигурация каналов аналогового ввода
Mode	Enumeration of USINT	4-20 mA	4-20 mA		Режим входа: 0 - 4..20mA, 1 - 0..20mA
Enable	BOOL	TRUE	TRUE		Разрешение входа
LPFilterEnable	BOOL	FALSE	FALSE		Разрешение ФНЧ
LPFilterTimeConstant	DWORD	0	0	мс	Постоянная времени ФНЧ
AI Config[2]					Конфигурация каналов аналогового ввода
AI Config[3]					Конфигурация каналов аналогового ввода
AI Config[4]					Конфигурация каналов аналогового ввода
AI Config[5]					Конфигурация каналов аналогового ввода
AI Config[6]					Конфигурация каналов аналогового ввода
AI Config[7]					Конфигурация каналов аналогового ввода
AI Config[8]					Конфигурация каналов аналогового ввода

Рис. 5.15 - Конфигурация каналов аналогового ввода модуля МК-516-008

Параметры каналов устройства МК-516-008 аналогичны параметрам модуля МК-513-016 описанным выше.

5.8. Модули дискретного ввода МК-521-032 и МК-521-032А

 **Примечание** Модули дискретного ввода МК-521-032 и МК-521-032А отличаются только конструктивно и совершенно не отличаются в работе.

Кроме диагностического канала, модули МК-521-032 и МК-521-032А имеют в своём составе:

- входной канал DI данных значений каналов дискретного ввода модуля, представляющий массив из 32 элементов типа BOOL;
- входной канал DI trigger флагов срабатывания триггера каналов дискретного ввода модуля, представляющий массив из 32 элементов типа BOOL;
- выходной канал DI trigger reset управления сбросом флагов триггера, представляющий массив из 32 элементов типа BOOL;
- входной канал DI History предназначен для получения изменявшихся значений дискретных входов за время последнего цикла программы пользователя (см. Рис. 5.17). В поле Values структуры DI32WithTimeStamp хранятся значения всех 32 дискретных входов (начиная с самого «старого»). Метки времени изменения входов фиксируется в поле TimeStamp. В следующей после завершающей записи в устройстве обнуляется поле .TimeStamp

Также имеется конфигурационный канал для настройки параметров каналов дискретного ввода модуля. Он представляется собой массив из 8 элементов структурного типа DIChannelParameters Рис. 5.16.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
DI Config	ARRAY [1..32] OF DIChannelParameters				Конфигурация каналов дискретного ввода
DI Config[1]					Конфигурация каналов дискретного ввода
Mode	Enumeration of USINT	Discrete	Discrete		Режим входа: 0 - Дискретный вход, 1 - Счетчик, 2 - Частотомер
Enable	BOOL	TRUE	TRUE		Разрешение входа
Polarity	BOOL	FALSE	FALSE		Режим инверсии (1 - значение TRUE при обесточенном входе)
Bounce Time	UDINT(0..100000)	0	0	мкс	Время антидребезга
DI Config[2]					Конфигурация каналов дискретного ввода
DI Config[3]					Конфигурация каналов дискретного ввода
DI Config[4]					Конфигурация каналов дискретного ввода
DI Config[5]					Конфигурация каналов дискретного ввода
DI Config[6]					Конфигурация каналов дискретного ввода
DI Config[7]					Конфигурация каналов дискретного ввода
DI Config[8]					Конфигурация каналов дискретного ввода
DI Config[9]					Конфигурация каналов дискретного ввода
DI Config[10]					Конфигурация каналов дискретного ввода

Рис. 5.16 - Конфигурация каналов дискретного ввода модуля МК-521-032

History DI	%ID203				Сохраненные значения 32 каналов дискретного ввода с меткой времени
History DI	%ID203	ARRAY [1..32] OF DI32WithTimeStamp			
History DI[1]	%ID203				
TimeStamp	%ID203				
sec_unix	%ID203	DINT			Число секунд с 1970/01/01
sec_2000	%ID204	DINT			Число секунд с 2000/01/01
msec	%ID205	DINT			Число миллисекунд в текущей секунде
usec	%ID206	DINT			Число микросекунд в текущей секунде
lastScan_msec	%ID207	DINT			Время выполнения последнего скана, в миллисекундах
current_date	%ID208	DATE			Текущая дата в формате DATE
current_time	%ID209	TIME			Текущее время в формате TIME
Values	%IB840	ARRAY [1..32] OF USINT			
Values[1]	%IB840	USINT			
Values[2]	%IB841	USINT			

Рис. 5.17 - Описание структуры DI32WithTimeStamp

Параметры каналов устройства МК-521-032 (структура DIChannelParameters):

- Mode – режим входа. 0 - Discrete (Дискретный вход), 1 - Counter (Счетчик), 2 - Frequency (Частотомер). Зарезервировано на будущее. В данный момент поддерживается только ‘Дискретный вход’;
- Enabled – разрешение работы канала. FALSE – запрещён, TRUE – разрешён. Для запрещённого канала элементы массивов DI и DI trigger всегда равны FALSE. По умолчанию все каналы разрешены;
- Invert – режим инверсии поля канала DI. FALSE – инверсия выключена, TRUE – инверсия включена. По умолчанию инверсия всех каналов выключена;
- BounceTime – минимально допустимая длина входного сигнала на канале, от 0 до 100000 мкс. Входной сигнал длительностью меньше значения BounceTime игнорируется. Данный параметр используется для защиты входов от дребезга. По умолчанию значение BounceTime всех каналов равно 0.

5.9. Модули дискретного вывода МК-531-032 и МК-531-032A

 **Примечание** Модули дискретного ввода МК-531-032 и МК-531-032A отличаются только конструктивно и совершенно не отличаются в работе.

Кроме диагностического канала, модули МК-531-032 и МК-531-032А имеют в своём составе:

- выходной канал DQ данных значений каналов дискретного вывода модуля, представляющий массив из 32 элементов типа BOOL;
- входной канал PWM значений ШИМ выходного сигнала канала модуля, представляющий массив из 32 элементов типа USINT.

В режиме ШИМ может работать 8 каналов устройства МК-531-032 (1, 2, 3, 8, 17, 18, 19, 24), при этом несущую частоту можно задавать только для групп каналов (Рис. 5.18). Допустимые значения частоты модуляции ШИМ – от 2 до 250 Гц. Переменная выхода управления pwm может принимать значения от 0 до 255, где 0 - 0% диапазона изменений, 255 - 100%. По умолчанию значение частоты ШИМ равно 0 Гц.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
DO Config	ARRAY [1..32] O...				Конфигурация каналов дискретного вывода
PWM Config					Конфигурация каналов ШИМ
PWM1_Freq	USINT	0	0		Частота модуляции ШИМ для каналов 1, 2, 17 и 18 (0..250Гц)
PWM2_Freq	USINT	0	0		Частота модуляции ШИМ для каналов 3 и 9 (0..250Гц)
PWM3_Freq	USINT	0	0		Частота модуляции ШИМ для каналов 8 и 24 (0..250Гц)
Powerlink multiplicity					Кратность PDO, применяется в Powerlink

Рис. 5.18 - Конфигурация несущей частоты каналов ШИМ модуля МК-531-032

Также имеется конфигурационный канал для настройки параметров каналов дискретного вывода модуля. Он представляется собой массив из 8 элементов структурного типа DOChannelParameters Рис. 5.19.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
DO Config	ARRAY [1..32] OF DOChannelParameters				Конфигурация каналов дискретного вывода
DO Config[1]					Конфигурация каналов дискретного вывода
Mode	Enumeration of USINT	Discrete	Discrete		Режим работы: 0 - дискретный выход, 1 - ШИМ
Enabled	BOOL	TRUE	TRUE		Разрешение выхода
Invert	BOOL	FALSE	FALSE		Режим инверсии (1 - выход выключен при значении TRUE)
Error mode	BOOL	FALSE	FALSE		Поведение при остановке/отказе CPU: 0 - удерживать предыдущее значение, 1 - применить значение 'ErrorValue'
Error value	BOOL	FALSE	FALSE		Подстановочное значение при 'ErrorMode' = 1
DO Config[2]					Конфигурация каналов дискретного вывода
DO Config[3]					Конфигурация каналов дискретного вывода
DO Config[4]					Конфигурация каналов дискретного вывода
DO Config[5]					Конфигурация каналов дискретного вывода
DO Config[6]					Конфигурация каналов дискретного вывода
DO Config[7]					Конфигурация каналов дискретного вывода
DO Config[8]					Конфигурация каналов дискретного вывода

Рис. 5.19 - Конфигурация каналов дискретного ввода модуля МК-531-032

— Mode – режим входа. 0 - Discrete (Дискретный вход), 1 - PWM (ШИМ). Режим ШИМ можно включить только для каналов с номерами 1, 2, 3, 8, 17, 18, 19 и 24. По умолчанию все каналы работают в режиме дискретного выхода;

— Enabled – разрешение работы канала. FALSE – запрещён, TRUE – разрешён. Выходной сигнал запрещённого канала всегда равен FALSE. По умолчанию все каналы разрешены;

— Invert – режим инверсии поля значений канала DO. FALSE – инверсия выключена, TRUE – инверсия включена. По умолчанию инверсия всех каналов выключена;

— ErrorMode – режим работы канала при потере модулем дискретного вывода связи с модулем CPU. FALSE – при потере связи с модулем CPU фиксировать значение канала, TRUE – присваивать каналу модуля значение параметра ErrorValue. Режим ErrorMode работает только для каналов в режиме дискретного выхода. По умолчанию все каналы имеют ErrorMode=FALSE;

— ErrorValue – значение канала в режиме работы ErrorMode=TRUE при потере модулем дискретного вывода связи с CPU. По умолчанию ErrorValue всех каналов равен FALSE.

5.10. Модули дискретного ввода МК-523-032А

Кроме диагностического канала, модули МК-523-032А имеют в своём составе:

— входной канал DI данных значений каналов дискретного ввода модуля, представляющий массив из 32 элементов типа DI32namurChannelValue;

Входные каналы устройства в структуре DI32NamurChannelValue возвращают текущее значение дискретного входа в поле Input и статусы в поле Status.

Имя	Тип	Наследо...	Адрес	Начальн.	Комментарий
IODiag	IOModuleDiag				Общая диагностическая информация модуля на шине CAN
DI	DI32NamurValueType				Значения каналов дискретного ввода

Рис. 5.20 - Каналы дискретного ввода модуля МК-523-032А

Также имеется конфигурационный канал для настройки параметров каналов дискретного ввода модуля. Он представляется собой массив из 8 элементов структурного типа

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
POWERLINK Id	USINT	240	240		Идентификатор POWERLINK
Rack Offset	USINT	1	1		Адрес корзины
Slot	USINT	1	1		Позиция в корзине
CAN Id	USINT	1	1		CANOpen-адрес
DI Config	ARRAY [1..32] OF DINamurChannelParameters				Конфигурация каналов дискретного ввода namur
DI Config[1]					Конфигурация каналов дискретного ввода namur
Enable	BOOL	TRUE	TRUE		Разрешение входа
Polarity	BOOL	FALSE	FALSE		Режим инверсии (1 - значение TRUE при обесточенном входе)
Bounce Time	UDINT(0..100000)	0	0	мкс	Время антидребезга
DI Config[2]					Конфигурация каналов дискретного ввода namur
DI Config[3]					Конфигурация каналов дискретного ввода namur
DI Config[4]					Конфигурация каналов дискретного ввода namur
DI Config[5]					Конфигурация каналов дискретного ввода namur
DI Config[6]					Конфигурация каналов дискретного ввода namur
DI Config[7]					Конфигурация каналов дискретного ввода namur
DI Config[8]					Конфигурация каналов дискретного ввода namur

Рис. 5.21 - Конфигурация каналов дискретного ввода модуля МК-523-032А

Параметры каналов устройства МК-523-032А (структура DINamurChannelParameters):

— Enabled – разрешение работы канала. FALSE – запрещён, TRUE – разрешён. Для запрещённого канала элементы массивов DI и DI trigger всегда равны FALSE. По умолчанию все каналы разрешены;

— Invert – режим инверсии поля канала DI. FALSE – инверсия выключена, TRUE – инверсия включена. По умолчанию инверсия всех каналов выключена;

— BounceTime – минимально допустимая длина входного сигнала на канале, от 0 до 100000 мкс. Входной сигнал длительностью меньше значения BounceTime игнорируется. Данный параметр используется для защиты входов от дребезга. По умолчанию значение BounceTime всех каналов равно 0.

Если модуль МК-523-032А работает с резервированием параметр Redundancy должен быть установлен в TRUE, как показано на Рис. 5.22..

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
POWERLINK Id	USINT	240	240		Идентификатор POWERLINK
Rack Offset	USINT	1	1		Адрес корзины
Slot	USINT	1	1		Позиция в корзине
CAN Id	USINT	1	1		CANOpen-адрес
DI Config	ARRAY [1..32] OF DINamurChannelParameters				Конфигурация каналов дискретного ввода namur
Redundancy	BOOL	TRUE	FALSE		FALSE - отключить режим резервирования, TRUE - включить режим резервирования
Powerlink multiplicity					Кратность PDO, применяется в Powerlink
TPDO_Mul	USINT	1	1		Коэффициент дробления при передаче через POWERLINK (к модулю)
RPDO_Mul	USINT	1	1		Коэффициент дробления при передаче через POWERLINK (от модуля)

Рис. 5.22 - Параметр резервирования дискретного ввода модуля МК-523-032А

5.11. Модули дискретного вывода МК-532-032А

Кроме диагностического канала, модули МК-532-032А имеют в своём составе:

— выходной канал DQ данных значений каналов дискретного вывода модуля, представляющий массив из 32 элементов типа BOOL;

— входной канал NamurDiagnostics диагностических Namur данных, представляющий собой массив из 32 элементов типа DO32NamurDiag.

Входные каналы устройства в структуре DO32NamurDiag возвращают текущее состояние дискретного входа в поле State и статусы в поле Status.

Имя	Тип	Наследо...	Адрес	Начальн.	Комментарий
IODiag	IOModuleDiag				Общая диагностическая информация модуля на шине CAN
DQ	D032ValueType				Значения каналов дискретного вывода
NamurDiagnostics	D032NamurDiagType				Namur диагностика модуля

Рис. 5.23 - Каналы дискретного ввода модуля МК-532-032А

Если модуль МК-532-032А работает с резервированием параметр Redundancy должен быть установлен в TRUE, как показано на Рис. 5.24.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					
POWERLINK Id	USINT	240	240		POWERLINK Id, адрес корзины, положение в корзине
Rack Offset	USINT	1	1		Идентификатор POWERLINK
Slot	USINT	1	1		Адрес корзины
CAN Id	USINT	1	1		Позиция в корзине
DO Config	ARRAY [1..32] OF DONamurChannelParameters				CANOpen-адрес
Extended settings					Конфигурация каналов дискретного вывода
Delay ErrorMode	USINT	0	0	C	Расширенные параметры
Redundancy	BOOL	TRUE	FALSE		Задержка перед переходом выходов в состояние Egor
Powerlink multiplicity					Режим работы в резерве
TPDO_Mul	USINT	1	1		Кратность PDO, применяется в Powerlink
RPDO_Mul	USINT	1	1		Кoeffициент дробления при передаче через POWERLINK (к модулю)
					Кoeffициент дробления при передаче через POWERLINK (от модуля)

Рис. 5.24 - Параметр резервирования дискретного ввода модуля МК-532-032A

5.12. Коммуникационные модули МК-541-002

Кроме диагностического канала, модуль МК-541-002 имеет в своём составе 2 группы каналов, каждая из которых включает:

- входной канал Statuses значений статусов запросов в режиме Master, представляющий массив из 32 элементов типа USINT. Расшифровка значений статусов представлена в Табл. 5.10;
- выходной канал Requests значений параметров запросов в режиме Master, представляющий массив из 32 элементов структурного типа MK541Request (см. Рис. 5.25);
- выходной канал Commands команд для управления запросами в режиме Master, представляющий массив из 32 элементов структурного типа MK541RequestCommand (см. Рис. 5.26). Каждая группа соответствует одному порту модуля МК-541-002.

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
COM1		Diagnostics	%ID235			CAN диагностика
Statuses		%IB1044	ARRAY [1..64] OF USINT			Modbus статусы порта
Requests		%QW1016				Modbus запросы порта
Val		%QW1016	ARRAY [1..64] OF MK541Request			Массив запросов
Val[1]		%QW1016				Массив запросов
SlaveId		%QB2032	USINT(1..247)			Адрес ведомого устройства
ModbusFunction		%QB2033	Enumeration of USINT			Код Modbus-функции
Address		%QW1017	UINT			Начальный адрес запроса
Count		%QW1018	UINT(1..1016)			Количество запрашиваемых адресов
ResultOffset		%QW1019	UINT(0..959)			Смещение результата в массиве данных
SingleRequest		%QB2040	USINT			1 - Однократный запрос по требованию, 0 - непрерывный запрос
OnModifyRequest		%QB2041	USINT			Выполнять запрос при изменении данных (только для запросов на запись)
RepeatOverScan		%QB2042	USINT			1 - Повторять неудачный запрос через скан, 0 - повторять сразу
SkipRepeatsWhenBad		%QW1022	UINT			Число запросов, которое пропускается перед следующим опросом неотвечившего устройства
Val[2]		%QW1023				Массив запросов
Val[3]		%QW1030				Массив запросов

Рис. 5.25 - Структура параметров запроса в режиме Master

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
COM1		Diagnostics	%ID235			CAN диагностика
Statuses		%IB1044	ARRAY [1..64] OF USINT			Modbus статусы порта
Requests		%QW1016				Modbus запросы порта
Commands		%QB2928				Modbus команды порта
Val		%QB2928	ARRAY [1..64] OF MK541RequestCommand			Массив команд запросов
Val[1]		%QB2928				Массив команд запросов
Enable		%QB2928	USINT			Разрешение запроса
DoSingleRequest		%QB2929	USINT			Выполнить запрос однократно
Val[2]		%QB2930				Массив команд запросов
Val[3]		%QB2932				Массив команд запросов

Рис. 5.26 - Структура команд управления запросом в режиме Master

Таблица 5.10 – Значения статусов Modbus в режиме “ведущий”

Имя	Код ошибки	Описание
NO_ERROR	0	Нет ошибки
ILLEGAL_FUNCTION	1	Неверная функция
ILLEGAL_DATA_ADDRESS	2	Неверный адрес данных
ILLEGAL_DATA_VALUE	3	Неверное значение данных
SLAVE_DEVICE_FAILURE	4	Общий сбой устройства сервера
ACKNOWLEDGE	5	Ведомое устройство приняло запрос и обрабатывает его, но это требует много времени. Этот ответ предохраняет ведущее устройство от генерации ошибки тайм-аута.
SLAVE_DEVICE_BUSY	6	Устройство сервера занято

Имя	Код ошибки	Описание
NEGATIVE_ACKNOWLEDGE	7	Ведомое устройство не может выполнить программную функцию, заданную в запросе
MEMORY_PARITY_ERROR	8	Данный код не поддерживается
EXCEPTION_NOT_DEFINED	9	Неопределенная ошибка
GATEWAY_PATH_UNAVAILABLE	10	Шлюз: путь не найден
GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND	11	Шлюз: целевое устройство не отвечает
CRC_ERROR	12	Контрольная сумма не совпала
INVALID_DATA	13	Данные ответа не соответствует запросу (число байт в ответе не соответствует ожидаемому)
UNKNOWN_EXCEPTION_CODE	14	Неизвестный код ошибки
INVALID_EXCEPTION_CODE	15	Код ошибки не соответствует запросу
TOO_MANY_DATA	16	При правильной CRC данных в ответе больше, чем 125 для регистров и 2000 для битов
RESPONSE_NOT_FROM_REQUESTED_SLAVE	17	Ответ от сервера (ведомого) с другим адресом
TIME_OUT	20	Таймаут запроса
INVALID_REQUEST_FORMAT	21	Неправильно заполнены поля структуры запроса
INIT_ERROR	22	Не удалось инициализировать системную библиотеку Modbus
CONNECT_ERROR	23	Не удалось соединиться (по ModbusTCP) с сервером
DATA_QUANTITY_ERROR	24	Передано/принято данных меньше, чем следовало
DATA_OUT_OF_RANGE_ERROR	25	Адрес/длина принимаемых/передаваемых данных выходят за пределы связанных с Modbus массивами переменных
UPDATING	30	Запрос обновляется (передается по CAN из модуля CPU в модуль МК-541-002). Только для устройства МК-541-002
UPDATE_ERROR	31	Ошибка обновления запроса модуля МК-541-002 (при этой ошибка следует повторить попытку обновления).
REQUEST_COMPLETED	100	Нет ошибки (для устройства МК-541-002 коды 0 и 100 чередуются)
PROCESSING	254	Запрос обрабатывается
NOT_PROCESSING_YET	255	Запрос ещё ни разу не обрабатывался

Кроме того, модуль МК- 541- 002 имеет в своём составе выходной канал Data, представляющий массив из 960 элементов типа WORD.

 **Примечание** Массив канала Data используется обоими портами независимо друг от друга. Задача разделения данных между командами обоих портов возлагается на разработчика программы пользователя.

В режиме «ведомый» доступны только Holding Registers, с адресами 0..959. Остальные типы данных Modbus не поддерживаются. Причем эти регистры имеют общую память для обоих портов модуля. Эти регистры непосредственно связаны с массивом каналом Data. Т.е. если оба порта настроены в режиме «ведомый» и по одному из портов записываются какие-то данные, то эти данные появятся и на втором порту по соответствующим адресам.

 **ВНИМАНИЕ** Кроме того, массив канала Data используется совместно портами как в режиме «ведущий» так и режиме «ведомый». Т.е. если один порт настроен как «ведущий», а второй как «ведомый». Ведомому доступна вся память массива канала Data и внешние устройства могут туда записывать любые данные. В это же время Мастер использует эту же память для передачи результатов своих запросов. В результате потенциально может получиться, что записываемые данные по линии «ведомого» перезапишут результаты работы мастера. Это необходимо учитывать при проектировании - ограничивать запись в ведомое устройство по адресам, память которых используется в режиме «ведущего».

Описание элементов структуры МК541Request:

- SlaveID - Идентификатор устройства, к которому адресуется запрос;
- ModbusFunction - Код Modbus функции запроса;

- Address - Начальный адрес данных запроса в зависимости от кода функции: регистра, coil или дискретного входа;
- Count - Количество адресов данных в зависимости от кода функции: регистров, coil'ов или дискретных входов;
- ResultOffset - Смещение в массиве канала Data, по которому из него будут взяты данные для записи (для функций записи), или в него будут переданы данные полученные в результате запроса (для функций чтения);
- SingleRequest - Флаг однократного запроса. Если значение не равно 0, то запрос будет выполняться однократно по команде DoSingleRequest канала Commands с соответствующим индексом;
- OnModifyRequest - Флаг выполнения запроса по изменению данных функций записи. Если флаг не равен 0, то перед каждым запросом проверяется изменились ли данные в массиве канала Data по соответствующему смещению ResultOffset. Если данные изменились запрос выполняется, если нет - пропускается;
- RepeatOverScan - Флаг повтора при неудачном запросе. Если значение равно 1, запрос повторяется через скан, иначе повторяется сразу;
- SkipRepeatsWhenBad - Число запросов, которое пропускается перед следующим запросом не ответившего устройства.

! **ВНИМАНИЕ** Если для функций 5 и 6 параметр Count не будет равен 1, то вместо этих функций будут выполнены соответственно 15 и 16 функции.

Описание элементов структуры MK541RequestCommand:

- DoSingleRequest - Выполнить однократный запрос с соответствующим индексом;
- Enable - Разрешение запроса с соответствующим индексом.

Также имеется конфигурационный канал для настройки параметров портов модуля МК-541-002 (Рис. 5.27).

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
COM1					
Config					Конфигурация Modbus-порта
Mode	Enumeration of USINT	Master	Master		Режим работы порта
Baudrate	Enumeration of UDINT	9600	9600		Скорость работы порта
Parity	Enumeration of USINT	None	None		Контроль четности
Databits	USINT	8	8		Число бит в символе
Stopbits	Enumeration of USINT	1	1		Число стоп-бит
SlaveId	USINT(1..247)	1	1		Slave ID для Slave режима
RepeatsOnError	USINT	0	0		Число повторов запрос при неудачной операции
DelayBefore	USINT	0	0 мс		Пауза перед выполнением запроса
Timeout	UINT	100	100 мс		Таймаут ожидания ответа
CycleTime	UINT	0	0 мс		Время полного цикла выполнения очереди команд
COM2					

Рис. 5.27 - Конфигурация модуля МК-541-002

Параметры каналов устройств Config:

- Mode – Режим работы порта: Master – ModbusRTU в режиме «ведущий», Slave - ModbusRTU в режиме «ведомый». По умолчанию включен режим Master;
- Baudrate – Скорость работы порта, по умолчанию 9600;
- Bits – Число бит в символе. На данный момент поддерживается только режим 8 бит;
- Parity – Контроль чётности: None – нет, Even – even, Odd – odd. По умолчанию None – без контроля чётности;
- Stopbits – Число стоп-бит: 1 – 1 стоп-бит, 2 – 2 стоп-бита. По умолчанию 1;
- SlaveID – Идентификатор порта на шине Modbus для режима работы «ведомый», по умолчанию установлен 1;
- RepeatsOnError – Число повторов неудачно выполненной команды для режима работы Modbus «ведущий», 0 – нет повторов. По умолчанию 0, без повторов;
- DelayBefore – Задержка перед выполнением следующего запроса в мс для режима «ведущий». По умолчанию 0 - без задержки;

— Timeout – Таймаут ожидания ответа на запрос (в мс) для режима работы Modbus «ведущий», 0 – автоматическое определение времени ожидания ответа. По умолчанию 100 мс;

— CycleTime – Заданное время полного выполнения очереди команд (в мс) для режима работы Modbus «ведущий», 0 – нет ограничений на полное время выполнения. По умолчанию 0 – без ограничений.

Если какой-либо из параметров лежит вне допустимых пределов (например, Stopbits=5), при работе в режиме «ведущий» все Modbus-запросы будут возвращать код 21 (Табл. 5.10), при работе в режиме «ведомый» этот параметр будет проигнорирован, и будет использоваться последнее корректное значение этого параметра в модуле МК-541-002.

5.12.1. Инициализация и передача команд в модуль МК-541-002

Для передачи команд в модуль МК-541-002 следует заполнить соответствующие элементы массивов переменных, привязанных к каналам [Requests](#) и [Commands](#), см. Листинг 5.1.

```
// Пример заполнения параметров запросов
FOR i := 1 TO 64 DO
  requests_port1.Val[i].SlaveID := 1;
  requests_port1.Val[i].ModbusFunction := 16;
  requests_port1.Val[i].Address := 10 + ANY_TO_UINT(i*13);
  requests_port1.Val[i].Count := 13;
  requests_port1.Val[i].ResultOffset := ANY_TO_UINT(i*13);
  requests_port1.Val[i].SingleRequest := 0;
  requests_port1.Val[i].OnModifyRequest := 0;
  requests_port1.Val[i].RepeatOverScan := 0;
  requests_port1.Val[i].SkipRepeatsWhenBad := 5;

  commands_port1.Val[i].DoSingleRequest := 0;
  commands_port1.Val[i].Enable := 1;

  requests_port2.Val[i].SlaveID := 1;
  requests_port2.Val[i].ModbusFunction := 16;
  requests_port2.Val[i].Address := 1000 + ANY_TO_UINT(i*13);
  requests_port2.Val[i].Count := 13;
  requests_port2.Val[i].ResultOffset := ANY_TO_UINT(i*13);
  requests_port2.Val[i].SingleRequest := 0;
  requests_port2.Val[i].OnModifyRequest := 0;
  requests_port2.Val[i].RepeatOverScan := 0;
  requests_port2.Val[i].SkipRepeatsWhenBad := 5;

  commands_port2.Val[i].DoSingleRequest := 0;
  commands_port2.Val[i].Enable := 1;
END_FOR
```

Листинг 5.1 – Пример заполнения параметров запросов

Запросы и команды можно обновлять в ходе выполнения программы пользователя. При обновлении команд следует учитывать следующие особенности:

- 1) Команды обновляются только блоками по 16 команд (1..16, 17..32 и так далее);
- 2) Блоки команд обновляются по одному. Блоки команд для разных портов обновляются последовательно: сначала для первого порта, потом для второго;
- 3) При обнаружении обновлений в первом встреченном блоке, обновления в последующих блоках игнорируются до завершения обновления первого встреченного блока;
- 4) Обновление блоков команд не происходит мгновенно, и может завершиться неудачей. Для контроля хода обновления данных следует анализировать значение массива канала [Statuses](#) (Табл. 5.10) с любым индексом, лежащем внутри изменяемого блока. Интерес представляют возвращаемые коды 30 и 31;
- 5) Если в программе пользователя точно изменялись команды, но в следующем цикле возвращаемые для этих команд коды не равны 30 или 31, это означает, что обновление команд завершилось успехом, и коды относятся уже к новым командам.

5.12.2. Особенности работы Modbus-функций 1, 2, 3 и 4 в модуле МК-541-002

При работе с данными устройства МК-541-002 следует учитывать, что если регистр данных (канал Data) задействован хотя бы в одной Modbus-функции чтения (1, 2, 3 или 4), то запись со стороны программы пользователя в этот регистр будет игнорироваться, и данные регистра будут обновляться только значениями, получаемыми со стороны модуля МК-541-002.

Если регистр не задействован ни в одной из операций чтения, запись в него (и в модуль МК-541-002) со стороны программы пользователя будет выполняться в конце каждого цикла работы программы пользователя.

5.12.3. Особенности работы Modbus-функций 1, 2, 5 и 15 в модуле МК-541-002

При работе с Modbus-данными типа Discrete Input и Coil следует учитывать следующие особенности:

1) Смещение (поле Offset структуры МК541Request) для модуля МК-541-002 указывается в регистрах, даже для команд 1, 2, 5 и 15. При этом битное смещение внутри выбранного регистра считается равным 0 и идёт с младших бит.

Например, при чтении 10 переменных типа Coil с адреса Modbus-устройства 8 в регистр МК-541-002 со смещением 3, все 10 прочитанных переменных будут скопированы в младшие 10 бит регистра 3 МК-541-002;

2) При копировании прочитанных данных в регистры МК-541-002 биты регистров МК-541-002, не участвующие в операции, не изменяются.

Например, при чтении 10 переменных типа Coil с адреса Modbus-устройства 8 в регистр МК-541-002 со смещением 3, старшие 6 бит регистра 3 МК-541-002 не изменятся;

3) При записи данных в Modbus-устройство в команду копируются только реальное число бит, указанных в команде;

4) Если число бит в команде превышает 16, используются биты последующих регистров МК-541-002, также начиная с младших бит.

5.13. Коммуникационные модули МК-545-010

Модуль МК-545-010 предназначен для работы с протоколом Powerlink в режиме ведомый. Он должен добавляться в устройство Rack с Rack Offset=1, при этом родительское устройство CAN должно быть дочерним устройства CN (см. в разделе Дистанционные стойки подключенные по протоколу POWERLINK). Модуль МК-545-010 заменяет модуль центрального процессора в сети CAN, расположенной за сетью POWERLINK.

Устройство МК-545-010 имеет в своём составе канал диагностической информации работы портов модуля Powerlink diagnostics, который полностью совпадает с одноимённым каналом диагностической информации модуля МК-546-010. Единственная разница в том, что диагностика производится для роли CN, в отличие от роли MN для модуля МК-546-010.

5.14. Модули аналогового вывода МК-574-008А с протоколом HART

Модуль МК-574-008А по каналам аналогового вывода полностью аналогичен модулю МК-514-008А. Кроме того, для поддержки чтения переменных по протоколу HART модуль МК-574-008А имеет в своём составе входной канал Hart, представляющий собой массив из 8 элементов типа HARTChannelValueType (Рис. 5.28-5.30). По каждому каналу ввода-вывода допускается чтение по HART до четырёх заранее настроенных переменных.

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%ID293			CAN диагностика
		AO Statuses	%IB1276	ARRAY [1..8] OF USINT		Статусы каналов аналогового вывода
		Hart	%ID321			Значения HART каналов аналогового вывода
		Values	%ID321	ARRAY [1..8] OF HartChannelValueType		
		Values[1]	%ID321			
		HartSelected	%IB1284	USINT		Количество используемых HART переменных
		Id	%IB1285	USINT		ID датчика
		ResponseError	%IB1286	USINT		Нет ответа от устройства
		ConfigError	%IB1287	USINT		Неверная конфигурация канала
		CommunicationError	%IB1288			Флаги коммуникационных ошибок
		CommandStatus	%IB1293			Флаги обработки команды
		HartValues	%ID324	ARRAY [1..4] OF REAL		Значения Hart переменных
		HartValues[1]	%ID324	REAL		Значения Hart переменных
		HartValues[2]	%ID325	REAL		Значения Hart переменных
		HartValues[3]	%ID326	REAL		Значения Hart переменных
		HartValues[4]	%ID327	REAL		Значения Hart переменных
		Values[2]	%ID328			

Рис. 5.28 - Описание структуры HartChannelValueType

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Diagnostics	%ID293			CAN диагностика
		AO Statuses	%IB1276	ARRAY [1..8] OF USINT		Статусы каналов аналогового вывода
		Hart	%ID321			Значения HART каналов аналогового вывода
		Values	%ID321	ARRAY [1..8] OF HartChannelValueType		
		Values[1]	%ID321			
		HartSelected	%IB1284	USINT		Количество используемых HART переменных
		Id	%IB1285	USINT		ID датчика
		ResponseError	%IB1286	USINT		Нет ответа от устройства
		ConfigError	%IB1287	USINT		Неверная конфигурация канала
		CommunicationError	%IB1288			Флаги коммуникационных ошибок
		Parity	%IB1288	USINT		Ошибка четности
		SpeedOverflow	%IB1289	USINT		Перегрузка по скорости
		Sync	%IB1290	USINT		Ошибка синхронизации или формата посылки
		Crc	%IB1291	USINT		Ошибка контрольной суммы
		BufferOverflow	%IB1292	USINT		Переполнение буфера приёмника
		CommandStatus	%IB1293			Флаги обработки команды
		HartValues	%ID324	ARRAY [1..4] OF REAL		Значения Hart переменных
		Values[2]	%ID328			

Рис. 5.29 - Описание поля CommunicationError структуры HartChannelValueType

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		AO Statuses	%IB1276	ARRAY [1..8] OF USINT		Статусы каналов аналогового вывода
		Hart	%ID321			Значения HART каналов аналогового вывода
		Values	%ID321	ARRAY [1..8] OF HartChannelValueType		
		Values[1]	%ID321			
		HartSelected	%IB1284	USINT		Количество используемых HART переменных
		Id	%IB1285	USINT		ID датчика
		ResponseError	%IB1286	USINT		Нет ответа от устройства
		ConfigError	%IB1287	USINT		Неверная конфигурация канала
		CommunicationError	%IB1288			Флаги коммуникационных ошибок
		CommandStatus	%IB1293			Флаги обработки команды
		NoCommand	%IX1293.0	BOOL		Нет команды
		WrongChoose	%IX1293.1	BOOL		Неверный адрес для опроса
		ParamTooBig	%IX1293.2	BOOL		Последний принятый параметр слишком велик
		ParamTooSmall	%IX1293.3	BOOL		Последний принятый параметр слишком мал
		FewData	%IX1293.4	BOOL		Получено недостаточное количество байт данных
		WriteProtect	%IX1293.5	BOOL		Для переменной выставлен режим защиты от записи
		AccessDenied	%IX1293.6	BOOL		Для переменной ограничен доступ
		DataNotUpdates	%IX1293.7	BOOL		Не обновляются данные переменной
		DeviceBusy	%IX1294.0	BOOL		Устройство занято
		CommandNotImplemented	%IX1294.1	BOOL		Команда не реализована
		DeviceMalfunction	%IX1294.2	BOOL		Устройство неисправно
		ConfigChanged	%IX1294.3	BOOL		Изменена конфигурация устройства
		ColdStart	%IX1294.4	BOOL		Выполняется "холодный старт" устройства
		MoreStatusAvailable	%IX1294.5	BOOL		Доступна дополнительная информация о статусе устройства
		AoFixed	%IX1294.6	BOOL		Значение аналогового выхода фиксировано
		AoSaturated	%IX1294.7	BOOL		Аналоговый выход в насыщении
		PrimaryOutOfLimits	%IX1295.0	BOOL		Значение главной переменной вне установленных пределов
		NonPrimaryOutOfLimits	%IX1295.1	BOOL		Значение неглавной переменной вне установленных пределов
		HartValues	%ID324	ARRAY [1..4] OF REAL		Значения Hart переменных
		Values[2]	%ID328			

Рис. 5.30 - Описание поля ComandStatus структуры HartChannelValueType

Канал HART Address Offset используется при подключении нескольких модулей МК-574-008А и МК-576-0xxА на общую шину. Подробнее настройка параметра HART Address Offset описана в Приложение А. Инструкция по интеграции в PDM-систему.

Также имеется конфигурационный канал Hart Config для настройки протокола HART на каналах аналогового вывода модуля. Он представляется собой массив из 8 элементов структурного типа HartChannelConfigType (Рис. 5.31).

Параметр	Тип	Значение	Значение по ум...	Единица	Описание
Common					POWERLINK Id, адрес корзины, положение в корзине
AO Config	ARRAY [1..8] OF AOChannelParameters				Конфигурация каналов аналогового вывода
Hart Config	ARRAY [1..8] OF HartChannelConfigType				Конфигурация HART каналов аналогового вывода
Hart Config[1]					Конфигурация HART каналов аналогового вывода
HartVarUsed	USINT(0..4)	0	0		Количество используемых HART переменных
TimeoutRequest	UINT(1000..10000)	0	0	мс	Таймаут ответа на запрос
DelayBetweenRequest	UINT(0..500)	0	0	мс	Пауза между запросами
SensorId	USINT				ID датчика на канале (0..63, 255). При ID=255 выполняется автоматический поиск датчика
Hart Config[2]					Конфигурация HART каналов аналогового вывода

Рис. 5.31 - Описание конфигурации HART

Структура HartChannelConfigType содержит следующие члены:

- HartVarUsed – число переменных HART, которые будут считываться по данному каналу, от 0 до 4. Указывается число подряд идущих переменных (3 – значит переменные 1, 2 и 3);
- SensorID – идентификатор датчика на канале, допустимые значения 0..63, и 255. При выборе идентификатора 255 выполняется автоматический поиск датчика на данном канале;
- TimeoutRequest – таймаут ответа на запросы по протоколу HART в мс, допустимые значения 1000..10000;
- DelayBetweenRequest – задержка между запросами разных мастеров в мс, допустимые значения 0..500.

Значения канала AO Statuses соответствуют состоянию электрических цепей соответствующих каналов модуля аналогового вывода. Код состояния канала имеет тип USINT и состоит из двух частей. Младшие 4 бита кода состояния (см. Табл. 5.11) описывают общее состояние канала, старшие 4 бита кода состояния (см. Табл. 5.12) содержат в себе код ошибки.

Таблица 5.11 – Расшифровка статусов каналов аналогового выхода

Значения битов статуса	Состояние канала аналогового выхода
0	1 – обрыв цепи канала
1	1 – отсутствует внешнее питание модуля
2	1 – канал в резерве (устанавливается одновременно на все каналы, кроме выключенных)
3	1 – ошибка канала (код ошибки находится в старших 4 битах статуса); приводит к дисквалификации модуля
4..7	Код ошибки канала устройства

Таблица 5.12 – Расшифровка кодов ошибки статусов каналов аналогового выхода

Значения битов статуса	Состояние канала аналогового выхода
0	Нет ошибок
1	Ошибка конфигурации (параметры канала Mode и Enabled резервированных каналов не совпадают)
2	Ошибка задания тока (разное значение кода ЦАП резервированных каналов)
3	Ошибка ЦАП канала
4	Ошибка связи между резервной парой модулей (отказ “перемычки”)
5	Обрыв цепи на одном из резервированных каналов
6	Отказ SIL ячейки
7	Перезагрузка по watchdog
8	Другой внутренний отказ модуля

5.15. Модули аналогового ввода МК-576-008А с протоколом HART

Модуль МК-576-008А по каналам аналогового ввода полностью аналогичен модулю МК-516-008А, по каналам для работы с протоколом HART полностью аналогичен модулю МК-574-008А.

5.16. Модули аналогового ввода МК-576-016А с протоколом HART

Модуль МК- 576- 016А по каналам аналогового ввода полностью аналогичен модулю МК- 513- 016А. По каналам для работы с протоколом HART практически аналогичен модулю МК-574-008А, за исключением того что Каналы Hart и Hart Config представляют собой массивы из 16 элементов соответствующих структурных типов вместо 8 элементов для модуля МК-574-008А.

5.17. Дистанционные стойки, подключенные по протоколу POWERLINK

Дистанционные стойки подключаются к основной по протоколу POWERLINK. Для этого Модуль центрального процессора должен иметь подключенный к шине PCI модуль МК-546-010. Модуль центрального процессора выступает как ведущее устройство и имеет роль MN. Каждая дистанционная корзина должна иметь один модуль МК- 545- 010, заменяющий собой модуль центрального процессора на шине CAN.

В дерево устройств должно быть добавлено устройство Powerlink, в которое, в качестве дочернего устройства, может быть добавлено необходимое количество устройств CN. В устройство CN необходимо добавить устройство CAN, которое уже было описано ранее. Добавление устройств в CAN производится также как и в CAN для основной шины CAN. Пример использования Powerlink для двух небольших дистанционных стоек с двумя корзинами представлен на Рис. 5.32.

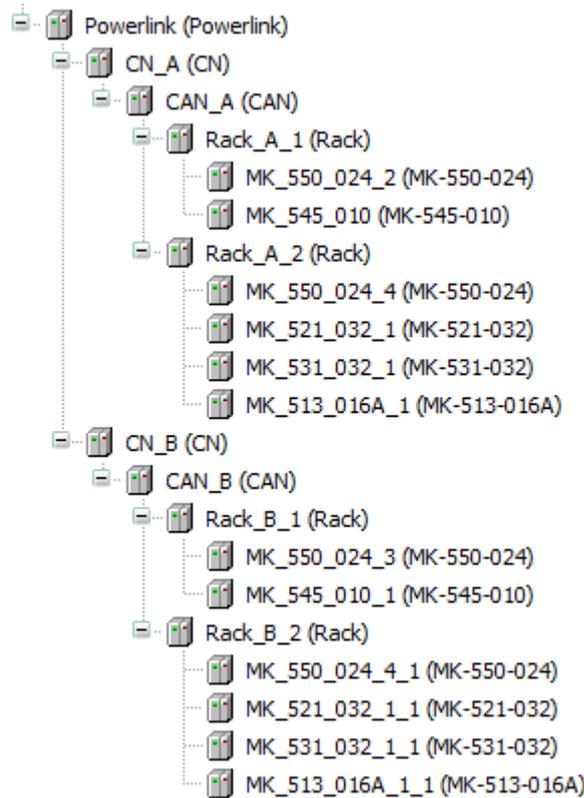


Рис. 5.32 - Пример дерева устройств при использовании Powerlink

Устройство Powerlink имеет конфигурационные параметры представленные на Рис. 5.33.

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
DLL					Уровень канала передачи данных
MN					DLL Параметры MN
1C00h: DLL_MinCRCError_REC					Данные объекты используются для мониторинга ошибок CRC.
03h: Threshold_U32	UINT(8...1000)	15		15	Значение счетчика ThresholdCnt_U32, по достижении которого возникает событие "CRC Error"
1C02h: DLL_MinCycleTimeExceed_REC					Данные объекты используются для мониторинга ошибок "Cycle time exceeded".
03h: Threshold_U32	UINT(8...1000)	15		15	Значение счетчика ThresholdCnt_U32, по достижении которого возникает событие "Cycle time exceeded"
CN					DLL Общие параметры CN
1C08h: DLL_OnLossOfSoc_REC					Данные объекты используются для мониторинга ошибок "Loss of Soc" обнаруженных на стороне CN.
03h: Threshold_U32	UINT(8...1000)	15		15	Значение счетчика ThresholdCnt_U32, по достижении которого возникает событие "Loss of Soc"
1C0Dh: DLL_CNLossPReq_REC					Данные объекты используются для мониторинга ошибок "Loss of PReq".
03h: Threshold_U32	UINT(8...1000)	15		15	Значение счетчика ThresholdCnt_U32, по достижении которого возникает событие "Loss of PReq"
1C14h: DLL_CNLossOfSocTolerance_U32	UDINT(10000...4294967295)	100000		100000 нс	Интервал допущена используемый CN для распознавания ошибки "Loss of Soc".
NFT					Объекты управления сетью
General					Общие объекты
1009h: NMT_CycleLen_U32	UDINT(10000...4294967295)	50000		50000 нкс	Время цикла Powerlink
MN					NFT Объекты MN
1F89h: NMT_BootTime_REC					Интервалы времени используемые MN при старте работы сети.
01h: MNWaitForAct_U32	UDINT(250...10000000)	1000000		1000000 нкс	Интервал времени, в течении которого MN должно оставаться в состоянии NMT_MS_NOT_ACTIVE и слушать кадры POWERLINK в сети перед изменением состояния на N
02h: MNTimeoutPreOp_1_U32	UDINT(0...5000000)	500000		500000 нкс	Интервал времени, в течении которого MN должно оставаться в состоянии NMT_MS_PRE_OPERATIONAL_1, чтобы все обязательные CN идентифицировались с помощью
03h: MNWaitPreOp_1_U32	UDINT(0...5000000)	500000		500000 нкс	Интервал времени, в течении которого MN должно оставаться в состоянии NMT_MS_PRE_OPERATIONAL_1
04h: MNTimeoutPreOp_2_U32	UDINT(0...5000000)	500000		500000 нкс	Интервал времени, в течении которого MN должно оставаться в состоянии NMT_MS_PRE_OPERATIONAL_2, чтобы все обязательные CN завершили инициализацию себя
05h: MNTimeoutReadyToOp_U32	UDINT(0...5000000)	500000		500000 нкс	Интервал времени, в течении которого MN должно оставаться в состоянии NMT_MS_READY_TO_OPERATE, чтобы все обязательные CN перешли в состояние NMT_CS_ON

Рис. 5.33 - Конфигурационные параметры устройства Powerlink

Для расчёта времени цикла Powerlink в микросекундах рекомендуется использовать следующую формулу: $T = ((N + 2) * 0.7 + 5) * 1000$, где N - количество CN.

В ходе работы сети Powerlink между MN и каждой CN происходит обмен одной PDO за скан: на 1 TDPO от MN следует ответ 1 RPDO от CN.

Максимальный размер TPDO/RPDO составляет 1490 байт. В эти данные должна поместиться оперативная информация о модулях и данные.

Оперативная информация занимает n+1 байт, где n - число модулей ввода-вывода (включая МК-545-010) на CAN-шине дистанционной стойки. То есть для стойки с 20 модулями остаётся свободным под данные 1469 байт в TPDO и 1469 байт в RPDO.

В Табл. 5.13 приводится информация о числе байт в TPDO и RPDO всех типов модулей ввода-вывода.

Таблица 5.13 – Число байт в TPDO и RPDO для всех типов модулей ввода-вывода

Тип модуля	Байт в TPDO	Байт в RPDO
МК-550-024	0	8
МК-545-010	0	8
МК-521-032(A)	8	32
МК-531-032(A)	16	0
МК-513-016(A)	0	32
МК-516-008(A)	0	16
МК-514-008(A)	16	8
МК-576-008A	0	176
МК-576-016A	0	352
МК-574-008A	16	168
МК-541-002	2048	2048

Из Табл. 5.13 понятно, что при большом количестве модулей общее количество данным может не укладываться в кадр PDO, причем данные даже одного модуля МК-541-002 не укладываются в кадр PDO. В связи с этим вводится понятие кратности данных для каждого модуля, означающее число частей на которое делятся его данные при передаче через PDO протокола POWERLINK в определённом направлении.

Например, для модуля МК-541-002 кратность 4 для TPDO и 8 для RPDO означает, что данные TPDO будут разделены на 4 части и будут передаваться в модуль в течение 4 циклов обмена Powerlink, занимая при этом 512 байт в одной PDO, а данные RPDO будут приниматься из модуля в течение 8 циклов обмена Powerlink, занимая при этом 256 байт.

Кратность для всех модулей рассчитывается автоматически по обоим направлениям независимо, расчет производится при инициализации программы пользователя. Рассчитанные значения записываются в параметр Powerlink multiplicity (Рис. 5.3).

Алгоритм расчета следующий:

- 1) Изначально все кратности равны 1;
- 2) Если общее число байт с учетом кратности не помещается в 1469 переходим к шагу 2, иначе шаг 4;
- 3) Находим модуль с максимальным числом байт поделенным на текущую кратность для модуля и увеличиваем кратность для него вдвое. Возвращаемся к шагу 2;
- 4) Кратности найдены. Выходим.

6. Реализация резервирования в МКLogic-500

6.1. Термины и определения

6.1.1. Роли Primary и Secondary

Роли Primary и Secondary определяются позиционным положением модулей CPU в корзине. Роль Primary имеет модуль центрального процессора установленный в 1-й корзине (Rack Offset = 1) в 3-м слоте (Slot = 3), роль Secondary - модуль центрального процессора установленный в любой другой позиции.

К ролям Primary и Secondary привязываются IP-адреса интерфейсов.

6.1.2. Роли Active и Passive

Роли Active и Passive определяют кто сейчас главный в резервной паре, т.е. активен, а кто находится в горячем резерве, т.е. пассивен. Активным может быть модуль как с ролью Primary, так и с ролью Secondary. Роль определяется исключительно текущей ситуацией и не связана с позицией контроллера. Единственная ситуация когда роли Primary и Secondary влияют на роли Active/Passive является одновременный старт двух контроллеров. В этом случае роль Active получает модуль с позицией Primary. Но в дальнейшей работе позиция не влияет на роли Active/Passive.

Смена ролей Active/Passive происходит по следующим причинам:

— Непосредственно исполнение функции горячего резерва. Тот модуль, который был Active, перестал отвечать, например, отказал или был удален из корзины физически. В этом случае пассивный становится активным и продолжает работу;

— Смена роли по изменению статуса порта с ролью UpLink. Если активный обнаружил что порт с ролью UpLink потерял физическое соединение, но при этом у второго контроллера соответствующий порт имеет физическое соединение, резервная пара контроллеров меняется ролями;

— Смена роли по статусу процедуры пингования. В контроллерах предусмотрена процедура пингования некоторых заданных внешних адресов. В данный момент возможно пинговать 2 адреса. Если активный обнаружил, что статусы его пингов ошибочны, т.е. он не “видит” заданные узлы, но при этом второй контроллер их “видит”, т.е. хотя бы один статус успешный, резервная пара контроллеров меняется ролями.

6.1.3. Состояние Standalone

Состояние Standalone возникает когда модуль центрального процессора не может установить связь со вторым модулем центрального процессора.

6.1.4. Области синхронизируемых данных

Существует 3 области синхронизируемых данных:

— INITIAL - данные синхронизируемые один раз при старте программы пользователя, либо при нарушении консистентности оперативного контекста с пассивным контроллером;

— SYNCHO - данные синхронизируемые каждый цикл программы пользователя;

— OUTPUT - данные не синхронизируются, но контролируется их контрольная сумма.

6.2. Ограничения при использовании резервирования

При использовании резервирования допускается использовать только одно приложение, которое имеет только одну циклическую задачу.

6.3. Использование резервирования

Для поддержки резервирования в проекте предусмотрено устройство Redundancy. Устройство Redundancy имеет несколько конфигурационных параметров (Рис. 6.1).

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Datalink timeout	UINT	40	40	мс	Таймаут связи по переключки
Ping config					Конфигурация процедуры пингования
Enabled	BOOL	FALSE	FALSE		Разрешение процедуры пингования
Timeout	UDINT	100	100	мс	Таймаут процедуры пингования
Period	UDINT	500	500	мс	Периодичность пингования
Ping IP 1	STRING	'127.0.0.1'	'127.0.0.1'		Первый пингуемый адрес
Ping IP 2	STRING	'127.0.0.1'	'127.0.0.1'		Второй пингуемый адрес
Ping status					Статусы процедуры пингования
PR status	SINT	0	0		Статус процедуры пингования Primary CPU
SE status	SINT	0	0		Статус процедуры пингования Secondary CPU
Config status	SINT	0	0		Статус применения конфигурации процедуры пингования

Рис. 6.1 - Параметры конфигурации резервирования

Кроме того, устройство Redundancy имеет несколько каналов диагностики (Рис. 6.2).

Переменная	Соотнесение	Канал	Адрес	Тип	Единица	Описание
		Error code	%ID0	UDINT		Код ошибки резервирования
		Data sync counter	%ID1			Статистика синхронизации данных резервирования
		'Synchro' area size	%ID1	UDINT	Байт	Размер SYNCHRO области
		'Synchro' counter	%ID2	UDINT		Счетчик синхронизаций SYNCHRO области
		Last 'Synchro' time	%ID3	UDINT	мс	Последнее время синхронизаций SYNCHRO области
		'Initial' area size	%ID4	UDINT	Байт	Размер INITIAL области
		'Initial' counter	%ID5	UDINT		Счетчик синхронизаций INITIAL области
		Last 'Initial' time	%ID6	UDINT	мс	Последнее время синхронизаций INITIAL области
		'Output' area size	%ID7	UDINT	Байт	Размер Output области
		Is active standalone	%IX32.0	BOOL		TRUE - если активный в состоянии Standalone, иначе FALSE
		Sync error counters	%ID9			Счётчики ошибок синхронизаций
		Cycle sync timeout counter	%ID9	UDINT		Счётчик таймаутов синхронизации времени циклов
		ARI sync timeout counter	%ID10	UDINT		Счётчик таймаутов в момент After Read Inputs
		AWO sync timeout counter	%ID11	UDINT		Счётчик таймаутов в момент After Write Outputs
		CRC Inconsistent counter	%ID12	UDINT		Счётчик несоответствий CRC оперативных данных

Рис. 6.2 - Каналы диагностики резервирования

Существующие коды ошибок резервирования описаны в Табл. 6.1.

Таблица 6.1 – Коды ошибок резервирования

Код ошибки	Название	Описание
0	None	Нет ошибок
1	SyncTimeout	Таймаут синхронизации циклов
2	ARITimeout	Таймаут синхронизации оперативного контекста
3	AWOTimeout	Таймаут синхронизации CRC оперативного контекста
4	InitialCRCInconsistent	Неконсистентные Initial area CRC
5	OutputCRCInconsistent	Неконсистентные Output area CRC

6.4. Библиотека NftRedundancy

Вспомогательные функции для работы с резервированием представлены в библиотеке NftRedundancy. Библиотека NftRedundancy представляет собой контейнерную библиотеку Procyon IDE, которая состоит из двух библиотек:

- NftRedundancyIfts - интерфейсная библиотека, содержащая все типы данных применяемые в работе резервированием;
- NftRedundancyImpl - библиотека, содержащая основные функции для работы с резервированием.

6.4.1. Функция AreaRegister

Функция AreaRegister библиотеки NftRedundancy регистрирует область памяти для синхронизации.

Параметры и возвращаемые значения функции AreaRegister приведены в Табл. 6.2.

Таблица 6.2 – Параметры и возвращаемое значение функции AreaRegister

Область	Имя	Тип параметра	Описание
Return	AreaRegister	BOOL	TRUE если регистрация успешна

Область	Имя	Тип параметра	Описание
Input	pArea	POINTER TO BYTE	Указатель на регистрируемую область данных
Input	udiSize	UDINT	Размер данных
Input	eType	NftRdcyltfs.AREA_TYPE	Область синхронизации

Пример использования функции AreaRegister представлен в Листинг 6.1:

```

VAR
  data : ARRAY [1..100] OF WORD;
  res : BOOL;
END_VAR
...
// Регистрация массива для синхронизации в области SYNCHRO
res := NftRdcy.NftRdcyImpl.AreaRegister(ADR(data[1]), SIZEOF(data), NftRdcy.NftRdcyItfs.AREA_
TYPE.AREA_INITIAL);
res := NftRdcy.NftRdcyImpl.AreaRegister(ADR(data[1]), SIZEOF(data), NftRdcy.NftRdcyItfs.AREA_
TYPE.AREA_SYNCHRO);
IF NOT res THEN
  ...
END_IF

```

Листинг 6.1 – Пример использования функции AreaRegister

! **ВНИМАНИЕ** Для корректной работы резервирования в функцию AreaRegister запрещено передавать адреса переменных указателей, ссылок, параметры типа VAR_IN_OUT и разного рода контейнеров (функциональных блоков) содержащих их. Вместо указателей и ссылок необходимо адрес передавать непосредственно переменные, от которых взяты ссылки или указатели.

6.4.2. Функция SwapActiveCPU

Функция SwapActiveCPU служит для принудительной смены ролей Active/Passive между модулями в резервированной паре.

Параметры и возвращаемые значения функции SwapActiveCPU приведены в Табл. 6.3.

6.5. Настройка областей синхронизации резервирования

Настройка областей синхронизации резервирования должна быть определена в программе пользователя. Для этого необходимо создать функцию, вызываемую после загрузки и инициализации программы пользователя.

Двойной щелчок на *Конфигурация задач*. В появившемся окне выбрать вкладку *Системные события* и нажать кнопку *Добавить обработчик событий* (Рис. 6.3). В появившемся окне выбрать *Событие* DownloadDone и ввести имя новой функции, например, “AreasFunc”. Остальные поля оставить по умолчанию (Рис. 6.4).

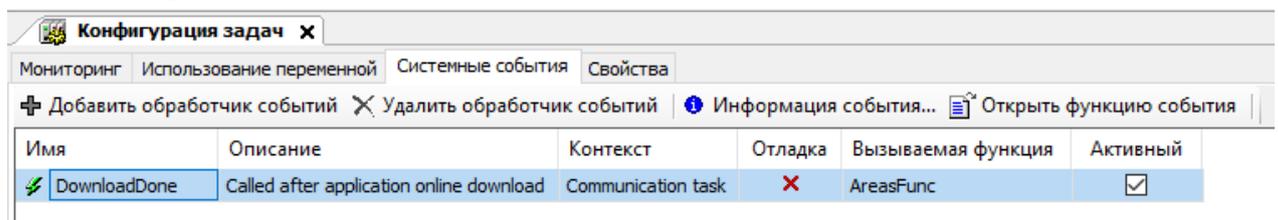


Рис. 6.3 - Окно системных событий Procyon IDE

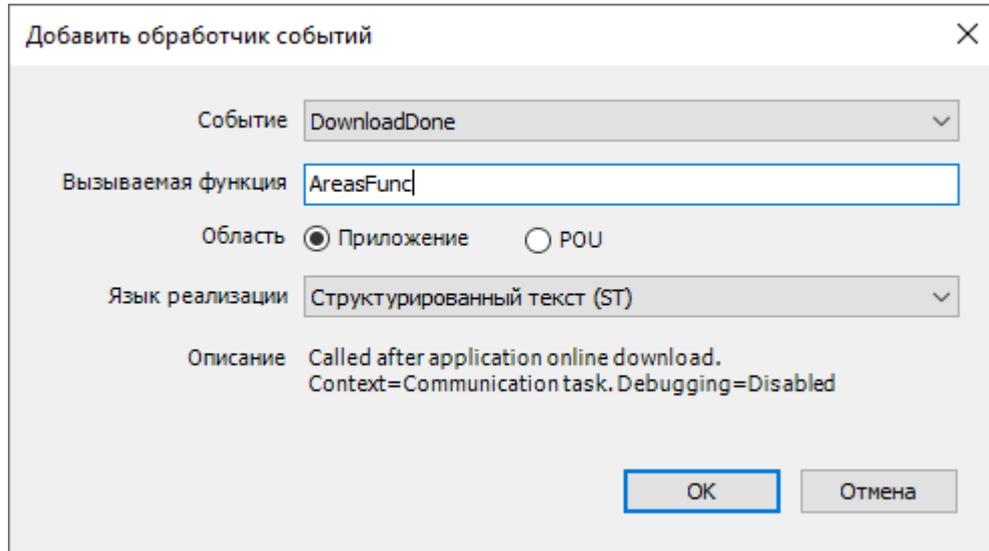


Рис. 6.4 - Добавление обработчика события

После нажатия на кнопку ОК будет создана новая функция которая будет вызывать один раз после загрузки программы пользователя.

В созданной функции необходимо вызвать функцию AreaRegister для всех переменных программы с параметром AREA_INITIAL.

Например, если в программе POU приложения есть переменная Abc, в функции вызываемой по событию DownloadDone, должен быть вызов функции AreaRegister, показанный в листинге 6.2.

```
NftRdcy.NftRdcyImpl.AreaRegister(ADR(POU.Abc), SIZEOF(POU.Abc), NftRdcy.NftRdcyItfs.AREA_
TYPE.AREA_INITIAL);
```

Листинг 6.2 – Пример настройки областей синхронизации для обычных переменных

Для тех переменных, которые необходимо синхронизировать каждый цикл программы пользователя, необходимо вызвать функцию AreaRegister два раза: один с параметров AREA_INITIAL, второй - AREA_SYNCHRO.

К таким переменным относятся:

- все входные переменные модулей ввода вывода;
- все переменные, связанные с работой по протоколам связи (Modbus, IEC 60870-5-104, OPC UA), и которые могут изменяться извне (т.е. переменные доступные для записи по протоколам связи);
- другие переменные на усмотрение пользователя.

Например, если для модуля МК-516-008А канал AI привязан к переменной ai_some1 (см. Рис. 6.5), в функции вызываемой по событию DownloadDone, должны быть вызовы функции AreaRegister, показанные в Листинг 6.3

Переменная	Соотнесение	Канал	Адрес	Тип
ai_some1		AI	%IW234	ARRAY [1..8] OF UINT

Рис. 6.5 - Пример для настройки для резервирования

```
NftRdcy.NftRdcyImpl.AreaRegister(ADR(ai_some1), SIZEOF(ai_some1), NftRdcy.NftRdcyItfs.AREA_
TYPE.AREA_INITIAL);
NftRdcy.NftRdcyImpl.AreaRegister(ADR(ai_some1), SIZEOF(ai_some1), NftRdcy.NftRdcyItfs.AREA_
TYPE.AREA_SYNCHRO);
```

Листинг 6.3 – Пример настройки областей синхронизации для входной переменной

6.6. Использование скрипта redundancyAreas для автоматизации настройки резервирования

Как видно из предыдущего раздела, настройка областей резервирования может оказаться сложной задачей. Для упрощения этой процедуры предусмотрен скрипт redundancyAreas.py, написанный на языке Python. Среда Procyon IDE поддерживает выполнение скриптов на языке Python (Рис. 6.6).

Скрипт redundancyAreas.py (скачать можно с официального сайта АО «Нефтеавтоматика») находит все переменные указанных программ, списков глобальных переменных и создает для них вызовы функции AreaRegister.

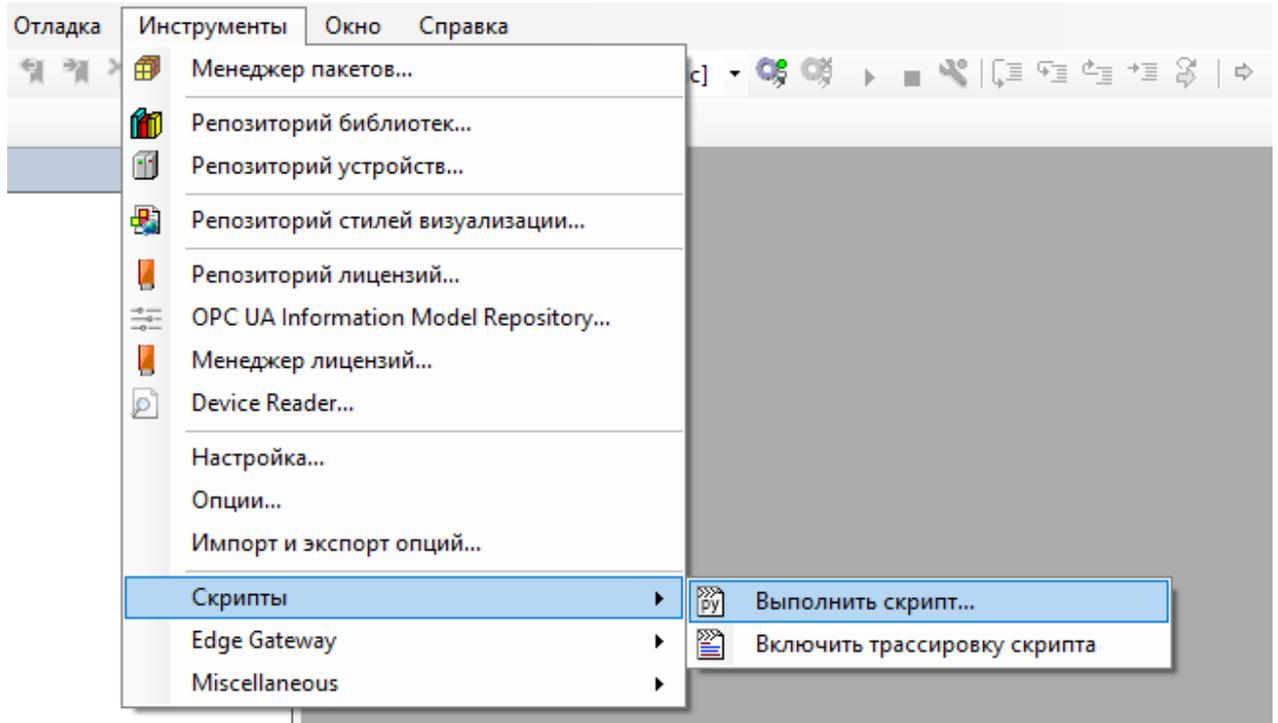


Рис. 6.6 - Запуск скрипта redundancyAreas.py

Для работы скрипта необходим настроечный файл в формате JSON. Файл должен располагаться в той же папке где располагается файл проекта и иметь такое же имя (без учета расширения) как и файл проекта. Формат настроечного файла представлен в Листинг 6.4.

```
{
  "POUs": "PLC_PRG",
  "GVLs": "GVL",
  "InitFunction": "areas"
}
```

Листинг 6.4 – Формат настроечного файла для скрипта redundancyAreas.py

Описание параметров настроечного файла:

- POU's - список программ, к которым необходимо применить скрипт (через пробел);
- GVL's - список Списков глобальных переменных к которым необходимо применить скрипт (через пробел);
- InitFunction - имя функции куда необходимо вставить соответствующие вызовы функции AreaRegister.

7. Работа с коммуникационными протоколами MKLogic-500

7.1. Протокол Modbus

Для поддержки протокола Modbus в дерево устройств добавляется устройство NftModbus. Это группирующее устройство, которое не содержит конфигурационных параметров и каналов ввода - вывода. Дочерним устройством к NftModbus является устройство NftModbusTCP реализующее протокол ModbusTCP. Это устройство также является группирующим, и не имеет каналов и конфигурационных параметров. Дочерними элементами устройства NftModbusTCP можно добавить устройства двух типов: NftModbus_Slave и NftModbus_Master, реализующие соответственно режимы “ведомого” и “ведущего” устройств протокола Modbus.

7.1.1. Режим “ведомого” в протоколе Modbus

Режим ведомого реализуется при добавлении в дерево устройств устройства NftModbus_Slave (см. Рис. 7.1).

Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Slave common					Общие настройки Modbus slave
Id	USINT(1..247)	1	1		Modbus slave id
CPU port number	Enumeration of USINT	Eth3	Eth3		Порт CPU
Modbus server port	UINT	502	502		TCP порт сервера Modbus
Modbus server max clients count	USINT	1	1		Максимальное количество клиентов modbus сервера
Modbus slave number	USINT				Номер Modbus slave
Holding registers info					Информация о массиве holding регистров для маппирования
Offset	UINT	0	0		Смещение массива
Size	UINT	0	0		Размер массива
Input registers info					Информация о массиве input регистров для маппирования
Offset	UINT	0	0		Смещение массива
Size	UINT	0	0		Размер массива
Coils info					Информация о массиве coils битов для маппирования
Offset	UINT	0	0		Смещение массива
Size	UINT	0	0		Размер массива
Discretes inputs info					Информация о массиве discrete битов для маппирования
Offset	UINT	0	0		Смещение массива
Size	UINT	0	0		Размер массива

Рис. 7.1 - Конфигурация режима “ведомый” протокола Modbus

- Id - Slave ID протокола Modbus;
- CPU port number - Порт Ethernet, который будет использовать сервер. Если выбрано All подключение будет открыто на всех доступных портах Ethernet;
- Modbus server port - TCP порт сервера;
- Modbus server max clients count - Максимальное количество клиентов, которое будет принимать сервер;
- Modbus slave number - идентификатор сервера для последующих вызовов функций библиотеки NftModbus
- Holding registers info - Информация о маппинге Holding registers;
- Input registers info - Информация о маппинге Input registers;
- Coils info - Информация о маппинге Coils;
- Discrete inputs info - Информация о маппинге Discrete inputs.

Структура маппинга:

— Offset - Начальный адрес карты памяти. С этого адреса будут располагаться легальные адреса. При попытке доступа к ячейкам с адресами меньше этого значения будет возвращаться ошибка Illegal data address;

— Size - Размер карты памяти. Последний доступный адрес будет Offset + Size и он не должен превышать 65535. При попытке доступа к ячейкам с адресами больше последнего доступного адреса будет возвращаться ошибка Illegal data address.

Дальнейшая настройка сервера производится при помощи библиотеки NftModbus поставляемой в пакете. Библиотека NftModbus представляет собой контейнерную библиотеку Procyon IDE, которая состоит из двух библиотек:

— NftModbusIfts - интерфейсная библиотека, содержащая все типы данных применяемые в работе с протоколом Modbus;

— NftModbusImpl - библиотека, содержащая основные функции для работы с протоколом Modbus.

Функции для работы с Modbus в режиме “ведомый” представлены в библиотеке NftModbusImpl в папке NftModbusImplementation/Functions/Slave.

Данные функции необходимо вызвать на этапе инициализации программы пользователя. Наиболее подходящим для этого является функция привязанная к событию DownloadDone, так же как при настройке резервирования.

Для создания такой функции необходимо перейти в «Конфигурация задач>Системные события» нажать на «Добавить обработчик событий». В открывшемся окне выбрать событие DownloadDone, ввести название функции и нажать кнопку ОК.

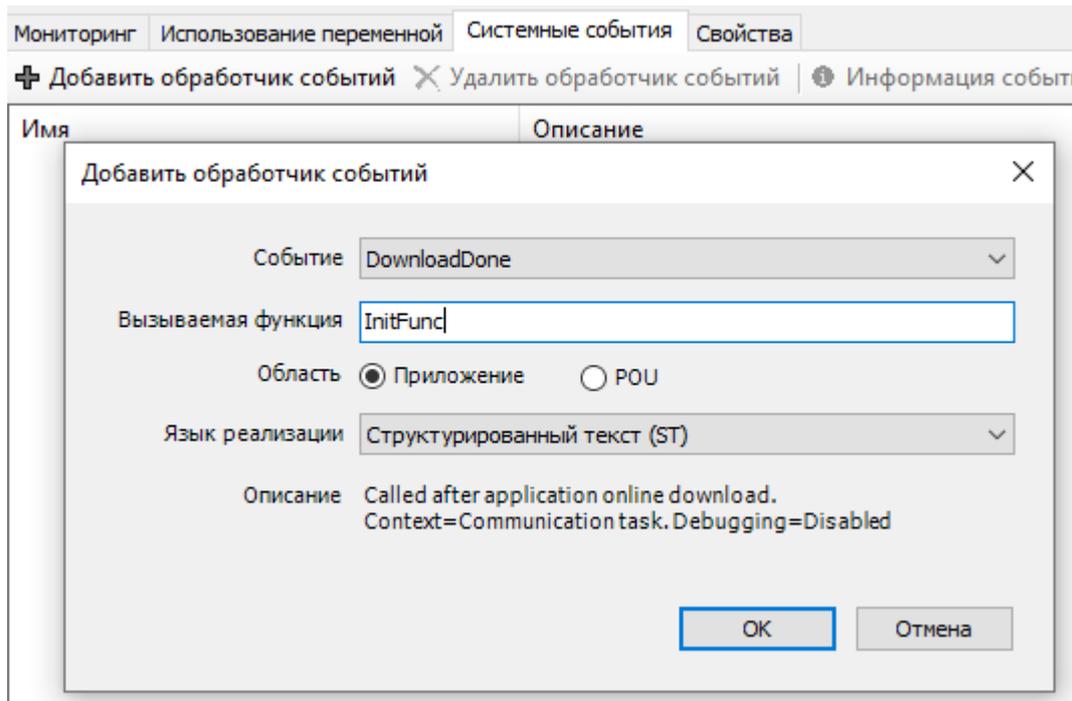


Рис. 7.2 - Создание функции через обработчик событий

Функции TcpSlaveMapBitData и TcpSlaveMapBitDataArray

Функция TcpSlaveMapBitData предназначена для привязки массива из программы пользователя к адресам карты Modbus.

Параметры и возвращаемые значения функции TcpSlaveMapBitData приведены в Табл. 7.1.

Таблица 7.1 – Параметры и возвращаемое значение функции TcpSlaveMapBitData

Область	Имя	Тип параметра	Описание
Return	TcpSlaveMapBitData	BOOL	TRUE если массив успешно связан
Input	SlaveHandle	USINT	Идентификатор сервера из конфигурации сервера в дереве устройств
Input	Offset	UINT	Начальный адрес данных, должен быть больше либо равен параметру Offset из конфигурации сервера в дереве устройств для соответствующего типа данных
Input	DataType	NftModbusIfts.ModbusSlaveBitDataType	Тип данных: Coils или Discrete Inputs
Input	pData	POINTER TO BOOL	Указатель на первый элемент привязываемого массива
Input	ulSize	UINT	Размер привязываемого массива
Input	ExcludeFromSynchronization	BOOL	Команда исключить из областей синхронизации
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpSlaveMapBitData представлен в Листинг 7.1.

```

VAR
  data : ARRAY [1..100] OF BOOL;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к COIL с адресами от 120 до 219
res := NftModbusImpl.TcpSlaveMapBitData(1, 120, NftModbusItfs.ModbusSlaveBitDataType.COILS, ADR
(data[1]), SIZEOF(data) / SIZEOF(BOOL), ulResult => result);
IF NOT res THEN
  ...
END_IF

```

Листинг 7.1 – Пример использования функции TcpSlaveMapBitData

Функция TcpSlaveMapBitDataArray является безопасной оберткой для функции TcpSlaveMapBitData. Она принимает на вход массив неизвестной заранее длины, а параметр ulSize при этом вычисляется автоматически.

Параметры и возвращаемые значения функции TcpSlaveMapBitDataArray приведены в Табл. 7.2.

Таблица 7.2 – Параметры и возвращаемое значение функции TcpSlaveMapBitDataArray

Область	Имя	Тип параметра	Описание
Return	TcpSlaveMap- BitData	BOOL	TRUE если массив успешно связан
Input	SlaveHandle	USINT	Идентификатор сервера из конфигурации сервера в дереве устройств
Input	Offset	UINT	Начальный адрес данных, должен быть больше либо равен параметру Offset из конфигурации сервера в дереве устройств для соответствующего типа данных
Input	DataType	NftModbusItfs.ModbusSlave BitDataType	Тип данных: Coils или Discrete Inputs
Input/ Output	DataArray	ARRAY[*] OF BOOL	Привязываемый массив
Input	ExcludeFrom Synchronization	BOOL	Команда исключить из областей синхронизации
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении

Пример использования функции TcpSlaveMapBitDataArray представлен в Листинг 7.2.

```

VAR
  data : ARRAY [1..100] OF BOOL;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к DISCRETE INPUTS с адресами от 120 до 219
res := NftModbusImpl.TcpSlaveMapBitDataArray(1, 120,
NftModbusItfs.ModbusSlaveBitDataType.DISCRETE_INPUTS, data, ulResult => result);
IF NOT res THEN
  ...
END_IF

```

Листинг 7.2 – Пример использования функции TcpSlaveMapBitDataArray

Функции TcpSlaveMapClientsDiag и TcpSlaveMapClientsDiagArray

Функция TcpSlaveMapClientsDiag предназначена для привязки массива диагностической информации из программы пользователя к внутренней диагностике Modbus сервера.

Параметры и возвращаемые значения функции TcpSlaveMapClientsDiag приведены в Табл. 7.3.

Таблица 7.3 – Параметры и возвращаемое значение функции TcpSlaveMapClientsDiag

Область	Имя	Тип параметра	Описание
Return	TcpSlaveMapClientsDiag	BOOL	TRUE если массив успешно связан
Input	SlaveHandle	USINT	Идентификатор сервера из конфигурации севера в дереве устройств
Input	pDiag	POINTER TO NftModbusItfs.ModbusClientsDiagnostics	Указатель на первый элемент привязываемого массива
Input	ulSize	DINT	Размер привязываемого массива
Output	usSizeOut	USINT	Возвращает максимальное количество клиентов разрешенное для данного сервера
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpSlaveMapClientsDiag представлен в Листинг 7.3.

```

VAR
  data : ARRAY [1..10] OF NftModbusItfs.ModbusClientsDiagnostics;
  res : BOOL;
  realSize : USINT;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к внутренней диагностике сервера
res := NftModbusImpl.TcpSlaveMapClientsDiag(1, ADR(data[1]), SIZEOF(data) / SIZEOF
(NftModbusItfs.ModbusClientsDiagnostics), usSizeOut => realSize, ulResult => result);
IF NOT res THEN
  ...
END_IF

```

Листинг 7.3 – Пример использования функции TcpSlaveMapClientsDiag

Функция TcpSlaveMapClientsDiagArray является безопасной оберткой для функции TcpSlaveMapClientsDiag. Она принимает на вход массив неизвестной заранее длины, а параметр ulSize при этом вычисляется автоматически.

Параметры и возвращаемые значения функции TcpSlaveMapClientsDiagArray приведены в Табл. 7.4.

Таблица 7.4 – Параметры и возвращаемое значение функции TcpSlaveMapClientsDiagArray

Область	Имя	Тип параметра	Описание
Return	TcpSlaveMapClientsDiag	BOOL	TRUE если массив успешно связан
Input	SlaveHandle	USINT	Идентификатор сервера из конфигурации севера в дереве устройств
Input	DiagArray	ARRAY[*] OF NftModbusItfs.ModbusClientsDiagnostics	Привязываемый массив
Output	usSizeOut	USINT	Возвращает максимальное количество клиентов разрешенное для данного сервера
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpSlaveMapClientsDiagArray представлен в Листинг 7.4.

```

VAR
  data : ARRAY [1..10] OF NftModbusItfs.ModbusClientsDiagnostics;
  res : BOOL;
  realSize : USINT;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...

```

```
// Привязка массива data к внутренней диагностике сервера
res := NftModbusImpl.TcpSlaveMapClientsDiag(1, data, usSizeOut => realSize, ulResult => result);
IF NOT res THEN
    ...
END_IF
```

Листинг 7.4 – Пример использования функции TcpSlaveMapClientsDiagArray

Функции TcpSlaveMapRegisterData и TcpSlaveMapRegisterDataArray

Функция TcpSlaveMapRegisterData предназначена для привязки массива из программы пользователя к адресам карты Modbus.

Параметры и возвращаемые значения функции TcpSlaveMapRegisterData приведены в Табл. 7.5.

Таблица 7.5 – Параметры и возвращаемое значение функции TcpSlaveMapRegisterData

Область	Имя	Тип параметра	Описание
Return	TcpSlaveMapRegisterData	BOOL	TRUE если массив успешно связан
Input	SlaveHandle	USINT	Идентификатор сервера из конфигурации сервера в дереве устройств
Input	Offset	UINT	Начальный адрес данных, должен быть больше либо равен параметру Offset из конфигурации сервера в дереве устройств для соответствующего типа данных
Input	DataType	NftModbusItfs.ModbusSlaveRegisterDataType	Тип данных: Input Registers или Holding Registers
Input	pData	POINTER TO WORD	Указатель на первый элемент привязываемого массива
Input	ulSize	DINT	Размер привязываемого массива
Input	ExcludeFromSynchronization	BOOL	Команда исключить из областей синхронизации
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpSlaveMapRegisterData представлен в Листинг 7.5.

```
VAR
    data : ARRAY [1..100] OF WORD;
    res : BOOL;
    result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к HOLDING REGISTERS с адресами от 120 до 219
res := NftModbusImpl.TcpSlaveMapRegisterData(1, 120,
NftModbusItfs.ModbusSlaveRegisterDataType.HOLDING_REGISTERS, ADR(data[1]), SIZEOF(data) / SIZEOF
(WORD), ulResult => result);
IF NOT res THEN
    ...
END_IF
```

Листинг 7.5 – Пример использования функции TcpSlaveMapRegisterData

Функция TcpSlaveMapRegisterDataArray является безопасной оберткой для функции TcpSlaveMapRegisterData. Она принимает на вход массив неизвестной заранее длины, а параметр ulSize при этом вычисляется автоматически.

Параметры и возвращаемые значения функции TcpSlaveMapRegisterDataArray приведены в Табл. 7.6.

Таблица 7.6 – Параметры и возвращаемое значение функции TcpSlaveMapRegisterDataArray

Область	Имя	Тип параметра	Описание
Return	TcpSlaveMapRegisterData	BOOL	TRUE если массив успешно связан

Область	Имя	Тип параметра	Описание
Input	SlaveHandle	USINT	Идентификатор сервера из конфигурации сервера в дереве устройств
Input	Offset	UINT	Начальный адрес данных, должен быть больше либо равен параметру Offset из конфигурации сервера в дереве устройств для соответствующего типа данных
Input	DataType	NftModbusIifs.ModbusSlaveRegisterDataType	Тип данных: Input Registers или Holding Registers
Input/Output	DataArray	ARRAY[*] OF WORD	Привязываемый массив
Input	ExcludeFromSynchronization	BOOL	Команда исключить из областей синхронизации
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpSlaveMapRegisterDataArray представлен в Листинг 7.6.

```

AR
  data : ARRAY [1..100] OF WORD;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к INPUT REGISTERS с адресами от 120 до 219
res := NftModbusImpl.TcpSlaveMapRegisterDataArray(1, 120, NftMod-
busIifs.ModbusSlaveRegisterDataType.INPUT_REGISTERS, data, ulResult => result);
IF NOT res THEN
  ...
END_IF

```

Листинг 7.6 – Пример использования функции TcpSlaveMapRegisterDataArray

7.1.2. Режим “ведущий” в протоколе Modbus

Режим “ведущий” реализуется при добавлении в дерево устройств устройства NftModbus_Master (см. Рис. 7.3).

Параметр	Тип	Значение	Знач...	Единица	Описание
Master common					Общие настройки Modbus master
Cycle time	UINT	250	250	мс	Время цикла опроса
CPU port number	Enumeration of USINT	Eth3	Eth3		Порт CPU
Modbus master number	USINT				Номер Modbus master
Register mapping array size (READ direction)	UINT	0	0		Размер массива регистров для маппирования в направлении чтения
Register mapping array size (WRITE direction)	UINT	0	0		Размер массива регистров для маппирования в направлении записи
Bits mapping array size (READ direction)	UINT	0	0		Размер массива битов для маппирования в направлении чтения
Bits mapping array size (WRITE direction)	UINT	0	0		Размер массива битов для маппирования в направлении записи

Рис. 7.3 - Конфигурация режима “ведущий” протокола Modbus

- Cycle time - Время цикла опроса;
- CPU port number - Порт Ethernet, который будет использовать клиент;
- Modbus master number - идентификатор клиента для последующих вызовов функций библиотеки NftModbus;
- Register mapping array size (READ direction) - размер массива регистров для маппирования в направлении чтения;
- Register mapping array size (WRITE direction) - размер массива регистров для маппирования в направлении записи;
- Bits mapping array size (READ direction) - размер массива битов для маппирования в направлении чтения;
- Bits mapping array size (WRITE direction) - размер массива битов для маппирования в направлении записи.

Дальнейшая настройка клиента производится при помощи библиотеки NftModbus поставляемом в пакете.

Функции для работы с Modbus в режиме “ведущий” представлены в библиотеке NftModbusImpl в папке NftModbusImplementation/Functions/Master.

Данные функции необходимо вызвать на этапе инициализации программы пользователя. Наиболее подходящим для этого является функция привязанная к событию DownloadDone.

Функции TcpMasterAddRequest и TcpMasterAddRequest2

Функция TcpMasterAddRequest предназначена для добавления в барабан запросов новый запрос с привязкой структур ModbusTcpRequest и ModbusRequestCommand из программы пользователя к новому запросу.

Параметры и возвращаемые значения функции TcpMasterAddRequest приведены в Табл. 7.7.

Таблица 7.7 – Параметры и возвращаемое значение функции TcpMasterAddRequest

Область	Имя	Тип параметра	Описание
Return	TcpMasterAddRequest	BOOL	TRUE если массив успешно связан
Input	MasterHandle	USINT	Идентификатор клиента из конфигурации клиента в дереве устройств
Input	Request	POINTER TO NftModbusItfs.ModbusTcpRequest	Указатель на структуру ModbusTcpRequest
Input	Cmd	POINTER TO NftModbusItfs.ModbusRequestCommand	Указатель на структуру ModbusRequestCommand
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при не успешном выполнении функции

Пример использования функции TcpMasterAddRequest представлен в Листинг 7.7.

```

VAR
  requests : ARRAY [1..10] OF NftModbusItfs.ModbusTcpRequest;
  commands : ARRAY [1..10] OF NftModbusItfs.ModbusRequestCommand;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
  i : INT;
END_VAR
...
// Добавление в клиент 10 запросов с привязкой к ним массивов requests и commands
FOR i := 1 TO 10 DO
  res := NftModbusImpl.TcpMasterAddRequest(1, ADR(requests[i]), ADR(commands[i]), ulResult =>
  result);
  IF NOT res THEN
    ...
  END_IF
END_FOR

```

Листинг 7.7 – Пример использования функции TcpMasterAddRequest

Функция TcpMasterAddRequest2 является расширенной версией функции TcpMasterAddRequest. Она позволяет добавить структуру, в которой будет отображаться диагностическая информация по текущему запросу.

Параметры и возвращаемые значения функции TcpMasterAddRequest2 приведены в Табл. 7.8.

Таблица 7.8 – Параметры и возвращаемое значение функции TcpMasterAddRequest2

Область	Имя	Тип параметра	Описание
Return	TcpMasterAddRequest	BOOL	TRUE если массив успешно связан
Input	MasterHandle	USINT	Идентификатор клиента из конфигурации клиента в дереве устройств
Input	Request	POINTER TO NftModbusItfs.ModbusTcpRequest	Указатель на структуру ModbusTcpRequest
Input	Cmd	POINTER TO NftModbusItfs.ModbusRequestCommand	Указатель на структуру ModbusRequestCommand
Input	Status	POINTER TO NftModbusItfs.ModbusRequestStatus	Указатель на структуру ModbusRequestStatus
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении

Область	Имя	Тип параметра	Описание
			функции

Пример использования функции TcpMasterAddRequest2 представлен в Листинг 7.8.

```

VAR
  requests : ARRAY [1..10] OF NftModbusItfs.ModbusTcpRequest;
  commands : ARRAY [1..10] OF NftModbusItfs.ModbusRequestCommand;
  statuses : ARRAY [1..10] OF NftModbusItfs.ModbusRequestStatus;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
  i : INT;
END_VAR
...
// Добавление в клиент 10 запросов с привязкой к ним массивов requests, commands и statuses
FOR i := 1 TO 10 DO
  res := NftModbusImpl.TcpMasterAddRequest(1, ADR(requests[i]), ADR(commands[i]), ADR(statuses
[i]), ulResult => result);
  IF NOT res THEN
    ...
  END_IF
END_FOR

```

Листинг 7.8 – Пример использования функции TcpMasterAddRequest2

Структура ModbusTcpRequest расширяет структуру ModbusRequest и добавляет дополнительные элементы (см. Табл. 7.9)

Таблица 7.9 – Описание дополнительных полей структуры ModbusTcpRequest

Имя	Тип параметра	Описание
IpAddress	ARRAY[1..4] OF USINT	IP адрес сервера к которому выполняется запрос
TcpPort	UINT	TCP порт сервера к которому выполняется запрос

Элементы структуры ModbusRequest описания в Табл. 7.10

Таблица 7.10 – Описание полей структуры ModbusRequest

Имя	Тип параметра	Описание
SlaveId	USINT	Modbus идентификатор сервера
FunctionCode	ModbusFunction	Код функции Modbus
Address	UINT	Начальный адрес данных
Count	UINT	Количество адресов данных
TimeoutMs	UINT	Таймаут ожидания ответа на запрос, мс
DelayBeforeMs	UINT	Задержка перед выполнением запроса, мс
Repeats	UINT	Количество повторов при таймауте
SkipRepeatsWhenBad	UINT	Число запросов, которое пропускается перед следующим запросом не ответившего устройства
SingleRequest	BOOL	Флаг однократного запроса
OnModifyRequest	BOOL	Флаг выполнения запроса по изменению данных функций записи
RepeatOverScan	BOOL	Флаг повтора при неудачном запросе. Если значение равно TRUE, запрос повторяется через скан, иначе повторяется сразу
ResultOffset	UDINT	Смещение в массиве данных. В зависимости от типа запроса используется один из 4х массивов данных
pExtParam	POINTER TO ModbusRequestExt	Дополнительные параметры для функции READ_WRITE_MULTIPLE_REGISTERS (см. Табл. 7.11)

Таблица 7.11 – Структура дополнительных параметров для функции READ_WRITE_MULTIPLE_REGISTERS

Имя	Тип параметра	Описание
Address	UINT	Дополнительный начальный адрес
Count	UINT	Дополнительное количество адресов данных
ResultOffset	UDINT	Дополнительное смещение в массиве данных

Таблица 7.12 – Использование полей структуры ModbusRequest и массивов данных клиента

Функция Modbus	Массив данных клиента	Назначение полей структуры ModbusRequest
READ_COILS	Bits READ	Address - начальный адрес COILS Count - количество COILS ResultOffset - смещение в массиве данных клиента
READ_DISCRETE_INPUTS	Bits READ	Address - начальный адрес DISCRETE INPUTS Count - количество DISCRETE INPUTS ResultOffset - смещение в массиве данных клиента
READ_HOLDING_REGISTERS	Register READ	Address - начальный адрес HOLDING REGISTERS Count - количество HOLDING REGISTERS ResultOffset - смещение в массиве данных клиента
READ_INPUT_REGISTERS	Register READ	Address - начальный адрес INPUT REGISTERS Count - количество INPUT REGISTERS ResultOffset - смещение в массиве данных клиента
WRITE_SINGLE_COIL	Bits WRITE	Address - начальный адрес COILS Count - количество COILS ResultOffset - смещение в массиве данных клиента
WRITE_SINGLE_REGISTER	Register WRITE	Address - начальный адрес HOLDING REGISTERS Count - количество HOLDING REGISTERS ResultOffset - смещение в массиве данных клиента
WRITE_MULTIPLE_COILS	Bits WRITE	Address - начальный адрес COILS Count - количество COILS ResultOffset - смещение в массиве данных клиента
WRITE_MULTIPLE_REGISTERS	Register WRITE	Address - начальный адрес HOLDING REGISTERS Count - количество HOLDING REGISTERS ResultOffset - смещение в массиве данных клиента
MASK_WRITE_REGISTER	Register WRITE	Address - адрес HOLDING REGISTERS Count - игнорируется ResultOffset - смещение в массиве данных клиента (из массива данных берется 2 регистра: 1 - AND-mask, 2 - OR-mask)
READ_WRITE_MULTIPLE_REGISTERS	Register WRITE Register READ	Address - адрес HOLDING REGISTERS для записи Count - количество HOLDING REGISTERS для записи ResultOffset - смещение в массиве данных клиента для записи pExtParam^.Address - адрес HOLDING REGISTERS для чтения pExtParam^.Count - количество HOLDING REGISTERS для чтения pExtParam^.ResultOffset - смещение в массиве данных клиента для чтения

 **Примечание** Функции MASK_WRITE_REGISTER и READ_WRITE_MULTIPLE_REGISTERS пока не реализованы.

Таблица 7.13 – Описание полей структуры ModbusRequestCommand

Имя	Тип параметра	Описание
Enable	BOOL	Разрешение запроса
DoSingle	BOOL	Выполнить однократный запрос

Таблица 7.14 – Описание полей структуры ModbusRequestStatus

Имя	Тип параметра	Описание
Status	ModbusRequestError	Статус запроса см. описание типа ModbusRequestError

Имя	Тип параметра	Описание
ExecuteTime	UINT	Время выполнения запроса
Repeats	UINT	Количество повторов
Skips	UINT	Количество пропусков
CntTotal	UDINT	Общее количество запросов
CntGood	UDINT	Количество успешно выполненных запросов

Функции TcpMasterMapBitData и TcpMasterMapBitDataArray

Функция TcpMasterMapBitData предназначена для привязки массива из программы пользователя к адресам карты Modbus.

Параметры и возвращаемые значения функции TcpMasterMapBitData приведены в Табл. 7.15.

Таблица 7.15 – Параметры и возвращаемое значение функции TcpMasterMapBitData

Область	Имя	Тип параметра	Описание
Return	TcpMasterMapBitData	BOOL	TRUE если массив успешно связан
Input	MasterHandle	USINT	Идентификатор клиента из конфигурации клиента в дереве устройств
Input	Offset	UINT	Смещение в массиве данных из конфигурации клиента
Input	DataDirection	NftModbusItfs.ModbusMasterDataDirection	Направление потока данных: Чтение или запись
Input	pData	POINTER TO BOOL	Указатель на первый элемент привязываемого массива
Input	ulSize	UINT	Размер привязываемого массива
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpMasterMapBitData представлен в Листинг 7.9.

```

VAR
    data : ARRAY [1..100] OF BOOL;
    res : BOOL;
    result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к массиву данных мастера для чтения со смещением 120, т.е. 1 элемент
// массива data будет
// соответствовать 120-му элементу массива мастера
res := NftModbusImpl.TcpMasterMapBitData(1, 120, NftModbusItfs.ModbusMasterDataDirection.READ,
ADR(data[1]), SIZEOF(data) / SIZEOF(BOOL), ulResult => result);
IF NOT res THEN
    ...
END_IF

```

Листинг 7.9 – Пример использования функции TcpMasterMapBitData

Функция TcpMasterMapBitDataArray является безопасной обёрткой для функции TcpMasterMapBitData. Она принимает на вход массив неизвестной заранее длины, а параметр ulSize при этом вычисляется автоматически.

Параметры и возвращаемые значения функции TcpMasterMapBitDataArray приведены в Табл. 7.16.

Таблица 7.16 – Параметры и возвращаемое значение функции TcpMasterMapBitDataArray

Область	Имя	Тип параметра	Описание
Return	TcpMasterMapBitData	BOOL	TRUE если массив успешно связан
Input	MasterHandle	USINT	Идентификатор клиента из конфигурации клиента в дереве устройств
Input	Offset	UINT	Смещение в массиве данных из конфигурации клиента
Input	DataDirection	NftModbusItfs.ModbusMasterDataDirection	Направление потока данных: Чтение или запись

Область	Имя	Тип параметра	Описание
Input/ Output	dataArray	ARRAY[*] OF BOOL	Привязываемый массив
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpMasterMapBitDataArray представлен в Листинг 7.10.

```

VAR
  data : ARRAY [1..100] OF BOOL;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к массиву данных мастера для записи со смещением 120, т.е. 1 элемент
// массива data будет
// соответствовать 120-му элементу массива мастера
res := NftModbusImpl.TcpMasterMapBitDataArray(1, 120,
NftModbusItfs.ModbusMasterDataDirection.WRITE, data, ulResult => result);
IF NOT res THEN
  ...
END_IF

```

Листинг 7.10 – Пример использования функции TcpMasterMapBitDataArray

Функции TcpMasterMapRegisterData и TcpMasterMapRegisterDataArray

Функция TcpMasterMapRegisterData предназначена для привязки массива из программы пользователя к адресам карты Modbus.

Параметры и возвращаемые значения функции TcpMasterMapRegisterData приведены в Табл. 7.17.

Таблица 7.17 – Параметры и возвращаемое значение функции TcpMasterMapRegisterData

Область	Имя	Тип параметра	Описание
Return	TcpMasterMapRegisterData	BOOL	TRUE если массив успешно связан
Input	MasterHandle	USINT	Идентификатор клиента из конфигурации клиента в дереве устройств
Input	Offset	UINT	Смещение в массиве данных из конфигурации клиента
Input	DataDirection	NftModbusItfs.ModbusMasterDataDirection	Направление потока данных: Чтение или запись
Input	pData	POINTER TO WORD	Указатель на первый элемент привязываемого массива
Input	ulSize	UINT	Размер привязываемого массива
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpMasterMapRegisterData представлен в Листинг 7.11.

```

VAR
  data : ARRAY [1..100] OF WORD;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к массиву данных мастера для записи со смещением 120, т.е. 1 элемент
// массива data будет
// соответствовать 120-му элементу массива мастера
res := NftModbusImpl.TcpMasterMapRegisterData(1, 120,
NftModbusItfs.ModbusMasterDataDirection.WRITE, ADR(data[1]), SIZEOF(data) / SIZEOF(WORD),
ulResult => result);

```

```
IF NOT res THEN
  ...
END_IF
```

Листинг 7.11 – Пример использования функции TcpMasterMapRegisterData

Функция TcpMasterMapRegisterDataArray является безопасной обёрткой для функции TcpMasterMapRegisterData. Она принимает на вход массив неизвестной заранее длины, а параметр ulSize при этом вычисляется автоматически.

Параметры и возвращаемые значения функции TcpMasterMapRegisterDataArray приведены в Табл. 7.18.

Таблиц 7.18 – Параметры и возвращаемое значение функции TcpMasterMapRegisterDataArray

Область	Имя	Тип параметра	Описание
Return	TcpMasterMapRegisterData	BOOL	TRUE если массив успешно связан
Input	MasterHandle	USINT	Идентификатор клиента из конфигурации клиента в дереве устройств
Input	Offset	UINT	Смещение в массиве данных из конфигурации клиента
Input	DataDirection	NftModbusItfs.ModbusMasterDataDirection	Направление потока данных: Чтение или запись
Input/Output	DataArray	ARRAY[*] OF WORD	Привязываемый массив
Output	ulResult	NftModbusImpl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции TcpMasterMapRegisterDataArray представлен в Листинг 7.12.

```
VAR
  data : ARRAY [1..100] OF WORD;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR
...
// Привязка массива data к массиву данных мастера для чтения со смещением 120, т.е. 1 элемент массива data будет
// соответствовать 120-му элементу массива мастера
res := NftModbusImpl.TcpMasterMapRegisterDataArray(1, 120,
NftModbusItfs.ModbusMasterDataDirection.READ, data, ulResult => result);
IF NOT res THEN
  ...
END_IF
```

Листинг 7.12 – Пример использования функции TcpMasterMapRegisterDataArray

! **ВНИМАНИЕ** Для корректной работы резервирования в функции регистрации массивов в области Modbus запрещено передавать адреса переменных указателей, ссылок и разного рода контейнеров содержащих их. Вместо указателей и ссылок необходимо адрес передавать непосредственно переменные, от которых взяты ссылки или указатели.

! **ВНИМАНИЕ** Если проект предназначен на работу в резерве, для всех привязанных массивов данных и диагностической информации автоматически вызывается функция AreaRegister из компонента Redundancy.

7.2. Протокол IEC 60870-5-104 (сервер)

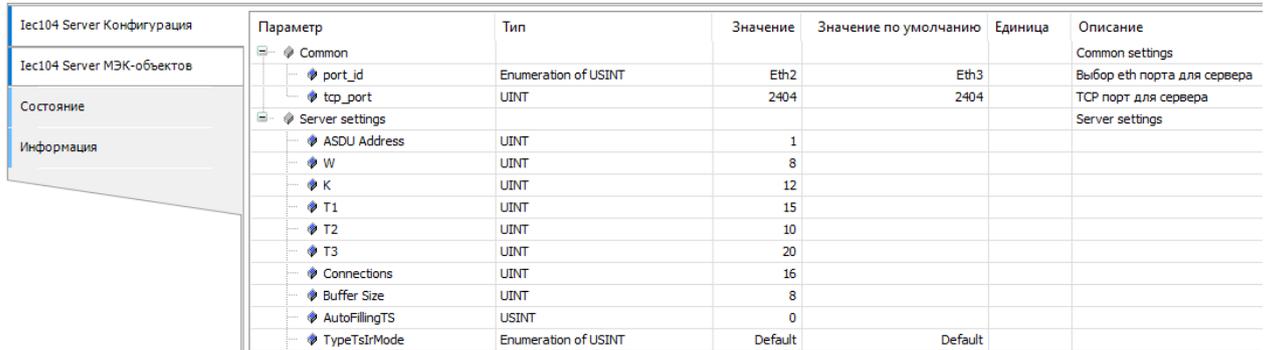
Для поддержки протокола IEC 60870-5-104 в дерево устройств добавляется устройство NftIec104. Это группирующее устройство, которое не содержит конфигурационных параметров и каналов ввода-вывода. Дочерним устройством к NftIec104 является устройство NftIec104_Server

реализующее протокол IEC 60870-5-104. При необходимости включить в проект дополнительный сервер IEC 60870-5-104, нужно в группирующем устройстве Nftlec104 добавить дочернее устройство Nftlec104_Server (поддерживается до 5 устройств Nftlec104_Server).

! **ВНИМАНИЕ** Не допускается добавление второго группирующего устройства Nftlec104. Данное устройство должно быть в единственном экземпляре.

7.2.1. Режим ведомого в протоколе IEC 60870-5-104

Режим ведомого реализуется при добавлении в дерево устройств устройства Nftlec104_Server (см. Рис. 4.2).



Параметр	Тип	Значение	Значение по умолчанию	Единица	Описание
Common					
port_id	Enumeration of USINT	Eth2	Eth3		Выбор eth порта для сервера
tcp_port	USINT	2404	2404		TCP порт для сервера
Server settings					
ASDU Address	USINT	1			
W	USINT	8			
K	USINT	12			
T1	USINT	15			
T2	USINT	10			
T3	USINT	20			
Connections	USINT	16			
Buffer Size	USINT	8			
AutoFillingTS	USINT	0			
TypeTslrMode	Enumeration of USINT	Default	Default		

Рис. 7.4 - Конфигурация режима “ведомый” протокола IEC 60870-5-104

Параметры ASDU Address, W, K, T1-T3 должны совпадать с параметрами мастера.

Конфигурационные настройки сервера:

- port_id - порт Ethernet, который будет использовать сервер. Если выбрано All подключение будет открыто на всех доступных портах Ethernet;
- tcp_port - TCP порт сервера;
- server number - порядковый номер сервера (должен быть уникален для каждого экземпляра сервера IEC 60870-5-104);
- ASDU Address - общий адрес ASDU для устройства;
- W - последнее подтверждение после приема W APDU;
- K - максимальная разность между переменной состояния передачи и номером последнего подтвержденного APDU;
- T1 - таймаут при отправке или тестировании APDU, в секундах;
- T2 - таймаут для подтверждения в случае отсутствия сообщения с данными (меньше T1) , в секундах;
- T3 - таймаут для отправки блоков тестирования в случае долгого простоя, в секундах (0 – не посылать блоки тестирования);
- Connections - Максимально доступное количество подключений с адресов;
- Buffer Size - Размер буферов для каждого соединения в МБ (от 1 до 8, по умолчанию 1 МБ);
- AutoFillingTS - Используется для автозаполнения меток времени (1 – включает автозаполнение поля timestamp). Время меток берётся из процессорного модуля;
- TypeTslrMode - режим поведения типов с меткой времени при общем опросе станции: Default - стандартный режим работы протокола (без замены типо и 20-й причиной передачи) , Sporadic - замена причины передачи с 20 (INTERROGATED_COT) на 3 (SPONTANEOUS), Mirror - замена типа с меткой времени на аналогичный без метки времени при общем опросе.

Дальнейшая настройка сервера производится при помощи библиотеки Nftlec104 поставляемой в пакете. Библиотека Nftlec104 представляет собой контейнерную библиотеку Procyon IDE, которая состоит из двух библиотек: - Nftlec104Ifs - интерфейсная библиотека, содержащая все типы данных применяемые в работе с протоколом IEC104. - Nftlec104Impl - библиотека, содержащая основные функции для работы с протоколом IEC104. Функции для работы с IEC104 в режиме “ведомый” представлены в библиотеке Nftlec104Impl в папке Nftlec104Implementation/Functions/Server. Данные функции необходимо вызвать на этапе инициализации программы пользователя. Наиболее подходящим для этого является функция привязанная к событию DownloadDone, так же как при настройке резервирования.

Для создания такой функции необходимо перейти в «Конфигурация задач->Системные события» нажать на «Добавить обработчик событий». В открывшемся окне выбрать событие DownloadDone, ввести название функции и нажать кнопку ОК.

! **ВНИМАНИЕ** Не допускается подключение двух клиентов с одного IP-адреса к одному серверу. Это приведет к нестабильной работе сервера и последующего отказа сервера.

Функции AddlecVariable

Функция AddlecVariable предназначена для создания элементов данных, передаваемых по IEC104.

Параметры и возвращаемые значения функции AddlecVariable приведены в Табл. 7.19.

Таблица 7.19 – Параметры и возвращаемое значение функции AddlecVariable

Область	Имя	Тип параметра	Описание
Return	AddlecVariable	BOOL	TRUE если массив/переменная успешно связан(-ы)
Input	Handle	USINT	Порядковый номер сервера из конфигурации
Input	IecVariable	POINTER TO BYTE	Указатель на первый элемент привязываемого массива
Input	DataSize	UINT	Размер привязываемого массива
Input	VarType	NftIec104Itfs.IEC_60870_5_VARIABLES	Тип переменной
Input	IOA	UDINT	Уникальный адрес объекта информации
Input	IsSporadic	BOOL	Является ли объект информации спорадически передаваемым
Input	IsCyclic	BOOL	Передавать ли данные объекта информации с каждым циклом
Output	ulResult	NftIec104Impl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции AddlecVariable представлен в Листинг 7.13.

```

VAR
  data1 : NftIec104Itfs.Iec104BoTsData;
  data2 : ARRAY [1..100] OF NftIec104Itfs.Iec104MecTsData;
  res1 : BOOL;
  result1 : NftModbusImpl.RTS_IEC_RESULT;
  res2 : BOOL;
  result2 : NftModbusImpl.RTS_IEC_RESULT;
END_VAR

...

// Привязка переменной data1
res1 := NftIec104Impl.AddIecVariable(1, ADR(data1), SIZEOF(data1), NftIec104Itfs.IEC_60870_5_VARIABLES.M_BO_TB_1, 1, TRUE, FALSE, ulResult => result1);
// Привязка массива переменных data2
res2 := NftIec104Impl.AddIecVariable(1, ADR(data2), SIZEOF(data2), NftIec104Itfs.IEC_60870_5_VARIABLES.M_ME_TF_1, 2, TRUE, FALSE, ulResult => result2);

IF NOT res1 THEN
  ...
END_IF

IF NOT res2 THEN
  ...
END_IF

```

Листинг 7.13 – Пример использования функции AddlecVariable

Функции AddlecCommand

Функция AddlecCommand предназначена для создания элементов команд, передаваемых по IEC104.

Параметры и возвращаемые значения функции AddlecCommand приведены в Табл. 7.20.

Таблица 7.20 – Параметры и возвращаемое значение функции AddlecCommand

Область	Имя	Тип параметра	Описание
Return	AddlecCommand	BOOL	TRUE если массив/переменная успешно связан(-ы)
Input	Handle	USINT	Порядковый номер сервера из конфигурации
Input	IecVariable	POINTER TO BYTE	Указатель на первый элемент привязываемого массива
Input	DataSize	UINT	Размер привязываемого массива
Input	VarType	NftIec104Itfs.IEC_60870_5_VARIABLES	Тип команды
Input	IOA	UDINT	Уникальный адрес объекта информации
Input	IsSelectable	BOOL	При наличии этого параметра значения команд меняются только, если пришла команда с выставленным флагом select, и затем в течение указанного периода пришла команда с флагом execute. Период задаётся в значении параметра SelectableTime, в мс.
Input	SelectableTime	UINT	Допускается значение от 5 до 65535 мс, при выходе за пределы принимается значение 10 мс.
Output	ulResult	NftIec104Impl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции AddlecCommand представлен в Листинг 7.14.

```

VAR
  cmd1 : NftIec104Itfs. Iec104MecTsCommand;
  cmd2 : ARRAY [1..100] OF NftIec104Itfs.Iec104MecTsCommand;

  res1 : BOOL;
  result1 : NftModbusImpl.RTS_IEC_RESULT;
  res2 : BOOL;
  result2 : NftModbusImpl.RTS_IEC_RESULT;
END_VAR

...

// Привязка команды cmd1
res1 := NftIec104Impl.AddIecCommand(1, ADR(POU.newCmd1), SIZEOF(POU.newCmd1), NftIec104Itfs.IEC_
60870_5_COMMANDS.C_SE_TC_1, 1, FALSE, 0, FALSE, ulResult => result1);
// Привязка массива команд cmd2
res2 := NftIec104Impl.AddIecCommand(1, ADR(POU.newCmd2), SIZEOF(POU.newCmd2), NftIec104Itfs.IEC_
60870_5_COMMANDS.C_SE_TC_1, 2, FALSE, 0, FALSE, ulResult => result2);

IF NOT res1 THEN
  ...
END_IF

IF NOT res2 THEN
  ...
END_IF

```

Листинг 7.14 – Пример использования функции AddlecCommand

Функции AddServerDiagnostics

Функция AddServerDiagnostics предназначена для создания элементов данных, передаваемых по IEC104.

Структура Iec104ServerDiagnostics содержит в себе диагностические данные сервера IEC 60870-5-104: — поле IsWork показывает, запущен сервер IEC 60870-5-104 или нет; — в поле ClientsCount передается число подключенных клиентов IEC 60870-5-104; — поле Status содержит код статуса работы сервера IEC 60870-5-104 (Табл. 7.21); — поле ErrorCode содержит код ошибки работы сервера IEC 60870-5-104 (Табл. 7.21); — поле TimeSyncCount показывает, сколько запросов синхронизации времени приходило; — поле SyncTime показывает метку времени, которая пришла в последней команде синхронизации времени.

Таблица 7.21 – Расшифровка статусов работы сервера IEC 60870-5-104

Поле Status	Поле ErrorCode	Расшифровка
0	OK	Сервер работает без ошибок
1	INIT_ERROR	Ошибка инициализации сервера
2	UNKNOWN_TYPE	Неизвестный тип IEC 60870-5-104
3	WRONG_COMBINATION	Недопустимая комбинация тип IEC 60870-5-104-IOA
4	CLIENT_CONNECTING	Выполняется подключение клиента
5	SERVER_STOPPED	Сервер остановлен

Параметры и возвращаемые значения функции AddServerDiagnostics приведены в (Табл. 7.22)

Таблица 7.22 – Параметры и возвращаемое значение функции AddServerDiagnostics

Область	Имя	Тип параметра	Описание
Return	AddServerDiagnostics	BOOL	TRUE если массив/переменная успешно связан(-ы)
Input	Handle	USINT	Порядковый номер сервера из конфигурации
Input	Diagnostics	POINTER TO NftIec104Itfs.Iec104ServerDiagnostics	Указатель на первый элемент привязываемой переменной
Output	ulResult	NftIec104Impl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции AddServerDiagnostics представлен в Листинг 7.15.

```

VAR
  diag : NftIec104Itfs.Iec104ServerDiagnostics;
  res : BOOL;
  result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR

...

// Привязка переменной diag к внутренней диагностике сервера
res := NftIec104Impl.AddServerDiagnostics(1, ADR(diag), ulResult => result);

IF NOT res THEN
  ...
END_IF

```

Листинг 7.15 – Пример использования функции AddServerDiagnostics

Функции ForcedUpdateVariable

Функция «ForcedUpdateVariable» предназначена для принудительного обновления в памяти сервера 60870-5-104 текущих значений из программы пользователя.

Параметры и возвращаемые значения функции «ForcedUpdateVariable» приведены в Табл. 7.23.

Таблица 7.23 – Параметры и возвращаемое значение функции ForcedUpdateVariable

Область	Имя	Тип параметра	Описание
Return	ForcedUpdateVariable	BOOL	TRUE если массив/переменная успешно связан(-ы)

Область	Имя	Тип параметра	Описание
Input	Handle	USINT	Порядковый номер сервера из конфигурации
Output	ulResult	NftIec104Impl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции ForcedUpdateVariable представлен в Листинг 7.16.

```

VAR
    res : BOOL;
    result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR

...

res := NftIec104Impl.ForcedUpdateVariable(1, ulResult => result);

IF NOT res1 THEN
    ...
END_IF
    
```

Листинг 7.16 – Пример использования функции ForcedUpdateVariable

Функции GetDecIpConnectedClient

Функция «GetDecIpConnectedClient» предназначена для получения списка IP- адресов подключенных клиентов. Результатом выполнения функции будет заполняющийся массив IP-адресов подключенных клиентов (используется альтернативная форма записи IP-адреса в виде числа).

Параметры и возвращаемые значения функции «GetDecIpConnectedClient» приведены в Табл. 7.24.

Таблица 7.24 – Параметры и возвращаемое значение функции GetDecIpConnectedClient

Область	Имя	Тип параметра	Описание
Return	GetDecIpConnectedClient	BOOL	TRUE если массив/переменная успешно связан(-ы)
Input	Handle	USINT	Порядковый номер сервера из конфигурации
Input	pointerForArray	POINTER TO BYTE	Указатель на первый элемент привязываемого массива
Input	DataSize	UINT	Размер привязываемого массива
Output	ulResult	NftIec104Impl.RTS_IEC_RESULT	Код ошибки при неуспешном выполнении функции

Пример использования функции GetDecIpConnectedClient представлен в Листинг 7.17.

```

VAR
    arrayIP : ARRAY [1..10] OF DWORD;
    res : BOOL;
    result : NftModbusImpl.RTS_IEC_RESULT;
END_VAR

...

res := NftIec104Impl.GetDecIpConnectedClient(1, ADR(POU.arrayIP), SIZEOF(POU.arrayIP), ulResult => result);

IF NOT res1 THEN
    ...
END_IF
    
```

Листинг 7.17 – Пример использования функции GetDecIpConnectedClient

8. Использование дополнительных библиотек MKLogic-500

8.1. Библиотека NftSys

Библиотека предназначена для предоставления дополнительных системных возможностей для работы. Библиотека представляет следующие функции: FTPEnable, RestoreRetainsFromFile, StoreRetainsInFile, CreateHistoricalDataFile, StoreHistoricalDataInFile, ReadRealFromFile, ReadReal2DimArray, WriteRealToFile, WriteReal2DimArray.

8.1.1. Функция FTPEnable

Функция FTPEnable предназначена для включения/выключения FTP-сервера.

Параметры и возвращаемые значения функции FTPEnable приведены в Табл. 8.1.

Таблица 8.1 – Параметры и возвращаемое значение функции FTPEnable

Область	Имя	Тип параметра	Описание
Return	FTPEnable	BOOL	TRUE если операция выполнена успешно
Input	enable	BOOL	TRUE - разрешить работу FTP-сервера, FALSE - запретить работу FTP-сервера

Пример использования функции FTPEnable представлен в Листинг 8.1:

```
// Включение FTP-сервера
res := NftSys.FTPEnable(TRUE);
...
// Выключение FTP-сервера
res := NftSys.FTPEnable(FALSE);
```

Листинг 8.1 – Пример использования функции FTPEnable

8.1.2. Функция StoreRetainsInFile

Функция StoreRetainsInFile предназначена для принудительной записи Retain данных. Данную функцию необходимо вызывать в цикле прикладной программы. После выполнения функции переданный в неё параметр с TRUE перейдёт в FALSE.

Параметры и возвращаемые значения функции StoreRetainsInFile приведены в Табл. 8.2.

Таблица 8.2 – Параметры и возвращаемое значение функции StoreRetainsInFile

Область	Имя	Тип параметра	Описание
Return	StoreRetainsInFile	BOOL	TRUE если операция выполнена успешно
Input	fSaveRetain	REFERENCE TO BOOL	TRUE - выполнить запись retain данных

Пример использования функции StoreRetainsInFile представлен в Листинг 8.2:

```
// Запись retain данных.
BOOL needWriteRetain := TRUE;
res := NftSys.StoreRetainsInFile(needWriteRetain);
```

Листинг 8.2 – Пример использования функции StoreRetainsInFile

8.1.3. Функция CreateHistoricalDataFile

Функция CreateHistoricalDataFile предназначена для создания файла. В данный файл можно будет запись данных из пользовательского массива. Если в ПЛК вставлена SD карта, то файл создастся на ней, иначе в каталоге с проектом. Как только размер файла превысит установленную

границу текущие данные перенесутся в файл с заданным именем и постфиксом Prev, новые данные будут записываться в файл с заданным при вызове функции именем. Данную функцию необходимо вызывать в событии DownloadDone.

Параметры и возвращаемые значения функции CreateHistoricalDataFile приведены в Табл. 8.3.

Таблица 8.3 – Параметры и возвращаемое значение функции CreateHistoricalDataFile

Область	Имя	Тип параметра	Описание
Return	CreateHistoricalDataFile	BOOL	TRUE если операция выполнена успешно
Input	FileName	STRING(20)	Имя файла
Input	FileId	USINT	Уникальный ID файла
Input	FileSizeMb	USINT	Максимальный размер файла в Мегабайтах
Output	ulResult	RTS_IEC_RESULT	Код выполнения функции

Пример использования функции CreateHistoricalDataFile представлен в Листинг 8.3:

```
// Запись retain данных.
VAR_OUTPUT
Result: RTS_IEC_RESULT;
END_VAR
res := NftSys.CreateHistoricalDataFile('TestFile', 1, 1, ulResult=>Result);
```

Листинг 8.3 – Пример использования функции CreateHistoricalFile

8.1.4. Функция StoreHistoricalFile

Функция StoreHistoricalFile предназначена для записи данных в файл.

Параметры и возвращаемые значения функции StoreHistoricalFile приведены в Табл. 8.4.

Таблица 8.4 – Параметры и возвращаемое значение функции StoreHistoricalFile

Область	Имя	Тип параметра	Описание
Return	CreateHistoricalDataFile	BOOL	TRUE если операция выполнена успешно
Input	FileName	STRING(20)	Имя файла
Input	FileId	USINT	Уникальный ID файла
Input	FileSizeMb	USINT	Максимальный размер файла в Мегабайтах
Output	ulResult	RTS_IEC_RESULT	Код выполнения функции

Пример использования функции StoreHistoricalFile представлен в Листинг 8.4:

```
// Чтение из файла в массив REAL.
VAR
TestArray: ARRAY [1..3000] OF REAL;
END_VAR
VAR_OUTPUT
Result: RTS_IEC_RESULT;
END_VAR
res := NftSys.ReadReal2DimArray('TestArrayFile.ret', ADR(TestArray), SIZEOF(TestArray) ulResult
=> Result);
```

Листинг 8.4 – Пример использования функции StoreHistoricalFile

8.1.5. Функция ReadRealFromFile

Функция ReadRealFromFile предназначена для чтения данных из файла в массив REAL. Файл должен находиться в каталоге Retain в FTP.

Параметры и возвращаемые значения функции ReadRealFromFile приведены в Табл. 8.5.

Таблица 8.5 – Параметры и возвращаемое значение функции ReadRealFromFile

Область	Имя	Тип параметра	Описание
Return	ReadRealFromFile	BOOL	TRUE - если операция выполнена успешно
Input	FileName	STRING(20)	TRUE - если операция выполнена успешно
Input	Data	POINTER TO REAL	Указатель на данные
Input	DataSize	UINT	Размер данных
Output	ulResult	RTS_IEC_RESULT	Код выполнения функции(0 - успешно, 1 - некорректное имя файла, например, содержит '/', 2 - не удаётся открыть файл или его не существует, 4 - ошибка чтения, 5 - не совпадение размера массива и файла)

Пример использования функции ReadRealFromFile представлен в Листинг 8.5:

```
// Чтение из файла в двумерный массив REAL.
VAR
TestArray: ARRAY [1..30, 1..10] OF REAL;
END_VAR
VAR_OUTPUT
Result: RTS_IEC_RESULT;
END_VAR
res := NftSys.ReadReal2DimArray(TestArray, 'TestArrayFile.ret', ulResult => Result);
```

Листинг 8.5 – Пример использования функции ReadRealFromFile

8.1.6. Функция ReadReal2DimArray

Функция ReadReal2DimArray предназначена для чтения данных из файла в двумерный массив REAL. Файл должен находиться в каталоге Retain в FTP.

Параметры и возвращаемые значения функции ReadReal2DimArray приведены в Табл. 8.6.

Таблица 8.6 – Параметры и возвращаемое значение функции ReadReal2DimArray

Область	Имя	Тип параметра	Описание
Return	ReadReal2DimArray	BOOL	TRUE если операция выполнена успешно
Input	DataArray	ARRAY [*,*] OF REAL	Массив с данными
Input	FileName	STRING(20)	Имя файла
Output	ulResult	RTS_IEC_RESULT	Код выполнения функции(0 - успешно, 1 - некорректное имя файла, например, содержит '/', 2 - не удаётся открыть файл или его не существует, 4 - ошибка чтения, 5 - не совпадение размера массива и файла)

Пример использования функции ReadReal2DimArray представлен в Листинг 8.6:

```
// Чтение из файла в двумерный массив REAL.
VAR
TestArray: ARRAY [1..30, 1..10] OF REAL;
END_VAR
VAR_OUTPUT
Result: RTS_IEC_RESULT;
END_VAR
res := NftSys.ReadReal2DimArray(TestArray, 'TestArrayFile.ret', ulResult => Result);
```

Листинг 8.6 – Пример использования функции ReadReal2DimArray

8.1.7. Функция WriteRealToFile

Функция WriteRealToFile предназначена для записи REAL данных в файл. Файл должен находиться в каталоге Retain в FTP, если файла нет, то он будет создан.

Параметры и возвращаемые значения функции WriteRealToFile приведены в Табл. 8.7.

Таблица 8.7 – Параметры и возвращаемое значение функции WriteRealToFile

Область	Имя	Тип параметра	Описание
Return	WriteRealToFile	BOOL	TRUE если операция выполнена успешно

Область	Имя	Тип параметра	Описание
Input	FileName	STRING(20)	Имя файла
Input	Data	POINTER TO REAL	Указатель на данные
Input	DataSize	UINT	Размер данных
Output	ulResult	RTS_IEC_RESULT	Код выполнения функции(0 - успешно, 1 - некорректное имя файла, например, содержит '/', 2 - не удаётся открыть файл или его не существует, 4 - ошибка чтения, 5 - не совпадение размера массива и файла)

Пример использования функции WriteRealToFile представлен в Листинг 8.7:

```
// Запись из массива в файл.
VAR
TestArray: ARRAY [1..3000] OF REAL;
END_VAR
VAR_OUTPUT
Result: RTS_IEC_RESULT;
END_VAR
res := NftSys.WriteRealToFile(TestArray, 'TestArrayFile.ret', ulResult => Result);
```

Листинг 8.7 – Пример использования функции WriteRealToFile

8.1.8. Функция WriteReal2DimArray

Функция WriteReal2DimArray предназначена для записи данных из двумерного массива REAL в файл. Файл должен находиться в каталоге Retain в FTP, если файла нет, то он будет создан.

Параметры и возвращаемые значения функции WriteReal2DimArray приведены в Табл. 8.8.

Таблица 8.8 – Параметры и возвращаемое значение функции WriteReal2DimArray

Область	Имя	Тип параметра	Описание
Return	WriteReal2DimArray	BOOL	TRUE если операция выполнена успешно
Input	DataArray	ARRAY [*,*] OF REAL	Массив с данными
Input	FileName	STRING(20)	Имя файла
Output	ulResult	RTS_IEC_RESULT	Код выполнения функции(0 - успешно, 1 - некорректное имя файла, например, содержит '/', 2 - не удаётся открыть файл или его не существует, 4 - ошибка чтения, 5 - не совпадение размера массива и файла)

Пример использования функции WriteReal2DimArray представлен в Листинг 8.8:

```
// Запись из двумерного массива в файл.
VAR
TestArray: ARRAY [1..30, 1..10] OF REAL;
END_VAR
VAR_OUTPUT
Result: RTS_IEC_RESULT;
END_VAR
res := NftSys.WriteReal2DimArray(TestArray, 'TestArrayFile.ret', ulResult => Result);
```

Листинг 8.8 – Пример использования функции WriteReal2DimArray

8.2. Библиотека NftSpecial

Библиотека предназначена для предоставления дополнительных возможностей отладки программы пользователя. Библиотека представляет следующие функции: UdpMessage.

8.2.1. Функция UdpMessage

Функция UdpMessage выполняет отправку отладочных сообщений в специальную программу-приёмник сообщений udp_debug (на 06.12.2022 актуальная версия 1.0.3) из текста программы пользователя. Метка времени CPU автоматически добавляется к тексту сообщения. Его можно

бесплатно скачать с сайта разработчиков АО “Нефтеавтоматика”.

Параметры и возвращаемые значения функции UdpMessage приведены в Табл. 8.9.

Таблица 8.9 – Параметры и возвращаемое значение функции UdpMessage

Область	Имя	Тип параметра	Описание
Return	UdpMessage	BOOL	Результат выполнения функции. TRUE - удалось отправить сообщение, FALSE - не удалось отправить сообщение (некорректный UDP-адрес, либо ошибка функции отправки сообщения)
Input	UdpMessage	STRING	UDP-адрес компьютера, на котором запущена программа udp_debug, принимающая сообщения
Input	message	STRING	Текст сообщения

Пример использования функции UdpMessage представлен в Листинг 8.9:

```

VAR
    res  : BOOL;
    var  : USINT;
END_VAR

...

// отправка текстового сообщения в udp_debug
res := NftSpcl.UdpMessage('10.155.27.1', 'error');

// отправка значения переменной var в udp_debug
res := NftSpcl.UdpMessage('10.155.27.1', ANY_TO_STRING(var));

IF NOT res THEN
    ...
END_IF

```

Листинг 8.9 – Пример использования функции UdpMessage

8.3. Библиотека DiagMK-500

8.3.1. Функциональный блок для диагностики модулей без горячего резервирования ПЛК

Назначение и область применения

Блок getDiagNoRedu предназначен для получения данных о состоянии модулей серии MKLogic-500 в явном, структурированном, удобном для передачи по протоколу верхнего уровня виде. Блок применим при построении сети среднего уровня на протоколах Powerlink или CAN и на не резервируемой сборке ПЛК.

Входная информация блока

В блоке предусмотрены входы, описанные в таблице 8.10.

Таблица 8.10 – Описание входной информации

№	Наименование переменной	Тип переменной	Описание переменной
1.	imitOn	BOOL	Флаг наличия симуляции
2.	msBlk	UINT	Количество миллисекунд, прошедших с предыдущего скана программы
3.	rsTimeOut	UINT	Таймаут на потерю связи с портом интерфейсного модуля. Используется если не указана ссылка на таймаут в структуре typeCfgPortRS (описание структуры представлено в таблице 8.51)
4.	timeNotSyncTimeOut	UINT	Таймаут на отсутствие синхронизации времени

Выходная информация блока

В таблице 8.11 описаны выходные переменные блока.

Таблица 8.11 – Описание выходной информации

№	Наименование переменной	Тип переменной	Описание переменной
1.	pwlRing	BOOL	Powerlink в режиме Ring

Входные/выходные переменные блока

В таблице 8.12 описаны переменные блока с направлением вход/выход.

Таблица 8.12 – Описание входных/выходных переменных блока

№	Наименование переменной	Тип переменной	Описание переменной
1.	cmdDiag	WORD	Диагностические команды (бит 0..7 - номер корзины, бит 8..12 - номер модуля (начиная с 0), бит 13..15 - команда: 1 - квитировать неисправность модуля)
2.	healthAI	ARRAY[*] OF BOOL	"Здоровье" модулей AI, где индекс массива – сквозной номер модуля AI.
3.	healthAO	ARRAY[*] OF BOOL	"Здоровье" модулей АО, где индекс массива – сквозной номер модуля АО
4.	healthDI	ARRAY[*] OF BOOL	"Здоровье" модулей DI, где индекс массива – сквозной номер модуля DI
5.	healthDO	ARRAY[*] OF BOOL	"Здоровье" модулей DO, где индекс массива – сквозной номер модуля DO
6.	cfgPSU	ARRAY [*] OF typeCfgMod	Настройки модулей PSU (описание структуры представлено в таблице 8.13)
7.	cfgCPU	ARRAY [*] OF typeCfgCPU	Настройки модулей CPU (описание структуры представлено в таблице 8.14)
8.	cfgCN	ARRAY [*] OF typeCfgMod	Настройки модулей CN (описание структуры представлено в таблице 8.13)
9.	cfgAI	ARRAY [*] OF typeCfgMod	Настройки модулей AI (описание структуры представлено в таблице 8.13)
10.	cfgAO	ARRAY [*] OF typeCfgMod	Настройки модулей АО (описание структуры представлено в таблице 8.13)
11.	cfgDI	ARRAY [*] OF typeCfgMod	Настройки модулей DI (описание структуры представлено в таблице 8.13)
12.	cfgDO	ARRAY [*] OF typeCfgMod	Настройки модулей DO (описание структуры представлено в таблице 8.13)
13.	cfgRS	ARRAY [*] OF typeCfgMod	Настройки модулей RS (описание структуры представлено в таблице 8.13)
14.	mkAI	ARRAY[*,*] OF UINT	Значения в мКА с модулей AI, где первый индекс массива – номер модуля, второй – номер канала
15.	mBUS	ARRAY[*] OF DWORD	Состояние модулей, где индекс массива - номер корзины, бит N = 1 - модуль N + 1 исправен
16.	mBUSandCh	ARRAY[*] OF DWORD	Состояние модулей и каналов, где индекс массива - номер корзины, бит N=1 - модуль N + 1 исправен и его каналы исправны
17.	mBUSblink	ARRAY[*] OF DWORD	Неквитированные неисправности, где индекс массива - номер корзины, бит N=1 - модуль N + 1 имеет неквитированную неисправность
18.	statePSU	ARRAY[*] OF typePSU_HMI	Структура состояния модуля PSU (описание структуры представлено в таблице 8.17)
19.	stateCPU	ARRAY[*] OF typeCPU_HMI	Структура состояния модуля CPU (описание структуры представлено в таблице 8.20)
20.	stateCN	ARRAY[*] OF typeCN_HMI	Структура состояния модуля CN (описание структуры представлено в таблице 8.23)
21.	stateMN	ARRAY[*] OF typeMN_HMI	Структура состояния модуля MN (описание структуры представлено в таблице 8.26)
22.	stateAI8	ARRAY[*] OF typeAI8_HMI	Структура состояния 8-ми канального AI (описание структуры представлено в таблице 8.27)
23.	stateAI16	ARRAY[*] OF typeAI16_HMI	Структура состояния 16-ти канального AI (описание структуры представлено в таблице 8.28)

№	Наименование переменной	Тип переменной	Описание переменной
24.	stateAI8hart	ARRAY [*] OF typeAI8hart_HMI	Структура состояния 8-ми канального AI с поддержкой протокола HART (описание структуры представлено в таблице 8.29)
25.	stateAI16hart	ARRAY [*] OF typeAI16hart_HMI	Структура состояния 16-ти канального AI с поддержкой протокола HART (описание структуры представлено в таблице 8.33)
26.	stateAO	ARRAY[*] OF typeAO_HMI	Структура состояния модуля AO (описание структуры представлено в таблице 8.34)
27.	stateAOhart	ARRAY [*] OF typeAOhart_HMI	Структура состояния модулей AO с поддержкой протокола HART (описание структуры представлено в таблице 8.35)
28.	stateDI	ARRAY[*] OF typeDI_HMI	Структура состояния модуля DI (описание структуры представлено в таблице 8.36)
29.	stateDInamur	ARRAY [*] OF typeDInamur_HMI	Структура состояния модуля DI с Namur (описание структуры представлено в таблице 8.37)
30.	stateDO	ARRAY[*] OF typeDO_HMI	Структура состояния модуля DO (описание структуры представлено в таблице 8.38)
31.	stateRS	ARRAY[*] OF typeRS_HMI	Структура состояния модуля RS (описание структуры представлено в таблице 8.39)
32.	stateEthEx	ARRAY[*] OF typeEthEx_HMI	Структура состояния модуля расширения Ethernet (описание структуры представлено в таблице 8.44)
33.	stateNTP	typeNTP_HMI	Структура состояния NTP сервера (описание структуры представлено в таблице 8.43)
34.	mPSU	ARRAY[*] OF Nftloltfs.MK_550_024_Type	Данные с каналов модуля PSU (тип данных поставляется с пакетом MKLogic-500)
35.	mCPU	ARRAY[*] OF Nftloltfs.MK_505_120_Type	Данные с каналов процессорных модулей (тип данных поставляется с пакетом MKLogic-500)
36.	mEthEx	ARRAY [*] OF ARRAY [1..4] OF Nftloltfs.MK_544_040_Type	Данные с каналов модулей расширения Ethernet (тип данных поставляется с пакетом MKLogic-500)
37.	mMN	ARRAY[*] OF Nftloltfs.MK_546_010_Type	Данные с каналов модулей powerlink MN (тип данных поставляется с пакетом MKLogic-500)
38.	mCN	ARRAY[*] OF typeDeviceCN	Данные с каналов модулей powerlink CN (описание структуры представлено в таблице 8.50)
39.	mAI	ARRAY[*] OF typeDeviceAI	Данные с каналов модулей AI (описание структуры представлено в таблице 8.48)
40.	mAO	ARRAY[*] OF typeDeviceAO	Данные с каналов модулей AO (описание структуры представлено в таблице 8.49)
41.	mDI	ARRAY[*] OF typeDeviceDI	Данные с каналов модулей DI (описание структуры представлено в таблице 8.50)
42.	mDO	ARRAY[*] OF Nftloltfs.MK_531_032_Type	Данные с каналов модулей DO (тип данных поставляется с пакетом MKLogic-500)
43.	mRS	ARRAY[*] OF Nftloltfs.MK_541_002_Type	Данные с каналов модулей RS (тип данных поставляется с пакетом MKLogic-500)
44.	mNTP	NFTSys.NTPDiag	Данные с диагностического канала устройства System (Диагностика NTP сервера) (тип данных поставляется с пакетом MKLogic-500)
45.	portRS	ARRAY[*] OF typeCfgPortRS	Настроечные параметры по портам интерфейсных модулей (описание структуры представлено в таблице 8.51)
46.	internalRS	ARRAY[*] OF typeRSinternal	Внутреннее состояние интерфейсных модулей (описание структуры представлено в таблице 8.52)

Структуры модулей

В таблицах 8.13 - 8.53 представлены описания структур модулей.

Таблица 8.13 – Описание структуры typeCfgMod

№	Наименование переменной	Тип переменной	Описание переменной
1.	typ	Nftloltfs.MKLogic500	Тип модуля (тип данных поставляется с пакетом MKLogic-500)
2.	pos	Nftloltfs.loCommonPositionInfoType	Позиция модуля (тип данных поставляется с пакетом MKLogic-500)
3.	portsEnbl	BYTE	Используемые порты, где 0 бит - Eth1, 1 бит - Eth2

Таблица 8.14 – Описание структуры typeCfgCPU

№	Наименование переменной	Тип переменной	Описание переменной
1.	typ	WORD	Тип модуля (тип данных поставляется с пакетом MKLogic-500)
2.	pos	Nftloltfs.ioCommon PositionInfoType	Позиция модуля (тип данных поставляется с пакетом MKLogic-500)
3.	MN	typeCfgMN	Настройки модуля MN (описание структуры представлено в таблице 8.15)
4.	EthEx	ARRAY [1..4] OF typeCfgEthEx	Настройки модуля расширения Ethernet (описание структуры представлено в таблице 8.16)
5.	portsEnbl	BYTE	Используемые порты, где 0 бит - Eth1, 1 бит - Eth2, 2 бит - SFP

Таблица 8.15 – Описание структуры typeCfgMN

№	Наименование переменной	Тип переменной	Описание переменной
1.	slot	USINT	Номера слота модуля в корзине
2.	portsEnbl	BYTE	Используемые порты, где 0 бит - Eth1, 1 бит - Eth2

Таблица 8.16 – Описание структуры typeCfgEthEx

№	Наименование переменной	Тип переменной	Описание переменной
1.	slot	USINT	Номера слота модуля в корзине
2.	portsEnbl	BYTE	Используемые порты, где 0 бит - Eth1, 1 бит - Eth2, 2 бит - Eth3, 3 бит - Eth4

Таблица 8.17 – Описание структуры typePSU_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mod_State_ext	typePSUModStateExt_union	Расширенное состояние модуля. Объединение битов (typePSUModStateExt таблица 8.19) в регистр
3.	SupplyVoltage	REAL	Входное напряжение (В)
4.	CanBusSpeed	WORD	Скорость CAN-шины (кбит/с)
5.	Reserve	WORD	Резервный регистр

Таблица 8.18 – Описание структуры typeModState

№	Наименование переменной	Тип переменной	Описание переменной
1.	sConOK	BIT	Есть связь с модулем
2.	sNotCompability	BIT	Модуль не совместим с требуемым
3.	sNotPresent	BIT	Модуль не установлен
4.	sNotCAN1	BIT	BusA отсутствует соединение CAN
5.	sNotCAN2	BIT	BusB отсутствует соединение CAN
6.	ledRun	BIT	Индикатор Run - горит
7.	ledErr	BIT	Индикатор Err - горит
8.	blkErr	BIT	Индикатор Err - мигает
9.	ledBusA	BIT	Индикатор BusA - горит

№	Наименование переменной	Тип переменной	Описание переменной
10.	blkBusA	BIT	Индикатор BusA - мигает
11.	ledBusB	BIT	Индикатор BusB - горит
12.	blkBusB	BIT	Индикатор BusB - мигает
13.	eNoLeftMod	BIT	Отсутствует модуль слева
14.	sNotDataReady	BIT	Нет готовности модуля к передаче данных
15.	sNotLedRun	BIT	Не горит индикатор Run

Таблица 8.19 – Описание структуры typePSUModStateExt

№	Наименование переменной	Тип переменной	Описание переменной
1.	sLowVoltage	BIT	Пониженное напряжение
2.	sNoCorrAddress	BIT	Переключатель адреса CAN в запрещенном положении
3.	sNoCorrBtrate	BIT	Переключатель скорости CAN в запрещенном положении
4.	sNotCAN1Ext	BIT	Отсутствует соединение по CAN1 (внешняя шина)
5.	sNotCAN2sNotCAN2Ext	BIT	Отсутствует соединение по CAN2 (внешняя шина)

Таблица 8.20 – Описание структуры typeCPU_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mod_State_ext	typeModStateExt_union	Расширенное состояние модуля. Объединение битов (typeModStateExt таблица 8.21) в регистр
3.	modStateErr	typeModStateErr_union	Ошибки на модуле. Объединение битов (typeModStateErr таблица 8.22) в регистр
4.	CPUMemFree	WORD	Свободная память (МБ)
5.	CPUload	REAL	Загрузка CPU (%)
6.	CRC32	DWORD	CRC32 файлов ресурса в ПЛК
7.	DiskFreeSpace	WORD	Свободное место на диске (МБ)
8.	Reserve	WORD	Резервный регистр

Таблица 8.21 – Описание структуры typeModStateExt

№	Наименование переменной	Тип переменной	Описание переменной
1.	sPrimary	BIT	В режиме основной
2.	sStandby	BIT	В режиме резервный
3.	ledFOErr	BIT	Индикатор FOErr - горит
4.	blkFOErr	BIT	Индикатор FOErr - мигает
5.	ledPrim	BIT	Индикатор Prim – горит
6.	blkPrim	BIT	Индикатор Prim - мигает
7.	ledAct	BIT	Индикатор Act - горит
8.	blkAct	BIT	Индикатор Act - мигает
9.	isSync	BIT	Время синхронизировано (с выдержкой времени на отсутствие синхронизации времени)
10.	isSyncCh	BIT	Время синхронизировано (без выдержки времени на отсутствие синхронизации времени)
11.	notSync	BIT	Время не синхронизировано (с выдержкой времени на отсутствие синхронизации времени)

Таблица 8.22 – Описание структуры typeModStateErr

№	Наименование переменной	Тип переменной	Описание переменной
1.	eFOErr	BIT	Наличие ошибок резервирования (с выдержкой времени на отсутствие синхронизации данных)
2.	eSFPNoLink	BIT	Порт SFP. Нет связи
3.	eP1NoLink	BIT	Порт 1 Ethernet. Нет связи
4.	eP2NoLink	BIT	Порт 2 Ethernet. Нет связи
5.	eSFPErr	BIT	SFP-модуль неисправен или отсутствует
6.	eFOErrCh	BIT	Наличие ошибок резервирования (без выдержки времени на отсутствие синхронизации данных)
7.	eCPUload	BIT	Загрузка процессора выше нормы
8.	eNTPDstop	BIT	Остановлен сервис ntpd

Таблица 8.23 – Описание структуры typeCN_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mod_State_ext	typeCNModStateExt_union	Расширенное состояние модуля. Объединение битов (typeCNModStateExt таблица 8.24) в регистр
3.	ports_State	typeCNPortsState_union	Состояние портов. Объединение битов (typeCNPortsState таблица 8.25) в регистр
4.	pwl_id	WORD	Powerlink ID
5.	badFrameCounter	WORD	Счетчик плохих фреймов в сети
6.	Reserve	WORD	Резервный регистр

Таблица 8.24 – Описание структуры typeCNModStateExt

№	Наименование переменной	Тип переменной	Описание переменной
1.	ledLine	BIT	Светодиод Line включен, Powerlink в режиме Line
2.	ledRing	BIT	Светодиод Ring включен, Powerlink в режиме Ring
3.	ledEth	BIT	Светодиод Eth включен, порт в режиме Ethernet
4.	ledPwl	BIT	Светодиод Pwl включен, порт в режиме Powerlink
5.	sP1Ok	BIT	Порт 1. В норме
6.	eP1FDplx	BIT	Порт 1. Режим Full Duplex
7.	eP1n10MB	BIT	Порт 1. Режим 10 Мбит
8.	eP1n100MB	BIT	Порт 1. Режим 100 Мбит
9.	eP1n1000MB	BIT	Порт 1. Режим 1000 Мбит
10.	sP2Ok	BIT	Порт 2. В норме
11.	eP2FDplx	BIT	Порт 2. Режим Full Duplex
12.	eP2n10MB	BIT	Порт 2. Режим 10 Мбит
13.	eP2n100MB	BIT	Порт 2. Режим 100 Мбит
14.	eP2n1000MB	BIT	Порт 2. Режим 1000 Мбит
15.	badFrame	BIT	Наличие плохого фрейма в сети

Таблица 8.25 – Описание структуры typeCNPortsState

№	Наименование переменной	Тип переменной	Описание переменной
1.	eP1NotLink	BIT	Порт 1. Нет связи
2.	eP2NotLink	BIT	Порт 2. Нет связи

Таблица 8.26 – Описание структуры typeMN_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State_ext	typeCNModStateExt_union	Расширенное состояние модуля. Объединение битов (typeCNModStateExt таблица 8.24) в регистр
2.	ports_State	typeCNPortsState_union	Состояние портов. Объединение битов (typeCNPortsState таблица 8.25) в регистр

Таблица 8.27 – Описание структуры typeAI8_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mAI	ARRAY[1..8] OF UINT	Значения каналов аналогового ввода в кодах АЦП

Таблица 8.28 – Описание структуры typeAI16_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mAI	ARRAY[1..16] OF UINT	Значения каналов аналогового ввода в кодах АЦП

Таблица 8.29 – Описание структуры typeAI8hart_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mAI	ARRAY[1..8] OF UINT	Значения каналов аналогового ввода в кодах АЦП
3.	mHart	ARRAY[1..8] OF typeHartCh	Диагностика HART по каждому каналу (описание структуры представлено в таблице 8.30)

Таблица 8.30 – Описание структуры typeHartCh

№	Наименование переменной	Тип переменной	Описание переменной
1.	hartSelected	WORD	Количество используемых HART переменных
2.	sensorID	WORD	ID датчика
3.	commErr	typeCommErr_union	Коммуникационные ошибки. Объединение битов (typeCommErr таблица 8.31) в регистр
4.	cmdStatus	typeCmdStatus_union	Состояние обработки команды. Объединение битов (typeCmdStatus таблица 8.32) в DWORD
5.	hartValues	ARRAY [1..4] OF REAL	HART значения

Таблица 8.31 – Описание структуры typeCommErr

№	Наименование переменной	Тип переменной	Описание переменной
1.	responseError	BIT	Нет ответа от устройства
2.	configError	BIT	Неверная конфигурация канала
3.	hartFailure	BIT	Отказ канала
4.	parityErr	BIT	Ошибка четности
5.	speedOverflow	BIT	Перегрузка по скорости
6.	syncErr	BIT	Ошибка синхронизации или формата посылки
7.	crcErr	BIT	Ошибка контрольной суммы
8.	bufOverflow	BIT	Переполнение буфера приемника

Таблица 8.32 – Описание структуры typeCmdStatus

№	Наименование переменной	Тип переменной	Описание переменной
1.	noCmd	BIT	Нет команды
2.	wrongChoose	BIT	Неверный адрес для опроса
3.	prmTooBig	BIT	Последний принятый параметр слишком велик
4.	prmTooSmall	BIT	Последний принятый параметр слишком мал
5.	fewData	BIT	Получено недостаточное количество байт
6.	writeProtect	BIT	Выставлен режим защиты от записи
7.	accessDenied	BIT	Ограничен доступ
8.	dataNotUpdates	BIT	Не обновляются данные
9.	deviceBusy	BIT	Устройство занято
10.	cmdNotImpl	BIT	Команда не реализована
11.	deviceFailure	BIT	Устройство неисправно
12.	configChanged	BIT	Изменена конфигурация устройства
13.	coldStart	BIT	Выполняется «холодный старт» устройства
14.	moreStatusAvailable	BIT	Доступна дополнительная информация о статусе
15.	fixedAO	BIT	Значение аналогового выхода фиксировано
16.	saturatedAO	BIT	Аналоговый выход в насыщении
17.	primOutOfLimits	BIT	Значение главной переменной вне установленных пределов
18.	nonPrimOutOfLimits	BIT	Значение неглавной переменной вне установленных пределов

Таблица 8.33 – Описание структуры typeAI16hart_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mAI	ARRAY[1..16] OF UINT	Значения каналов аналогового ввода в кодах АЦП
3.	mHart	ARRAY[1..16] OF typeHartCh	Диагностика HART по каждому каналу (описание структуры представлено в таблице 8.30)

Таблица 8.34 – Описание структуры typeAO_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	chHealth	WORD	"Здоровье" каналов, где бит N = 1 – канал N + 1 исправен
3.	mAO	ARRAY[1..8] OF UINT	Значения каналов аналогового вывода в кодах ЦАП

Таблица 8.35 – Описание структуры typeAOhart_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	chHealth	WORD	"Здоровье" каналов, где бит N = 1 – канал N исправен
3.	mAO	ARRAY[1..8] OF UINT	Значения каналов аналогового вывода в кодах ЦАП
4.	mHart	ARRAY[1..8] OF typeHartCh	Диагностика HART по каждому каналу (описание структуры представлено в таблице 8.30)

Таблица 8.36 – Описание структуры typeDI_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	Reserve	WORD	Резервный регистр
3.	mDI	DWORD	Значения каналов дискретного ввода, где бит N=1 – на канале N + 1 установлено значение 1.

Таблица 8.37 – Описание структуры typeDInamur_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	Reserve	WORD	Резервный регистр
3.	mDI	DWORD	Значения каналов дискретного ввода, где бит N=1 – на канале N + 1 установлено значение 1.
4.	chHealth	DWORD	"Здоровье" каналов, где бит N = 1 - канал N + 1 исправен

Таблица 8.38 – Описание структуры typeDO_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	Reserve	WORD	Резервный регистр
3.	mDI	DWORD	Значения каналов дискретного ввода, где бит N=1 – на канале N + 1 установлено значение 1.

Таблица 8.39 – Описание структуры typeRS_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeModState_union	Общее состояние модуля. Объединение битов (typeModState таблица 8.18) в регистр
2.	mod_State_ext	typeRSModStateExt_union	Расширенное состояние модуля. Объединение битов (typeRSModStateExt таблица 8.40) в регистр

Таблица 8.40 – Описание структуры typeRSModStateExt

№	Наименование переменной	Тип переменной	Описание переменной
1.	sP1Link	BIT	Порт 1 - Идет обмен
2.	eP1ErrReq	BIT	Порт 1 - Есть ошибки обмена
3.	eP1ErrAll	BIT	Порт 1 - Нет связи
4.	sP2Link	BIT	Порт 2 - Идет обмен
5.	eP2ErrReq	BIT	Порт 2 - Есть ошибки обмена
6.	eP2ErrAll	BIT	Порт 2 - Нет связи
7.	sP1Slave	BIT	Порт 1 - В режиме Slave
8.	sP2Slave	BIT	Порт 2 - В режиме Slave

Таблица 8.41 – Описание структуры typeEthEx_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	mod_State	typeEthExModState_union	Состояние портов. Объединение битов (typeEthExModState таблица 8.42) в регистр

Таблица 8.42 – Описание структуры typeEthExModState

№	Наименование переменной	Тип переменной	Описание переменной
1.	eP1NotLink	BIT	Порт 1. Нет связи
2.	eP2NotLink	BIT	Порт 2. Нет связи
3.	eP3NotLink	BIT	Порт 3. Нет связи
4.	eP4NotLink	BIT	Порт 4. Нет связи

Таблица 8.43 – Описание структуры typeNTP_HMI

№	Наименование переменной	Тип переменной	Описание переменной
1.	NTPState	WORD	Состояние сервиса NTP (1 – включен, 0 – выключен)
2.	LeapCode	typeLeapCode_union	Состояние приращения. Объединение битов (typeLeapCode таблица 8.44) в регистр
3.	SyncCode	typeSyncCode_union	Источник синхронизации. Объединение битов (typeSyncCode таблица 8.45) в регистр
4.	EventState	typeEventState_union	Последнее системное событие. Объединение битов (typeEventState таблица 0) в регистр
5.	ServerIP	ARRAY[1..4] OF WORD	IP адрес сервера, где каждый элемент массива – октет IP-адреса
6.	ServerStratum	WORD	Стратум сервера
7.	LocalStratum	WORD	Локальный стратум
8.	ServerWhen	WORD	Время с момента последнего ответа
9.	ServerPoll	WORD	Интервал опроса

Таблица 8.44 – Описание структуры typeLeapCode

№	Наименование переменной	Тип переменной	Описание переменной
1.	NoWarn	BIT	Предупреждения нет
2.	LastMinDay61	BIT	Последняя минута содержит 61 секунду
3.	LastMinDay59	BIT	Последняя минута содержит 59 секунду
4.	Overload	BIT	Аварийный сигнал (часы не синхронизованы)

Таблица 8.45 – Описание структуры typeSyncCode

№	Наименование переменной	Тип переменной	Описание переменной
1.	Unspec	BIT	Источник не специфицирован или неизвестен
2.	PPS	BIT	Калиброванные атомные часы
3.	LFradio	BIT	Низкочастотный радиопередатчик
4.	HFRadio	BIT	Высокочастотный радиопередатчик
5.	UHFradio	BIT	УВЧ спутник
6.	Local	BIT	Локальная сеть
7.	NTP	BIT	UDP/NTP
8.	Other	BIT	UDP/time
9.	Wristwatch	BIT	Wristwatch
10.	Telephone	BIT	Телефонный модем

Таблица 8.46 – Описание структуры typeEventState

№	Наименование переменной	Тип переменной	Описание переменной
1.	Unspec	BIT	Не специфицировано
2.	Restart	BIT	Рестарт системы
3.	HWfault	BIT	Системный или аппаратный сбой
4.	SyncChange	BIT	Новое статусное слово системы
5.	NewSource	BIT	Новый источник синхронизации
6.	ClockReset	BIT	Сброс системных часов
7.	InvalidTS	BIT	Некорректное системное время или дата
8.	SysClockExcp	BIT	Исключение системных часов

Таблица 8.47 – Описание структуры typeDeviceCN

№	Наименование переменной	Тип переменной	Описание переменной
1.	pwL_ID	USINT(1..239)	Powerlink CN ID
2.	cfgIO_Ok	BOOL	Совпадение IO конфигурации
3.	module	Nftloltfs.MK_545_010_Type	Диагностика модуля (тип данных поставляется с пакетом MKLogic-500)

Таблица 8.48 – Описание структуры typeDeviceAI

№	Наименование переменной	Тип переменной	Описание переменной
1.	x8	Nftloltfs.MK_516_008A_Type	8-ми каналный модуль AI (тип данных поставляется с пакетом MKLogic-500)
2.	x8_Hart	Nftloltfs.MK_576_008A_Type	16-ти каналный модуль AI (тип данных поставляется с пакетом MKLogic-500)
3.	x16	Nftloltfs.MK_513_016A_Type	8-ми каналный модуль AI с поддержкой HART (тип данных поставляется с пакетом MKLogic-500)
4.	x16_Hart	Nftloltfs.MK_576_016A_Type	16-ти каналный модуль AI с поддержкой HART (тип данных поставляется с пакетом MKLogic-500)

Таблица 8.49 – Описание структуры typeDeviceAO

№	Наименование переменной	Тип переменной	Описание переменной
1.	x8	Nftloltfs.MK_514_008A_Type	8-ми каналный модуль AO (тип данных поставляется с пакетом MKLogic-500)
2.	x8_Hart	Nftloltfs.MK_574_008A_Type	8-ми каналный модуль AO с поддержкой Hart (тип данных поставляется с пакетом MKLogic-500)

Таблица 8.50 – Описание структуры typeDeviceDI

№	Наименование переменной	Тип переменной	Описание переменной
1.	x32	Nftloltfs.MK_521_032_Type	32-х каналный модуль DI (тип данных поставляется с пакетом MKLogic-500)
2.	x32_Namur	Nftloltfs.MK_523_032_Type	32-х каналный модуль DI Namur (тип данных поставляется с пакетом MKLogic-500)

Таблица 8.51 – Описание структуры typeCfgPortRS

№	Наименование переменной	Тип переменной	Описание переменной
1.	pCounter1	REFERENCE TO WORD	Ссылка на счетчик связи (для порта 1 в режиме Slave)
2.	pCounter2	REFERENCE TO WORD	Ссылка на счетчик связи (для порта 2 в режиме Slave)
3.	pTimeOut1	REFERENCE TO WORD	Ссылка на таймаут на потерю связи (сек.) (порт 1)

№	Наименование переменной	Тип переменной	Описание переменной
4.	pTimeOut2	REFERENCE TO WORD	Ссылка на таймаут на потерю связи (сек.) (порт 2)

Таблица 8.52 – Описание структуры typeRSinternal

№	Наименование переменной	Тип переменной	Описание переменной
1.	counterOnScan	ARRAY[1..2, 1..2] OF WORD	Счетчик на 2-х сканах по каждому каналу
2.	twTimer	ARRAY[1..2] OF typeTimerDiag	Таймер (описание структуры представлено в таблице 8.56)

Таблица 8.53 – Описание структуры typeTimerDiag

№	Наименование переменной	Тип переменной	Описание переменной
1.	State	WORD	Состояние таймера
2.	Ust	UINT	Уставка
3.	Q	BOOL	Таймер отсчитал
4.	Start	BOOL	Команда запуска
5.	Stop	BOOL	Команда остановки
6.	IsPause	BOOL	Пауза
7.	ET	UDINT	Прошедшее время

8.3.2. Функциональный блок для диагностики модулей с горячим резервированием ПЛК

Назначение и область применения

Блок getDiagRedu предназначен для получения данных о состоянии модулей серии MKLogic-500 в явном, структурированном, удобном для передачи по протоколу верхнего уровня виде. Блок применим при построении сети среднего уровня на протоколах Powerlink или CAN и на резервируемой сборке ПЛК.

Входная информация блока

В блоке предусмотрены входы, описанные в таблице 8.54.

Таблица 8.54 – Описание входной информации

№	Наименование переменной	Тип переменной	Описание переменной
1.	imitOn	BOOL	Флаг наличия симуляции
2.	redundancyDiag	typeDiagRedundancy	Диагностика системы резервирования ПЛК (данные с устройства Redundancy) (описание структуры представлено в таблице 8.57)
3.	msBlk	UINT	Количество миллисекунд, прошедших с предыдущего скана программы
4.	rsTimeOut	UINT	Таймаут на потерю связи с портом интерфейсного модуля. Используется если не указана ссылка на таймаут в структуре typeCfgPortRS (описание структуры представлено в таблице 8.51)
5.	timeNotSyncTimeOut	UINT	Таймаут на отсутствие синхронизации времени
6.	plcNotSyncTimeOut	UINT	Таймаут на потерю связи с резервным ПЛК

Выходная информация блока

В таблице 8.55 описаны выходные переменные блока.

Таблица 8.55 – Описание выходной информации

№	Наименование переменной	Тип переменной	Описание переменной
1.	activePLCNum	UINT	Индекс активного ПЛК

№	Наименование переменной	Тип переменной	Описание переменной
2.	pwlRing	BOOL	Powerlink в режиме Ring

Входные/выходные переменные блока

В таблице 8.56 описаны переменные блока с направлением вход/выход.

Таблица 8.56 – Описание входных/выходных переменных блока

№	Наименование переменной	Тип переменной	Описание переменной
1.	cmdDiag	WORD	Диагностические команды (бит 0..7 - номер корзины, бит 8..12 - номер модуля (начиная с 0), бит 13..15 - команда: 1 - квитиловать неисправность модуля)
2.	healthAI	ARRAY [*] OF BOOL	"Здоровье" модулей AI, где индекс массива – сквозной номер модуля AI
3.	healthAO	ARRAY [*] OF BOOL	"Здоровье" модулей AO, где индекс массива – сквозной номер модуля AO
4.	healthDI	ARRAY [*] OF BOOL	"Здоровье" модулей DI, где индекс массива – сквозной номер модуля DI
5.	healthDO	ARRAY [*] OF BOOL	"Здоровье" модулей DO, где индекс массива – сквозной номер модуля DO
6.	cfgPSU	ARRAY [*] OF typeCfgMod	Настройки модулей PSU (описание структуры представлено в таблице 8.13)
7.	cfgCPU	ARRAY [*] OF typeCfgCPU	Настройки модулей CPU (описание структуры представлено в таблице 8.14)
8.	cfgCN	ARRAY [*] OF typeCfgMod	Настройки модулей CN (описание структуры представлено в таблице 8.13)
9.	cfgAI	ARRAY [*] OF typeCfgMod	Настройки модулей AI (описание структуры представлено в таблице 8.13)
10.	cfgAO	ARRAY [*] OF typeCfgMod	Настройки модулей AO (описание структуры представлено в таблице 8.13)
11.	cfgDI	ARRAY [*] OF typeCfgMod	Настройки модулей DI (описание структуры представлено в таблице 8.13)
12.	cfgDO	ARRAY [*] OF typeCfgMod	Настройки модулей DO (описание структуры представлено в таблице 8.13)
13.	cfgRS	ARRAY [*] OF typeCfgMod	Настройки модулей RS (описание структуры представлено в таблице 8.13)
14.	mkAI	ARRAY [*,*] OF UINT	Значения в мКА с модулей AI, где первый индекс массива – номер модуля, второй – номер канала
15.	mBUS	ARRAY [*] OF DWORD	Состояние модулей, где индекс массива - номер корзины, бит N = 1 - модуль N + 1 исправен
16.	mBUSandCh	ARRAY [*] OF DWORD	Состояние модулей и каналов, где индекс массива - номер корзины, бит N=1 - модуль N + 1 исправен и его каналы исправны
17.	mBUSblink	ARRAY [*] OF DWORD	Неквитилованные неисправности, где индекс массива - номер корзины, бит N=1 - модуль N + 1 имеет неквитилованную неисправность
18.	statePSU	ARRAY [*] OF typePSU_HMI	Структура состояния модуля PSU (описание структуры представлено в таблице 8.17)
19.	stateCPU	ARRAY [*] OF typePSU_HMI	Структура состояния модуля CPU (описание структуры представлено в таблице 8.20)
20.	stateCN	ARRAY [*] OF typePSU_HMI	Структура состояния модуля CN (описание структуры представлено в таблице 8.23)
21.	stateMN	ARRAY [*] OF typePSU_HMI	Структура состояния модуля MN (описание структуры представлено в таблице 8.26)
22.	stateAI8	ARRAY [*] OF typeAI8_HMI	Структура состояния 8-ми канального AI (описание структуры представлено в таблице 8.27)
23.	stateAI16	ARRAY [*] OF typeAI16_HMI	Структура состояния 16-ти канального AI (описание структуры представлено в таблице 8.28)
24.	stateAI8hart	ARRAY [*] OF typeAI8hart_HMI	Структура состояния 8-ми канального AI с поддержкой протокола HART (описание структуры представлено в таблице 8.29)
25.	stateAI16hart	ARRAY [*] OF typeAI16hart_HMI	Структура состояния 16-ти канального AI с поддержкой протокола HART (описание структуры представлено в

№	Наименование переменной	Тип переменной	Описание переменной
			таблице 8.33)
26.	stateAO	ARRAY [*] OF typeAO_HMI	Структура состояния модуля AO (описание структуры представлено в таблице 8.34)
27.	stateAOhart	ARRAY [*] OF typeAOhart_HMI	Структура состояния модулей AO с поддержкой протокола HART (описание структуры представлено в таблице 8.35)
28.	stateDI	ARRAY [*] OF typeDI_HMI	Структура состояния модуля DI (описание структуры представлено в таблице 8.36)
29.	stateDInamur	ARRAY [*] OF typeDInamur_HMI	Структура состояния модуля DI с Namur (описание структуры представлено в таблице 8.37)
30.	stateDO	ARRAY [*] OF typeDO_HMI	Структура состояния модуля DO (описание структуры представлено в таблице 8.38)
31.	stateRS	ARRAY [*] OF typeRS_HMI	Структура состояния модуля RS (описание структуры представлено в таблице 8.39)
32.	stateEthEx	ARRAY [*] OF typeEthEx_HMI	Структура состояния модуля расширения Ethernet (описание структуры представлено в таблице 8.41)
33.	stateNTP	typeNTP_HMI	Структура состояния NTP сервера (описание структуры представлено в таблице 8.43)
34.	mPSU	ARRAY [*] OF Nftloltfs.MK_550_024_Type	Данные с каналов модуля PSU (тип данных поставляется с пакетом MKLogic-500)
35.	mCPU	ARRAY [*] OF Nftloltfs.MK_505_120_Type	Данные с каналов процессорных модулей (тип данных поставляется с пакетом MKLogic-500)
36.	mEthEx	ARRAY [*] OF ARRAY [1..4] OF Nftloltfs.MK_544_040_Type	Данные с каналов модулей расширения Ethernet (тип данных поставляется с пакетом MKLogic-500)
37.	mMN	ARRAY [*] OF Nftloltfs.MK_546_010_Type	Данные с каналов модулей powerlink MN (тип данных поставляется с пакетом MKLogic-500)
38.	mCN	ARRAY [*] OF typeDeviceCN	Данные с каналов модулей powerlink CN (описание структуры представлено в таблице 8.47)
39.	mAI	ARRAY [*] OF typeDeviceAI	Данные с каналов модулей AI (описание структуры представлено в таблице 8.58)
40.	mAO	ARRAY [*] OF typeDeviceAO	Данные с каналов модулей AO (описание структуры представлено в таблице 8.49)
41.	mDI	ARRAY [*] OF typeDeviceDI	Данные с каналов модулей DI (описание структуры представлено в таблице 8.50)
42.	mDO	ARRAY [*] OF Nftloltfs.MK_531_032_Type	Данные с каналов модулей DO (тип данных поставляется с пакетом MKLogic-500)
43.	mRS	ARRAY [*] OF Nftloltfs.MK_541_002_Type	Данные с каналов модулей RS (тип данных поставляется с пакетом MKLogic-500)
44.	mNTP	NFTSys.NTPDiag	Данные с диагностического канала устройства System (Диагностика NTP сервера) (тип данных поставляется с пакетом MKLogic-500)
45.	portRS	ARRAY [*] OF typeCfgPortRS	Настроечные параметры по портам интерфейсных модулей (описание структуры представлено в таблице 8.52)
46.	internalRS	ARRAY [*] OF typeRSinternal	Внутреннее состояние интерфейсных модулей (описание структуры представлено в таблице 8.52)

Структуры устройства Redundancy

В таблицах 8.57 - 8.59 представлено описания структур устройства Redundancy.

Таблица 8.57 – Описание структуры typeDiagRedundancy

№	Наименование переменной	Тип переменной	Описание переменной
1.	ErrCode	UDINT	Код ошибки резервирования
2.	DataSync	typeDataSync	Статистика синхронизации данных резервирования (описание структуры представлено в таблице 8.58)
3.	IsActiveStandalone	BOOL	TRUE - если активный ПЛК в состоянии Standalone, иначе FALSE
4.	ErrCounters	typeSyncErrCounters	Счётчики ошибок синхронизаций (описание структуры представлено в таблице 8.59)

Таблица 8.58 – Описание структуры typeDataSync

№	Наименование переменной	Тип переменной	Описание переменной
1.	cntSynchro	UDINT	Счетчик синхронизаций SYNCHRO области
2.	timeLastSynchro	UDINT	Последнее время синхронизаций SYNCHRO области, мс
3.	cntInitial	UDINT	Счетчик синхронизаций INITIAL области
4.	timeLastInitial	UDINT	Последнее время синхронизаций INITIAL области, мс

Таблица 8.59 – Описание структуры typeSyncErrCounters

№	Наименование переменной	Тип переменной	Описание переменной
1.	ToutSyncCycle	UDINT	Счётчик таймаутов синхронизации времени циклов
2.	ToutSyncARI	UDINT	Счётчик таймаутов в момент After Read Inputs
3.	ToutSyncAWO	UDINT	Счётчик таймаутов в момент After Write Outputs
4.	ErrorCRC	UDINT	Счётчик несовпадений CRC оперативных данных

8.3.3. Работа с библиотекой в проекте

Создание проекта MKLogic-500

Создание проекта происходит так же, как описано в справочной системе Procyon IDE. Рекомендуется создавать проект с помощью диалога «Стандартный проект». Для этого необходимо нажать на пункт меню «Файл» → «Новый проект», затем выбрать шаблон «Проекты» → «Стандартный проект». Ввести имя проекта (например: samplePowerlink) и расположение в файловой системе.

В результате откроется диалог «Стандартный проект».

Далее нужно выбрать устройство «Nefteavtomatika MKLogic- 500 (505)» из списка «Устройство» и «Структурированный текст (ST)» из списка «PLC_PRG». Затем нажать ОК.

В результате будет создан проект со стандартной структурой.

Имя проекта samplePowerlink показано на панели заголовка главного окна системы разработки.

Далее, согласно руководству по программированию MKLogic-500 соберите структуру проекта, добавив все необходимые устройства, и установите последнюю версию пакета MKLogic-500.

Встраивание библиотеки в проект

В среде Procyon IDE выберите меню Инструменты → Репозиторий библиотек (рисунок 8.1). В открывшемся окне нажмите кнопку «Установить» и выберите файл с названием diagMK500.compiledlibrary. Повторив аналогичную процедуру добавьте файл moduleInfo.compiledlibrary.

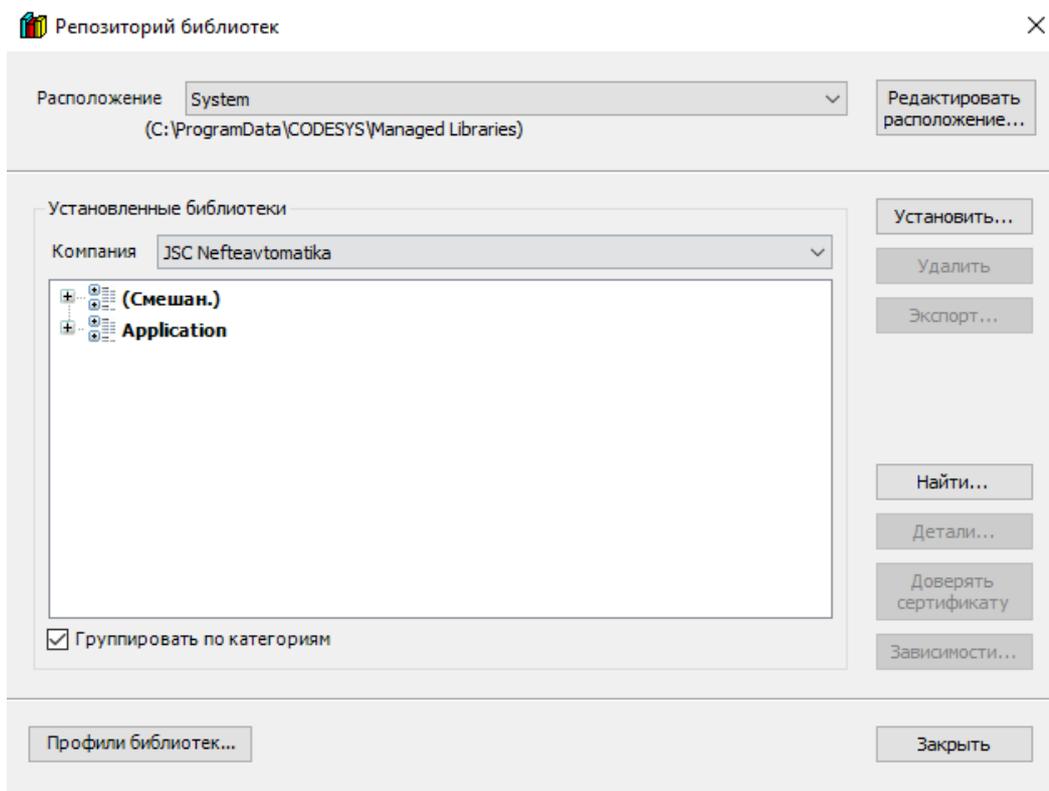


Рис. 8.1 - Репозиторий библиотек

Для добавления установленных в репозиторий библиотек нажмите на «Менеджер библиотек» в панели устройств. Окно менеджера библиотек представлено на рисунке 8.2.

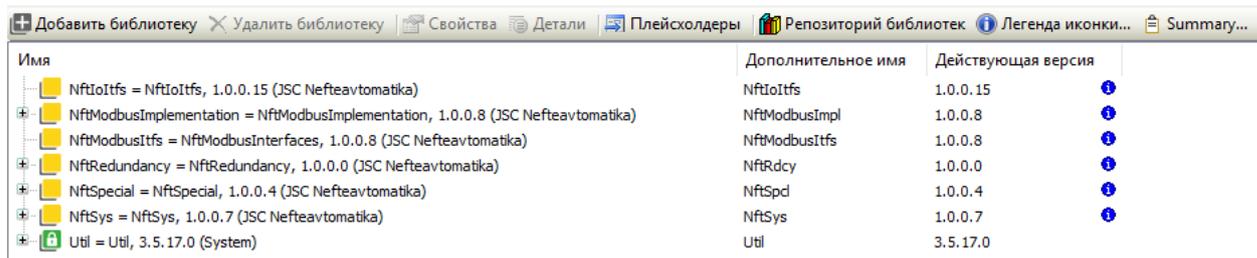


Рис. 8.2 - Менеджер библиотек

Нажмите на кнопку «Добавить библиотеку» и в разделе «Смешанные» выделите diagMK500, нажмите на кнопку «OK». Аналогичным образом добавьте в проект библиотеку moduleInfo.

Создание и конфигурирование переменных

В панели устройств нажмите правой кнопкой по приложению «Application» и последовательно добавьте списки глобальных переменных: cfgDiag, cfgHW, hw. В глобальном объекте cfgHW опишите константное количество единиц оборудования (листинг 8.10).

```

cnt      : typeCfgDiagCount :=      (Rack          := 4,
                                     CPU             := 2,
                                     modPSU          := 4,
                                     modMN           := 2,
                                     modCN           := 2,
                                     modEthEx        := 0,
                                     modAI           := 4,
                                     modAO           := 2,
                                     modDI           := 1,
                                     modDO           := 1,
                                     modRS           := 1);
    
```

Листинг 8.10 - Описание количества единиц оборудования

В том же глобальном объекте создайте массивы, с размерностью указанной ранее и типом данных typeCfgCPU (таблица 8.14) для модулей CPU и typeCfgMod (таблица 8.13) для всех остальных модулей. Заполните элементы массива константными значениями согласно вашей конфигурации устройств. Пример заполнения для типа typeCfgMod представлен в листинге 8.11.

```
mCN      :   ARRAY[1..cfgHW.cnt.modCN] OF typeCfgMod :=   [
              (typ := MKLogic500.MK_545_010,
               pos := (OpwlId := 1, RackOffset := 1, Slot := 3, Id := 3),
               portsEnbl := 3),
              (typ := MKLogic500.MK_545_010,
               pos := (OpwlId := 2, RackOffset := 1, Slot := 3, Id := 3),
               portsEnbl := 3)
            ];
```

Листинг 8.11 - Конфигурация модулей CN (МК-545-010)

В структуре typeCfgMod параметр portsEnbl является обязательным для заполнения только для модулей CN (МК-545-010), RS (МК-541-002).

Пример заполнения для типа typeCfgModCPU представлен в листинге 8.12.

```
mCPU:      ARRAY[1..cfgHW.cnt.CPU] OF typeCfgCPU :=   [
              (typ := MKLogic500.MK_504_120,
               pos := (OpwlId := 240, RackOffset := 1, Slot := 3, Id := 3),
               MN := (slot := 2, portsEnbl := 3),
               EthEx := [(slot := 0, portsEnbl := 0), (), (), ()],
               portsEnbl := 7),
              (typ := MKLogic500.MK_504_120,
               pos := (OpwlId := 240, RackOffset := 2, Slot := 3, Id := 3),
               MN := (slot := 2, portsEnbl := 3),
               EthEx := [(slot := 0, portsEnbl := 0), (), (), ()],
               portsEnbl := 7)
            ];
```

Листинг 8.12 - Конфигурация модулей CPU (МК-505-120)

При отсутствии модулей MN (МК-546-010) и EthEx (МК-544-040) в корзине с CPU (МК-505-120) допускается не заполнять их параметры конфигурации.

В глобальный объект hw необходимо добавить массивы, в которых будут храниться «сырые» данные с диагностических каналов устройств. Пример объявления массивов в объекте hw представлен в листинге 8.13.

```
{attribute 'qualified_only'}
VAR_GLOBAL
  NTPDiag:   NTPDiag;
  CPU:      ARRAY[1..cfgHW.cnt.CPU] OF NftIoItfs.MK_504_120_Type;
  PSU:      ARRAY[1..cfgHW.cnt.Rack] OF NftIoItfs.MK_550_024_Type;
  EthEx:    ARRAY[1..cfgHW.cnt.CPU] OF ARRAY [1..4] OF NftIoItfs.MK_544_040_Type;
  MN:      ARRAY[1..cfgHW.cnt.modMN] OF NftIoItfs.MK_546_010_Type;
  CN:      ARRAY[1..cfgHW.cnt.modCN] OF diagMK500.typeDeviceCN;
  mAI:     ARRAY[1..cfgHW.cnt.modAI] OF diagMK500.typeDeviceAI;
  mA0:     ARRAY[1..cfgHW.cnt.modA0] OF diagMK500.typeDeviceA0;
  mDI:     ARRAY[1..cfgHW.cnt.modDI] OF diagMK500.typeDeviceDI;
  mDO:     ARRAY[1..cfgHW.cnt.modDO] OF NftIoItfs.MK_531_032_Type;
  mRS:     ARRAY[1..cfgHW.cnt.modRS] OF NftIoItfs.MK_541_002_Type;
  RedundancyDiag : typeDiagRedundancy;
END_VAR
```

Листинг 8.13 - Данные с диагностических каналов устройств

Глобальный объект cfgDiag должен хранить следующие массивы: portRS типа typeCfgPortRS (таблица 8.51), internalRS типа typeRSinternal (таблица 8.52), массивы состояния исправности модулей AI, DI, AO, DO, массив со значениями в мКА с каналов AI модулей. Глобальный объект cfgDiag необходимо заполнить как в листинге 8.14.

Ссылки в массиве portRS необходимо заполнить единожды при инициализации программы. Для порта интерфейсного модуля, работающего в режиме Modbus Master, ссылка pCounter1⁽²⁾ не заполняется. Ссылки pTimeOut1⁽²⁾ также являются опциональными, заполняются только в том случае, если необходимо ввести задержку на потерю связи с портом интерфейсного модуля.

```
{attribute 'qualified_only'}
VAR_GLOBAL
  portRS:      ARRAY [1..cfgHW.cnt.modRS] OF diagMK500.typeCfgPortRS;
  internalRS:  ARRAY [1..cfgHW.cnt.modRS] OF diagMK500.typeRSinternal;
  mAHealth:    ARRAY [1..cfgHW.cnt.modAI] OF BOOL;
  mAoHealth:   ARRAY [1..cfgHW.cnt.modAO] OF BOOL;
  mDIHealth:   ARRAY [1..cfgHW.cnt.modDI] OF BOOL;
  mDOHealth:   ARRAY [1..cfgHW.cnt.modDO] OF BOOL;
  mkAI:        ARRAY [1..cfgHW.cnt.modAI, 1..16] OF UINT;
END_VAR
```

Листинг 8.14 - Заполнение глобального объекта cfgDiag

Далее нажмите правой кнопкой мыши по объекту «Application» и добавьте Persistent переменные. В них рекомендуется разместить массивы с обработанными блоком getDiagRedu (getDiagNoRedu) состояниями оборудования, а также подаваемый на вход блока командный регистр. Для удобства отладки и простоты навигации рекомендуется структурировать итоговые состояния. Пример представлен в структуре typeStateDiag (листинг 8.15)

```
TYPE typeStateDiag :
STRUCT
  mBUS:          ARRAY[1..cfgHW.cnt.Rack] OF DWORD;
  mBUSandCh:     ARRAY[1..cfgHW.cnt.Rack] OF DWORD;
  mBUSblink:     ARRAY[1..cfgHW.cnt.Rack] OF DWORD;
  diagPSU:       ARRAY[1..cfgHW.cnt.modPSU] OF diagMK500.typePSU_HMI;
  diagCPU:       ARRAY[1..cfgHW.cnt.CPU] OF diagMK500.typeCPU_HMI;
  diagCN:        ARRAY[1..cfgHW.cnt.modCN] OF diagMK500.typeCN_HMI;
  diagMN:        ARRAY[1..cfgHW.cnt.modMN] OF diagMK500.typeMN_HMI;
  diagAI16:      ARRAY[1..cfgHW.cnt.modAI] OF diagMK500.typeAI16_HMI;
  diagAI8:       ARRAY[1..cfgHW.cnt.modAI] OF diagMK500.typeAI8_HMI;
  diagAI16hart:  ARRAY[1..cfgHW.cnt.modAI] OF diagMK500.typeAI16hart_HMI;
  diagAI8hart:   ARRAY[1..cfgHW.cnt.modAI] OF diagMK500.typeAI8hart_HMI;
  diagAO:        ARRAY[1..cfgHW.cnt.modAO] OF diagMK500.typeAO_HMI;
  diagAOhart:    ARRAY[1..cfgHW.cnt.modAO] OF diagMK500.typeAOhart_HMI;
  diagDI:        ARRAY[1..cfgHW.cnt.modDI] OF diagMK500.typeDI_HMI;
  diagDInamur:   ARRAY[1..cfgHW.cnt.modDI] OF diagMK500.typeDInamur_HMI;
  diagDO:        ARRAY[1..cfgHW.cnt.modDO] OF diagMK500.typeDO_HMI;
  diagRS:        ARRAY[1..cfgHW.cnt.modRS] OF diagMK500.typeRS_HMI;
  diagEthEx:     ARRAY[1..cfgHW.cnt.modEthEx] OF diagMK500.typeEthEx_HMI;
  diagNTP:       diagMK500.typeNTP_HMI;
END_STRUCT
END_TYPE
```

Листинг 8.15 - Структура с обработанными состояниями модулей

Объявление переменной типа typeStateDiag и командного регистра представлено в листинге 8.16.

```
VAR_GLOBAL PERSISTENT
  stateDiag   :   typeStateDiag;
  cmdDiag     :   WORD;
END_VAR
```

Листинг 8.16 - Persistent переменные

Получение данных с диагностических каналов

Для получения данных в массивы, описанные в глобальном объекте hw, применяются функции ModuleMapIOData и PCIModuleMapIOData из библиотеки NftIoImplementation (MKLogic-500 Procyon IDEs Package). Процедура привязки данных с диагностических каналов, к объявленным в прикладном ПО переменным осуществляется по событию.

DownloadDone. Для добавления обработки события в панели устройств необходимо зайти в объект «Конфигурация задач», во вкладке «Системные события» нажать на кнопку «Добавить обработчик событий», выбрать событие «DownloadDone», ввести название вызываемой функции (например: «AfterDownloadDone») и нажать на кнопку «ОК». Окно системных событий представлено на рисунке 8.3.

Имя	Описание	Контекст	Отладка	Вызываемая функция	Активный
DownloadDone	Called after application online download	Communication task	✗	AfterDownloadDone	<input checked="" type="checkbox"/>
BeforeReadingInputs	Called before reading inputs	IEC task	✓	BRICallback	<input checked="" type="checkbox"/>

Рис. 8.3 - Системные события

Параметры и возвращаемые значения функции ModuleMapIOData представлены в таблице 8.60.

Таблица 8.60 Параметры и возвращаемые значения функции ModuleMapIOData

Область	Имя	Тип переменной	Описание переменной
Return	ModuleMap IOData	BOOL	TRUE – функция выполнена успешно
Input	Position	NftIoIltfs. IoCommonPositionInfoType	Позиция модуля. Конфигурируется в параметре pos из листинга 8.11
Input	ModuleType	NftIoIltfs. MKLogic500	Тип модуля. Конфигурируется в параметре typ из листинга 8.11
Input	pData	POINTER TO BYTE	Указатель на привязываемую переменную
Input	DataSize	UINT	Размер привязываемой переменной
Output	ulResult	RTS_IEC_RESULT	Код ошибки при неуспешном выполнении

Пример вызова функции ModuleMapIOData для получения данных о модуле CN (МК-545-010) представлен в листинге 8.17.

```
FOR i := 1 TO cfgHW.cnt.modCN DO
NftIoImpl.ModuleMapIODataCheck (Position := cfgHW.mCN[i].pos,
ModuleType := cfgHW.mCN[i].typ,
pData := ADR(hw.CN[i].module),
DataSize := sizeof(hw.CN[i].module));
END_FOR;
```

Листинг 8.17 - Вызов функции ModuleMapIOData для получения данных о модуле CN

Функция PCIModuleMapIOData предназначена для получения диагностической информации о модулях, являющихся расширением модуля центрального процессора. Таковыми являются модуль MN (МК-546-010) и EthEx (МК-544-040). Параметры и возвращаемые значения функции PCIModuleMapIOData представлены в таблице 8.61.

Таблица 9.61 Параметры и возвращаемые значения функции PCIModuleMapIOData

Область	Имя	Тип переменной	Описание переменной
Return	PCIModule MapIOData	BOOL	TRUE – функция выполнена успешно
Input	CPUPosition	NftIoIltfs. IoCommonPositionInfoType	Позиция модуля CPU. Конфигурируется в параметре pos из листинга 8.12
Input	PCISlot	USINT	Номер слота модуля на PCI шине. Конфигурируется в параметре MN.slot (EthEx.slot) из листинга 8.12
Input	ModuleType	NftIoIltfs. MKLogic500	Тип модуля

Область	Имя	Тип переменной	Описание переменной
Input	pData	POINTER TO BYTE	Указатель на привязываемую переменную
Input	DataSize	UINT	Размер привязываемой переменной
Output	ulResult	RTS_IEC_RESULT	Код ошибки при неуспешном выполнении

Пример вызова функции PCIModuleMapIOData для получения данных о модуле MN (МК-546-010) представлен в листинге 8.18.

```

FOR i := 1 TO cfgHW.cnt.CPU DO
  NftIoImpl.PCIModuleMapIODataCheck(CPUPosition := cfgHW.mCPU[i].pos,
    PCISlot := cfgHW.mCPU[i].MN.slot,
    ModuleType := NftIoIifs.MKLogic500.MK_546_010,
    pData := ADR(hw.MN[i]),
    DataSize := SIZEOF(hw.MN[i]));
END_FOR;

```

Листинг 8.9 - Вызов функции PCIModuleMapIOData для получения данных о модуле MN

Для получения сведений о текущем состоянии резервирования контроллеров (Redundancy из панели устройств), синхронизации времени (System из панели устройств), узла сети Powerlink (CN из панели устройств) привязка к диагностическим каналам осуществляется вручную.

Пример привязки переменных к диагностическим каналам устройства Redundancy представлен на рисунке 8.4.

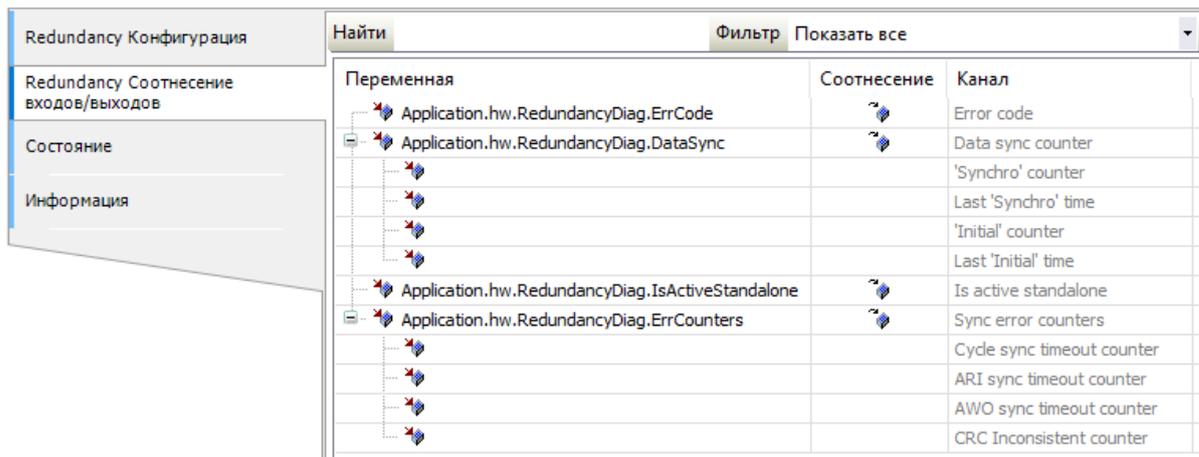


Рис. 8.4 - Привязка переменных к диагностическим каналам устройства Redundancy

Пример привязки переменных к диагностическим каналам устройства System представлен на рисунке 8.5.

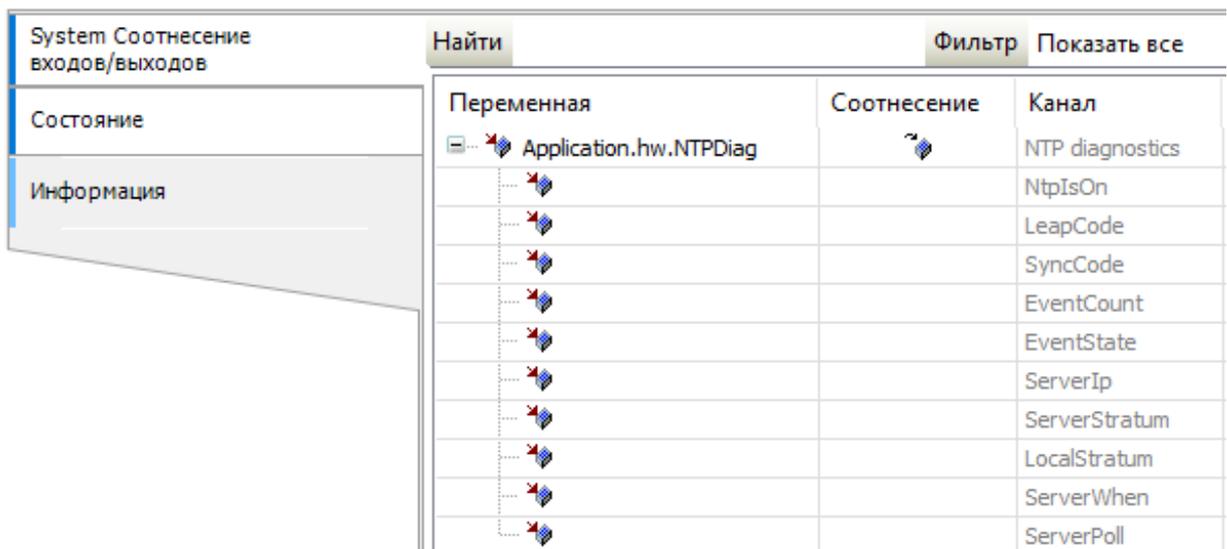


Рис. 8.5 - Привязка переменных к диагностическим каналам устройства System

Пример привязки переменных к диагностическим каналам устройства CN представлен на рисунке 8.6.

CN Конфигурация	Найти		Фильтр	Показать все
	Переменная	Соотн...	Канал	
CN Соотнесение входов/выходов	Application.hw.CN[1].pwl_ID	↔	Powerlink ID	
	Application.hw.CN[1].cfgIO_Ok	↔	IO config ok	
Состояние				
Информация				

Рис. 8.6 - Привязка переменных к диагностическим каналам устройства CN

Получение обработанных данных

Перейдите в Application → Добавление объекта → POU и в открывшемся окне (рисунок 8.7) выберите тип – программа, язык реализации – ST. Введите имя создаваемой программы (например Diag).

Добавить POU ✕

Создать новый POU (компонент организации программы)

Имя

Тип

Программа

Функциональный блок

Extends ...
 Implements ...
 Окончательный Абстрактный

Спецификатор доступа

Язык реализации метода:

Функция

Тип возвращаемого значения ...

Язык реализации

Рис. 8.7 - Окно добавления пользовательской программы

Затем перейдите в окно конфигурации исполняемой задачи (Task), представленное на рисунке 8.8 и нажмите на кнопку «Добавить вызов». В появившемся окне выберите ранее добавленную программу Diag.

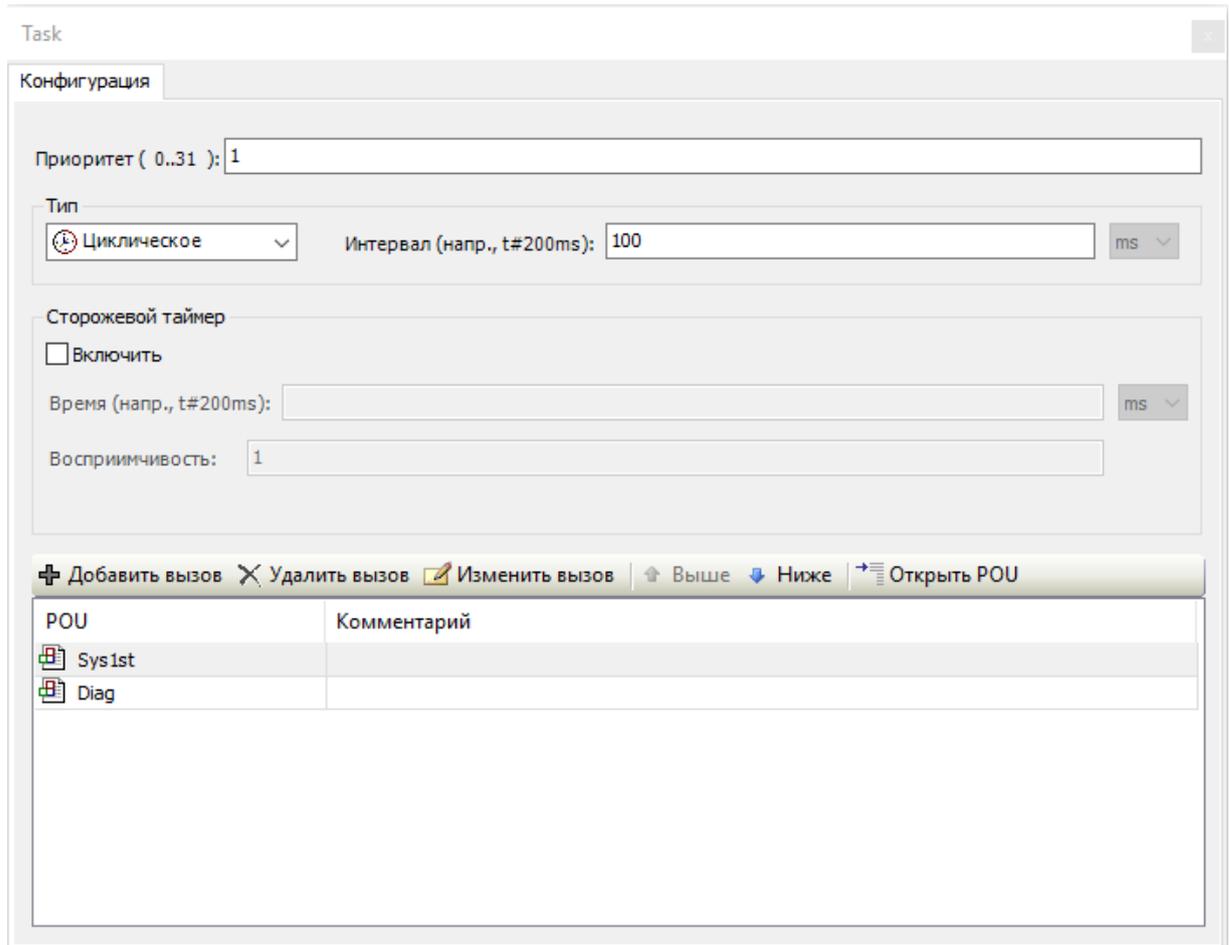


Рис. 8.8 - Окно конфигурации исполняемой задачи

В программе Diag объявите экземпляр блока getDiagRedu, если используете резервируемую пару контроллеров или getDiagNoRedu, если резервирование контроллеров отсутствует. Осуществите вызов экземпляра блока. Пример вызова представлен в листинге 8.19.

```

instGetDiagRedu (imitOn           := FALSE,
                 redundancyDiag   := hw.RedundancyDiag,
                 msBlk            := gv.timeBlk.mSecBlk,
                 rsTimeOut        := 1,
                 timeNotSyncTimeOut := 80,
                 plcNotSyncTimeOut := 1,
                 cmdDiag          := nvm.cmdDiag,
                 healthAI         := cfgDiag.mAIhealth,
                 healthAO         := cfgDiag.mAOhealth,
                 healthDI         := cfgDiag.mDIhealth,
                 healthDO         := cfgDiag.mDOhealth,
                 cfgPSU           := cfgHW.mPSU,
                 cfgCPU           := cfgHW.mCPU,
                 cfgCN            := cfgHW.mCN,
                 cfgAI            := cfgHW.mAI,
                 cfgAO            := cfgHW.mAO,
                 cfgDI            := cfgHW.mDI,
                 cfgDO            := cfgHW.mDO,
                 cfgRS            := cfgHW.mRS,
                 mkAI             := cfgDiag.mkAI,
                 mBUS             := nvm.stateDiag.mBUS,
                 mBUSandCh        := nvm.stateDiag.mBUSandCh,
                 mBUSblink        := nvm.stateDiag.mBUSblink,
                 statePSU         := nvm.stateDiag.diagPSU,

```

```

stateCPU           :=      nvm.stateDiag.diagCPU,
stateCN           :=      nvm.stateDiag.diagCN,
stateMN           :=      nvm.stateDiag.diagMN,
stateAI16         :=      nvm.stateDiag.diagAI16,
stateAI8          :=      nvm.stateDiag.diagAI8,
stateAI16hart     :=      nvm.stateDiag.diagAI16hart,
stateAI8hart      :=      nvm.stateDiag.diagAI8hart,
stateAO           :=      nvm.stateDiag.diagAO,
stateAOhart       :=      nvm.stateDiag.diagAOhart,
stateDI           :=      nvm.stateDiag.diagDI,
stateDInamur     :=      nvm.stateDiag.diagDInamur,
stateDO           :=      nvm.stateDiag.diagDO,
stateRS           :=      nvm.stateDiag.diagRS,
stateEthEx        :=      nvm.stateDiag.diagEthEx,
stateNTP          :=      nvm.stateDiag.diagNTP,
mPSU              :=      hw.PSU,
mCPU              :=      hw.CPU,
mEthEx            :=      hw.EthEx,
mMN               :=      hw.MN,
mCN               :=      hw.CN,
mAI               :=      hw.mAI,
mAO               :=      hw.mAO,
mDI               :=      hw.mDI,
mDO               :=      hw.mDO,
mRS               :=      hw.mRS,
mNTP              :=      hw.NTPDiag,
portRS            :=      cfgDiag.portRS,
internalRS        :=      cfgDiag.internalRS,
activePLCNum      =>      actPLC,
pwIRing           =>      pwIRing);

```

Листинг 8.19 - Вызов экземпляра блока getDiagRedu