

УТВЕРЖДЁН
КДСА.426471.004 РП-УЛ

КОНТРОЛЛЕР ПРОГРАММИРУЕМЫЙ ЛОГИЧЕСКИЙ *MKLogic-500®*

**Руководство по программированию
КДСА.426471.004 РП 1.1_00**

Уфа 2023 г.

Содержание

Глава 1	Подготовка к работе с изделием.....	1
1.1	Документация.....	1
1.2	Общие сведения о работе процессорных модулей изделия.....	2
1.3	Адреса интернет-ресурсов.....	5
Глава 2	Работа в среде разработки ACP Workbench ISaGRAF 6.5.....	6
2.1	Установка и настройка среды разработки ACP Workbench ISaGRAF 6.5.....	6
2.1.1	Установка предварительно требуемых приложений.....	6
2.1.2	Установка среды разработки ACP Workbench ISaGRAF 6.5.....	6
2.1.3	Регистрация среды разработки ACP Workbench ISaGRAF 6.5.....	6
2.1.4	Установка плагина MK500 IODevice.....	7
2.2	Создание проекта в среде разработки ACP Workbench ISaGRAF 6.5.....	8
2.3	Настройка и диагностика модулей CPU.....	11
2.3.1	Подключение к модулю CPU с помощью плагина MK500 IODevice.....	13
2.3.2	Настройка и диагностика модуля CPU с помощью плагина MK500 IODevice.....	17
2.4	Настройка сетевых параметров проекта.....	34
2.5	Структура проекта.....	36
2.5.1	Устройства (Device) проекта.....	37
2.5.2	Ресурсы (Resource) проекта.....	38
2.6	Сборка проекта.....	44
2.7	Загрузка проекта в модули CPU.....	46
2.7.1	Загрузка проекта в модули CPU без остановки работы (Online Change).....	47
2.8	Мониторинг и управление работой программы в модуле CPU.....	49
2.8.1	Диагностика работы программы пользователя.....	51
2.8.2	Просмотр и модификация значения переменных программы пользователя.....	53
2.8.3	Управление работой ресурса программы пользователя.....	55
2.8.4	Парольная защита проекта и целевого устройства.....	57
2.9	Работа с эмулятором МК-500.....	58
2.9.1	Установка эмулятора.....	58
2.9.2	Работа с эмулятором.....	58
2.9.3	Ограничения эмулятора.....	59
Глава 3	Работа с модулями изделия в среде разработки ACP Workbench ISaGRAF 6.5.....	60
3.1	Общие принципы работы с модулями изделия.....	60
3.1.1	Добавление и удаление модулей изделия.....	61
3.1.2	Ранжирование модулей изделия.....	66
3.1.3	Настройка параметров модуля изделия.....	67
3.1.4	Привязка каналов модулей изделия.....	69

3.1.5	Настройки параметров каналов модулей изделия	72
3.1.6	Диагностический канал модулей изделия	74
3.2	Общие принципы работы с дополнительными модулями ввода-вывода	76
3.3	Общие принципы работы с модулями центрального процессора	78
3.4	Общие принципы работы с модулями ввода-вывода.....	79
3.5	Общие принципы работы с коммуникационными модулями	80
3.6	Работа с модулями питания МК-550-024	82
3.7	Работа с модулями центрального процессора МК-501-022, МК-502-142 и МК-503-120	83
3.7.1	Реализация протоколов Modbus RTU и Modbus TCP (ведущий) в модулях CPU	88
3.7.2	Реализация протоколов Modbus RTU и Modbus TCP (ведомый) в модулях CPU ...	101
3.7.3	Реализация протокола IEC 60870-5-104 (сервер) в модулях CPU	106
3.7.4	Реализация протокола OPC UA (сервер) в модулях CPU МК-503-120	129
3.7.5	Реализация протокола Powerlink (ведущий) в модулях CPU МК-502-142 и МК-503-120	136
3.7.6	Сбор диагностической информации модуля CPU.....	145
3.7.7	Получение расширенной диагностики для модуля CPU МК-503-120.....	147
3.7.8	Получение значения текущего времени активного модуля CPU	152
3.7.9	Дополнительная ring-диагностика удалённых серверов	153
3.7.10	Получение системных диагностических сообщений модуля CPU	155
3.8	Работа с модулями аналогового ввода МК-513-016 и МК-513-016А.....	156
3.9	Работа с модулями аналогового вывода МК-514-008 и МК-514-008А	158
3.10	Работа с модулями аналогового ввода МК-516-008 и МК-516-008А.....	160
3.11	Работа с модулями дискретного ввода МК-521-032 и МК-521-032 А.....	162
3.12	Работа с модулями дискретного вывода МК-531-032 и МК-531-032 А	165
3.13	Работа с коммуникационными модулями МК-541-002	168
3.13.1	Режимы команд коммуникационного модуля МК-541-002	171
3.13.2	Инициализация и передача команд в модуль rs485new.....	172
3.13.3	Особенности работы Modbus-функций 1, 2, 3 и 4 в модуле rs485new.....	174
3.13.4	Особенности работы Modbus-функций 1, 2, 5 и 15 в модуле rs485new.....	175
3.14	Работа с коммуникационными модулями МК-545-010	176
3.14.1	Реализация протокола Powerlink (ведомый) в модулях cn545	176
3.15	Работа с модулями аналогового вывода МК-574-008А.....	178
3.16	Работа с модулями аналогового ввода МК-576-008А	182
3.17	Работа с модулями аналогового ввода МК-576-016А	183
3.18	Работа с модулями дискретного ввода NAMUR МК-523-032А	185
3.19	Работа с модулями дискретного вывода NAMUR МК-532-032А.....	187
Глава 4	Использование функций расширения МК500	188
4.1	Функции ByteSwap и WordSwap	190

4.2	Функции WordsToReal и RealToWords.....	191
4.3	Функции WordsToLReal и LRealToWords.....	193
4.4	Функции DWordToReal и RealToDWord.....	195
4.5	Функция Safe2DimWordArrayCopy.....	196
4.6	Функции SafeBoolArrayCopy и SafeWordArrayCopy.....	197
4.7	Функции SafeCopyFromModbusREGsArray и SafeCopyToModbusREGsArray.....	199
4.8	Функции WriteReal2DimArray и ReadReal2DimArray.....	201
4.9	Функция CompareWordArrays.....	203
4.10	Функция CRC16ForModbusRegs.....	204
4.11	Функция CRC16ForWords.....	205
4.12	Функция InitRS485ModuleModbus.....	206
4.13	Функция SendMessage.....	206
4.14	Функция SetDateTime.....	207
4.15	Функция UdpMessage.....	209
4.16	Функция SwapActiveCPU.....	210
4.17	Функции GetBitFromDInt, GetBitFromDWord и GetBitFromWord.....	211
4.18	Функции GetByteFromDWord и GetByteFromWord.....	213
4.19	Функция GetWordFromDWord.....	214
4.20	Функции SetBitToDInt, SetBitToDWord и SetBitToWord.....	215
4.21	Функции SetByteToDWord и SetByteToWord.....	217
4.22	Функция SetWordToDWord.....	218
4.23	Функция SynchronizeFtpFiles.....	219
4.24	Функция PredictByDerivative.....	220
4.25	Функция GetDevInfo.....	221
4.26	Функция GetFastModuleState.....	222
4.27	Функция FTPEnable.....	223
4.28	Функция UpdateOutputData.....	224
4.29	Функция ConvertToIec104Time.....	225
4.30	Функции WriteLogToFile и DeleteLogFile.....	226
Глава 5	Рекомендации по работе с модулями CPU.....	227
5.1	Оптимизация времени выполнения программ пользователя.....	227
5.1.1	Уменьшение числа параметров функций и функциональных блоков.....	227
5.1.2	Влияние параметра ресурса «Function Internal State Enabled».....	228
5.1.3	Работа с сохраняемыми (RETAIN) переменными.....	229
5.1.4	Использование оператора WHILE вместо FOR.....	230
5.1.5	Использование промежуточных переменных.....	231
5.1.6	Выход за пределы массива.....	232
5.2	Правила написания программ для модулей CPU с поддержкой режима Failover.....	233

5.2.1	Диагностика проблем при работе процессорных модулей с поддержкой режима Failover.....	234
5.3	Диагностика среды выполнения и программы пользователя.....	235
5.3.1	Журнал сообщений программы пользователя.....	235
5.3.2	Журнал запуска среды выполнения и программы пользователя.....	240
5.3.3	Оценка времени исполнения отдельных секций цикла ISaGRAF.....	241
5.4	Часто встречающиеся проблемы и меры по их устранению.....	243
5.4.1	Проблемы при загрузке программы пользователя.....	243
5.4.2	Зацикливание программы пользователя.....	244
5.4.3	Ошибка запуска программы пользователя.....	245
5.4.4	Ложные ошибки диагностики.....	246
5.4.5	Ошибки при сборке проекта.....	247
5.4.6	Проблемы с работой модулей на CAN-шине.....	248
5.4.7	Проблемы при работе по протоколу Modbus.....	250
5.4.8	Проблемы при работе по протоколу IEC 60870-5-104.....	251
5.4.9	Проблемы при работе по протоколу OPC-UA.....	252
5.4.10	Проблемы при работе по протоколу Powerlink.....	253
5.5	Сетевая безопасность.....	254
Приложение А. Инструкция по интеграции в PDM-систему.....		255
Приложение Б. Индикация модулей MKLogic-500.....		259
Лист регистрации изменений.....		279

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 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 **Контроллер программируемый логический МКLogic-500**.

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

В РП приняты следующие условные обозначения:

Обозначение	Комментарий
 ВНИМАНИЕ	Информация, на которую следует обратить особое внимание
Шрифт Courier New	Названия функций, параметров, переменных, папок, документов, листинги примеров
<u>Подчёркнутое начертание</u>	Ссылки на разделы руководства, рисунки и таблицы
«Текст в кавычках»	Названия пунктов меню, кнопок

Глава 1 Подготовка к работе с изделием

1.1 Документация

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

- КДСА.426471.004 РЭ_2.0_00 Контроллер программируемый логический MKLogic-500 Руководство по эксплуатации;
- Спецификация прикладного уровня и коммуникационного профиля CANopen CiA 301 (CANopen_CiA_301_spec_rus);
- ISaGRAF 6 (cam_ISa6_ru);
- ISaGRAF 6 - Начальное руководство (gst_ISa6_ru);
- Браузер перекрёстных ссылок (crb_ISa6_ru);
- Глоссарий ISaGRAF (gloss_ISa6_ru);
- Лицензирование (lmr_ISa6_ru);
- Механизм обеспечения отказоустойчивости (fvr_ISa6_ru);
- Монтаж ввода-вывода (iow_ISa6_ru);
- Руководство по языкам (lrf_ISa6_ru);
- Связывания (bnd_ISa6_ru);
- Словарь (dct_ISa6_ru);
- Стандартные операции (lrso_ISa6_ru);
- Функции (lrsf_ISa6_ru);
- Функциональные блоки (lrbs_ISa6_ru);
- Язык ST (stx_ISa6_ru).

Ознакомление со следующими документами является желательным, но не обязательным:

- Единая платформа автоматизации. Параметры среды разработки. (ode_ru);
- Генерация документов (dgn_ISa6_ru);
- Язык SAMA (sama_ISa6_ru);
- Язык SFC (sfc_ISa6_ru);
- Язык LD (ldr_ISa6_ru);
- Язык FBD (fbd_ISa6_ru);
- Язык IEC 61499 (iec_ISa6_ru);
- Нормативные функциональные блоки (lrnb_ISa6_ru).

Все вышеперечисленные документы поставляются вместе со средой разработки ACP Workbench ISaGRAF 6.5 и располагаются на диске в папке Docs (здесь и далее, все пути и имена файлов приводятся относительно папки с установочным комплектом ACP Workbench ISaGRAF 6.5).

1.2 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 (режим Failover). Однако они могут использоваться и не в паре, при этом поддержку режима Failover рекомендуется отключать ([п. 2.4](#)).

При работе режиме Failover один из модулей CPU всегда является активным (находится в режиме Primary), второй же является резервным (находится в режиме Secondary). Кроме исполнения программы пользователя, в режиме Failover модули CPU 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, [п. 2.3.1](#)). 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 ([п. 2.3.1](#));
- пропадание связи на сетевом интерфейсе в роли Uplink IP ([п. 2.3.1](#)) активного CPU при наличии связи на соответствующем сетевом интерфейсе резервного CPU;
- пропадание одного из контролируемых удалённых серверов для активного CPU (при настроенном устройстве `ping_diag`, [п. 3.7.9](#));
- полное пропадание связи с остальными устройствами на шине Powerlink активного модуля CPU при наличии связи с устройствами Powerlink у резервного модуля CPU (только для МК-503-120).



ВНИМАНИЕ

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

При старте модуль CPU с включённым режимом Failover 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 (п. 2.3.1), 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 бóльшим адресом на шине CAN. Настройки ведущего модуля CPU имеют приоритет над настройками ведомого модуля.
- Синхронизируются роли сетевых интерфейсов модулей CPU (п. 2.3.1), от ведущего модуля CPU к ведомому.
- После синхронизации ролей сетевых интерфейсов синхронизируются сетевые настройки модулей (п. 2.3.1), от ведущего модуля CPU к ведомому. Запускаемый модуль CPU применяет к себе сетевые настройки согласно своему положению на шине CAN (PR или SE), безотносительно 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, то 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 tdb-файлы, 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 (для модулей МК-503-120).
- Запускаемый модуль CPU передаёт управление ISaGRAF и переходит в штатный режим работы.

Таким образом, рекомендованная последовательность запуска модулей CPU такая:

- если оба модуля CPU настроены или оба модуля не настроены совершенно – запуск можно выполнять в любой последовательности;

- если один модуль настроен корректно и несёт в себе актуальную программу пользователя, а другой модуль не настроен (взят из ЗИПа) – первым следует запустить модуль с корректными настройками/программой, по возможности поместив его в позицию PR.

1.3 Адреса интернет-ресурсов

Скачать последние версии диагностических утилит, документации и обновлений для изделий серии МКLogic-500 можно с [официального сайта АО "Нефтеавтоматика"](http://www.nefteavtomatika.ru) (www.nefteavtomatika.ru, раздел Решения и продукция – Средства и системы промышленной автоматизации).

Следить за обновлениями можно с помощью официального телеграм-канала службы технической поддержки МКLogic-500: <https://t.me/NaftaSupport>.

Глава 2 Работа в среде разработки ACP Workbench ISaGRAF 6.5

2.1 Установка и настройка среды разработки ACP Workbench ISaGRAF 6.5

2.1.1 Установка предварительно требуемых приложений

Среда разработки ACP Workbench ISaGRAF 6.5 (далее среда разработки ACP) в ходе своей установки устанавливает сама почти все требуемые приложения, кроме Microsoft .NET Framework 4.5.1 и Microsoft .NET Framework 4.7.2. Перед началом установки настоятельно рекомендуется установить их самостоятельно, открыв папку Prerequisites\Microsoft .net\ и запустив файл NDP451-KB2858728-x86-x64-AllOS-ENU.exe и ndp472-kb4054530-x86-x64-allos-enu.exe.

2.1.2 Установка среды разработки ACP Workbench ISaGRAF 6.5

Для установки среды разработки ACP следует запустить программу–инсталлятор (файл Setup.exe) и следовать указаниям мастера установки.

При выборе языка настоятельно рекомендуется выбрать английский язык. В настоящем руководстве все рисунки и комментарии даны для англоязычной версии среды разработки.



ВНИМАНИЕ

Выбор русского языка при установке ACP может негативно повлиять на качество собираемого кода и на процесс отладки. Кроме того, перевод сообщений об ошибках в на русский язык не всегда выполнен верно, что может затруднить диагностику.

Установка может занять достаточно много времени в силу того, что устанавливается среда программирования MS Visual Studio 2013 Shell.

После завершения установки среды разработки ACP следует исправить её конфигурационные файлы, согласно рекомендациям документа: «Особенность работы ACP.pdf».

2.1.3 Регистрация среды разработки ACP Workbench ISaGRAF 6.5

После установки среды разработки ACP необходимо её зарегистрировать согласно рекомендациям документа «Активация ISaGRAF ACP 6.pdf». Без активации среда разработки не позволяет взаимодействовать с модулями центрального процессора изделия.

2.1.4 Установка плагина MK500 IODevice

Для конфигурирования модулей центрального процессора изделия необходимо установить плагин MK500 IODevice для среды разработки АСР. Для этого следует запустить файл `Plugin\MK500_IODevice_5.1.0.6.exe` и следовать указаниям мастера установки.

Последнюю версию плагина можно скачать с официального сайта АО «Нефтеавтоматика» либо с официального телеграм-канала службы технической поддержки MKLogic-500 (см. [п.1.3](#))

2.2 Создание проекта в среде разработки ACP Workbench ISaGRAF 6.5

Для создания проекта в среде разработки ACP следует выполнить следующие действия:

- 1) Запустить экземпляр приложения среды разработки ACP, дважды кликнув по ярлыку ISaGRAF 6.5 на рабочем столе либо выполнив Пуск→Все программы→ISaGRAF 6.5→Automation Collaborative Platform 6.5 - eng.

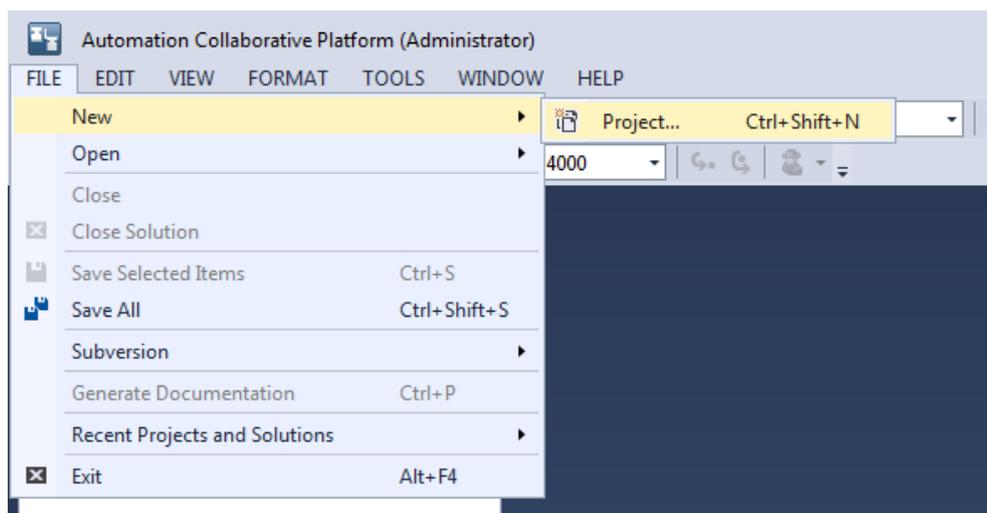


Рис. 2.1 – Создание нового проекта

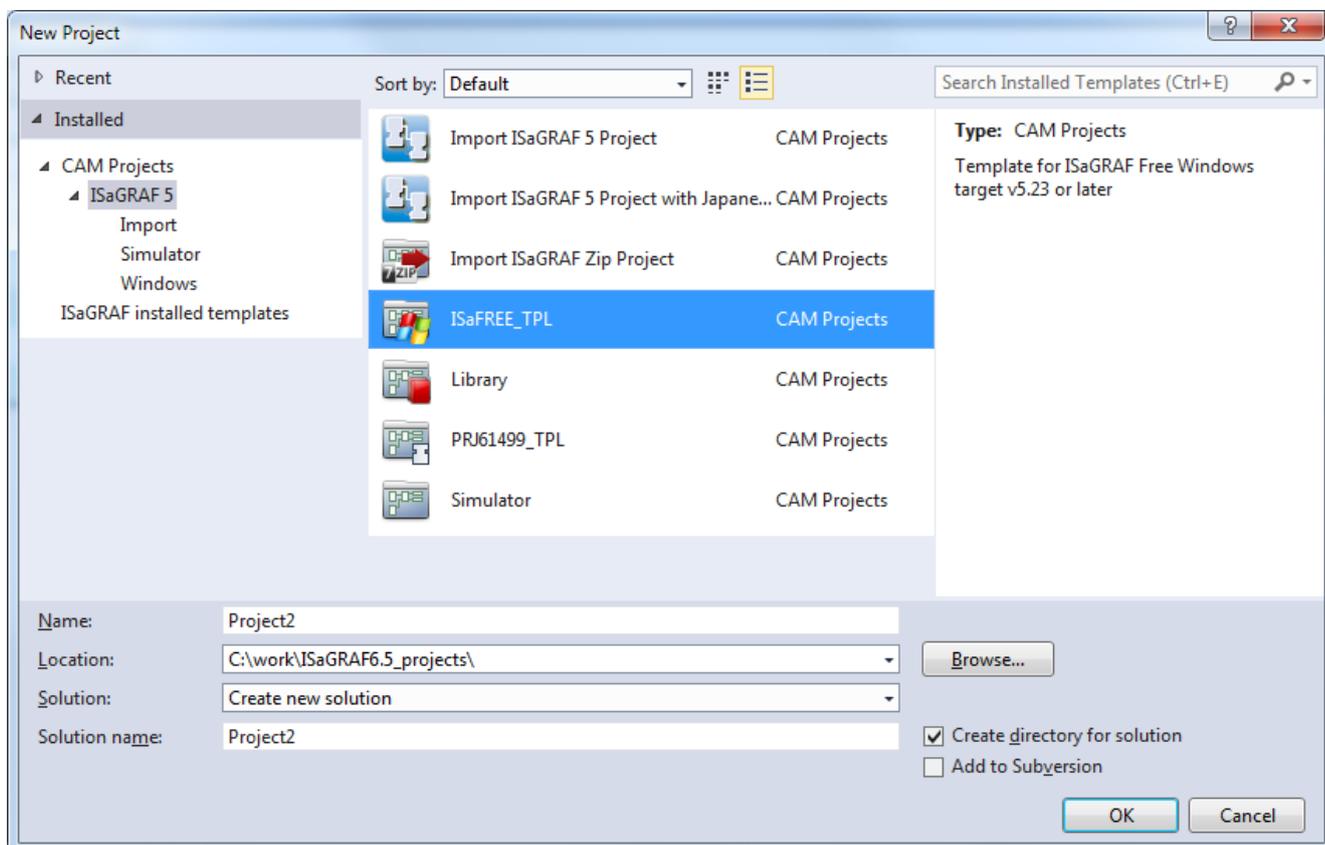


Рис. 2.2 – Окно создания нового проекта

2) После запуска среды разработки АСР следует создать проект, выполнив FILE→New→Project... (Рис. 2.1).

3) Далее следует импортировать файл определения свойств модулей изделия (далее – tdb-файл). См. п. 2.5. Для этого следует в контекстном меню проекта выбрать Import→Import Target Definitions (Рис. 2.3).

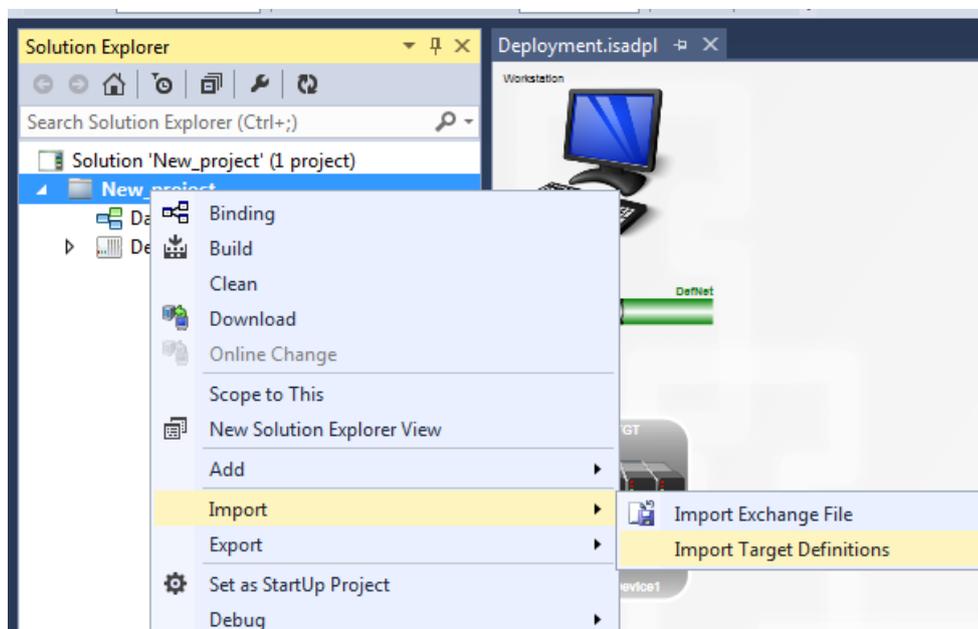


Рис. 2.3 – Импорт tdb-файла

В диалоговом окне следует выбрать актуальный tdb-файл (п.2.3) и нажать кнопку «Открыть» (Рис. 2.4).

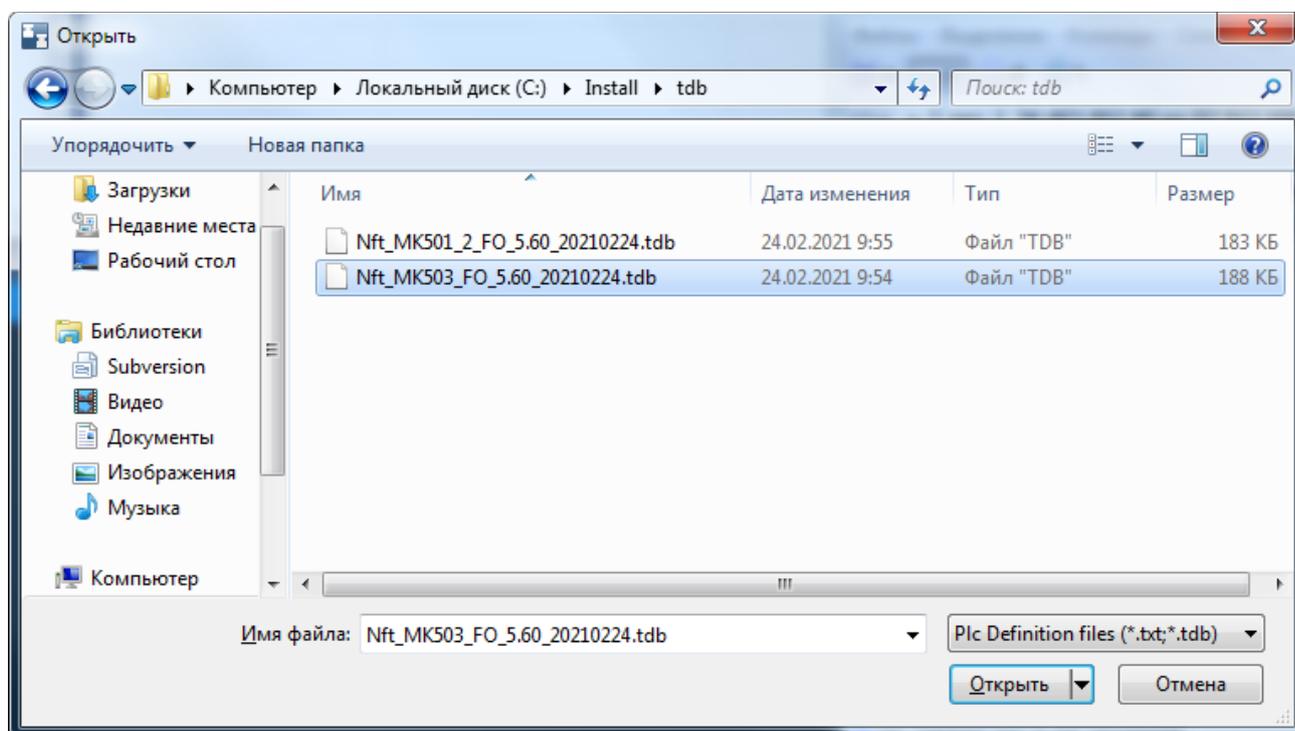


Рис. 2.4 – Окно импорта tdb-файла

⚠ ВНИМАНИЕ

Здесь и далее имена файлов в примерах могут отличаться от актуальных на момент прочтения документации.

Процесс импорта tdb-файла занимает несколько минут и сильно загружает процессор персонального компьютера.

После завершения импорта tdb-файла в блокноте будет открыт журнал импорта. Его можно закрывать сразу.

4) Для применения импортированного tdb-файла следует выбрать в дереве проекта устройство, и в окне «Properties» в секции «Hardware» для параметра «Target» выбрать «NFT_MK501_2-FO» или «NFT_MK503-FO» (Рис. 2.5), в зависимости от типа модуля центрального процессора. В ходе применения новых настроек будет выведено окно с перечнем отличий между старыми и новыми параметрами, для окончательного применения параметров следует нажать кнопку «Yes».

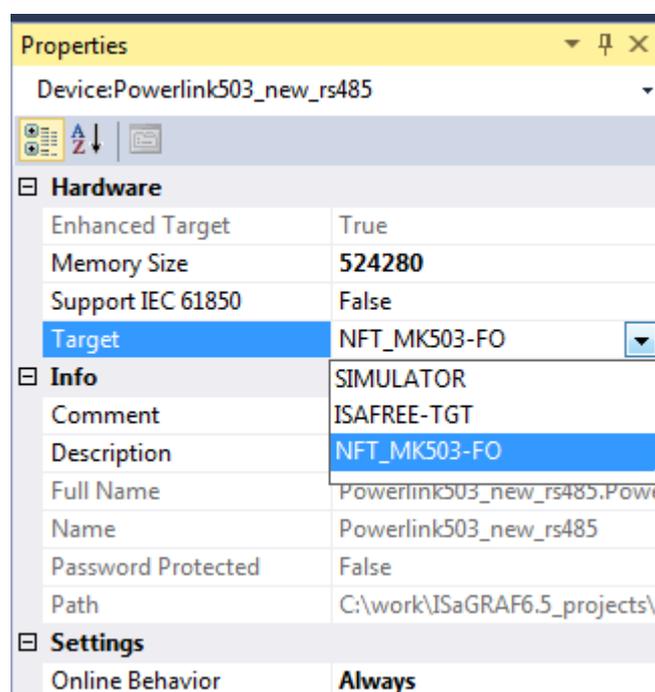


Рис. 2.5 – Применение импортированного tdb-файла

2.3 Настройка и диагностика модулей CPU

Модули CPU выпускаются с предустановленными сетевыми настройками. Для установки необходимых сетевых настроек, а также для настройки временных параметров и для выполнения диагностики следует использовать плагин MK500 IODevice.

Для вызова окна плагина MK500 IODevice следует выбрать в дереве проекта устройство и в его контекстном меню выбрать пункт «MK500 IODevice» (Рис. 2.6). Исходный вид окна плагина MK500 IODevice приведён на Рис. 2.7.

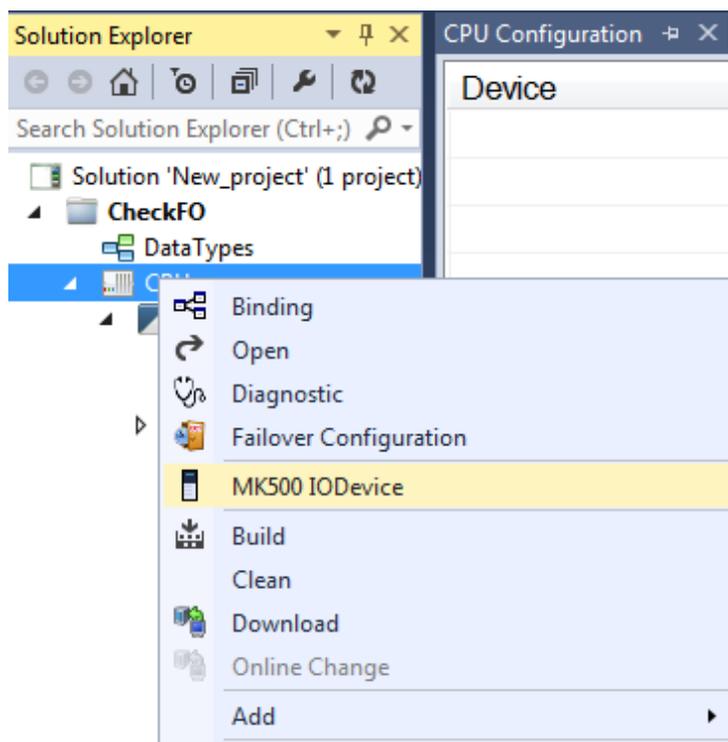


Рис. 2.6 – Вызов плагина MK500 IODevice

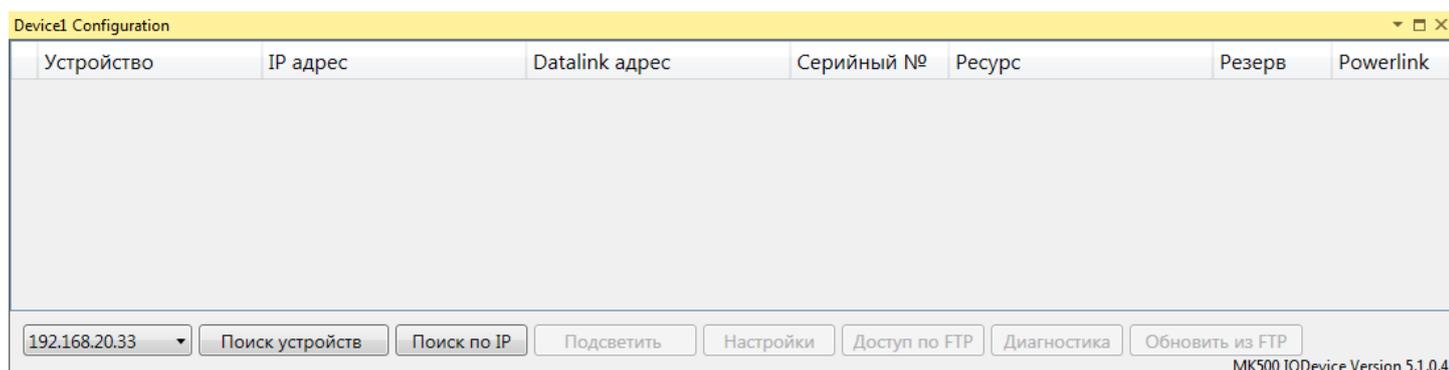


Рис. 2.7 – Исходный вид окна плагина MK500 IODevice

Также существует вариант плагина для работы вне среды разработки, распространяемый как приложение `Fallover.exe`. Его, как и плагин, можно бесплатно скачать с официального сайта АО «Нефтеавтоматика» (см. [п.1.3](#)). Функционально `Fallover.exe` полностью аналогичен плагину MK500 IODevice. В дальнейшем изложении для определённости все примеры приводятся для плагина MK500 IODevice.

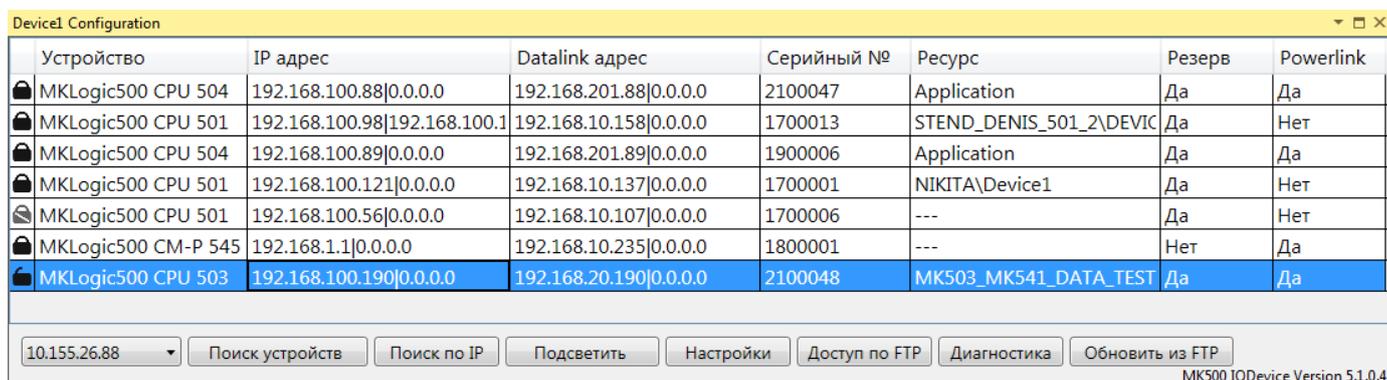
2.3.1 Подключение к модулю CPU с помощью плагина MK500 IODevice

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 (с установленной средой разработки АСР), либо к общему с рабочим ПК сетевому коммутатору.

2) В среде разработки АСР открыть окно плагина MK500 IODevice и 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. 2.8](#)). В таблицу выводятся следующая информация для каждого обнаруженного модуля CPU:

- Значок поддержки модулем CPU парольной защиты:
 -  – модуль не поддерживает парольную защиту (доступ без авторизации);
 -  – модуль поддерживает парольную защиту (доступ через авторизацию);
 -  – в модуль передан верный пароль (повторный доступ без авторизации);
- Наименование типа модуля CPU;
- IP-адрес+шлюз для сетевого интерфейса в роли IP ([п. 2.3.2.1](#));
- IP-адрес+шлюз для сетевого интерфейса в роли Datalink IP ([п. 2.3.2.1](#));
- Серийный номер модуля CPU;
- Имя ресурса выполняемой в модуле CPU программы пользователя;
- Поддержка модулем CPU режима резервирования (Failover);
- Поддержка модулем CPU протокола Powerlink.



Устройство	IP адрес	Datalink адрес	Серийный №	Ресурс	Резерв	Powerlink
 MKLogic500 CPU 504	192.168.100.88 0.0.0.0	192.168.201.88 0.0.0.0	2100047	Application	Да	Да
 MKLogic500 CPU 501	192.168.100.98 192.168.100.1	192.168.10.158 0.0.0.0	1700013	STEND_DENIS_501_2\DEVIC	Да	Нет
 MKLogic500 CPU 504	192.168.100.89 0.0.0.0	192.168.201.89 0.0.0.0	1900006	Application	Да	Да
 MKLogic500 CPU 501	192.168.100.121 0.0.0.0	192.168.10.137 0.0.0.0	1700001	NIKITA\Device1	Да	Нет
 MKLogic500 CPU 501	192.168.100.56 0.0.0.0	192.168.10.107 0.0.0.0	1700006	---	Да	Нет
 MKLogic500 CM-P 545	192.168.1.1 0.0.0.0	192.168.10.235 0.0.0.0	1800001	---	Нет	Да
 MKLogic500 CPU 503	192.168.100.190 0.0.0.0	192.168.20.190 0.0.0.0	2100048	MK503_MK541_DATA_TEST	Да	Да

10.155.26.88 Поиск устройств Поиск по IP Подсветить Настройки Доступ по FTP Диагностика Обновить из FTP

MK500 IODevice Version 5.1.0.4

Рис. 2.8 – Окно плагина MK500 IODevice с найденными модулями CPU и CN

3) Если поиск не дал результатов, следует провести поиск модуля по IP-адресу. Для этого в окне плагина MK500 IODevice следует нажать кнопку «Поиск по IP», в открывшемся окне ввести IP-адрес порта ETH1 желаемого модуля и нажать кнопку «ОК» ([Рис. 2.9](#)). Все IP-адреса модуля CPU бегущей строкой показываются на панели индикации.

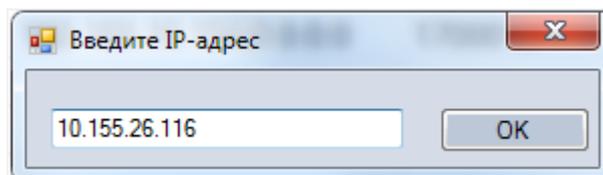


Рис. 2.9 – Окна ввода IP-адреса модуля CPU

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 скопировать из модуля CPU актуальные tdb-файлы или записать в модуль настроечные файлы либо файлы обновлений следует выбрать в таблице строку с модулем и нажать кнопку «Доступ по FTP».

Соединение с модулем CPU по FTP шифруется TLS-сертификатом, который следует принять при установлении первого подключения ([Рис. 2.10](#)).

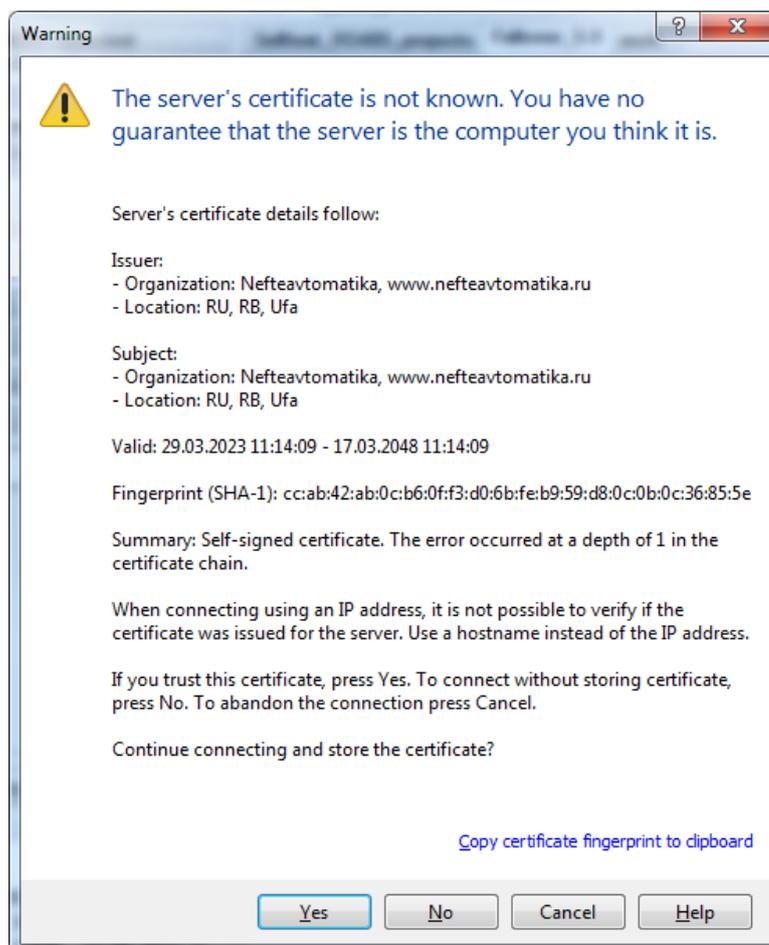


Рис. 2.10 – Окно приёма TLS-сертификата при подключении к CPU по FTPS

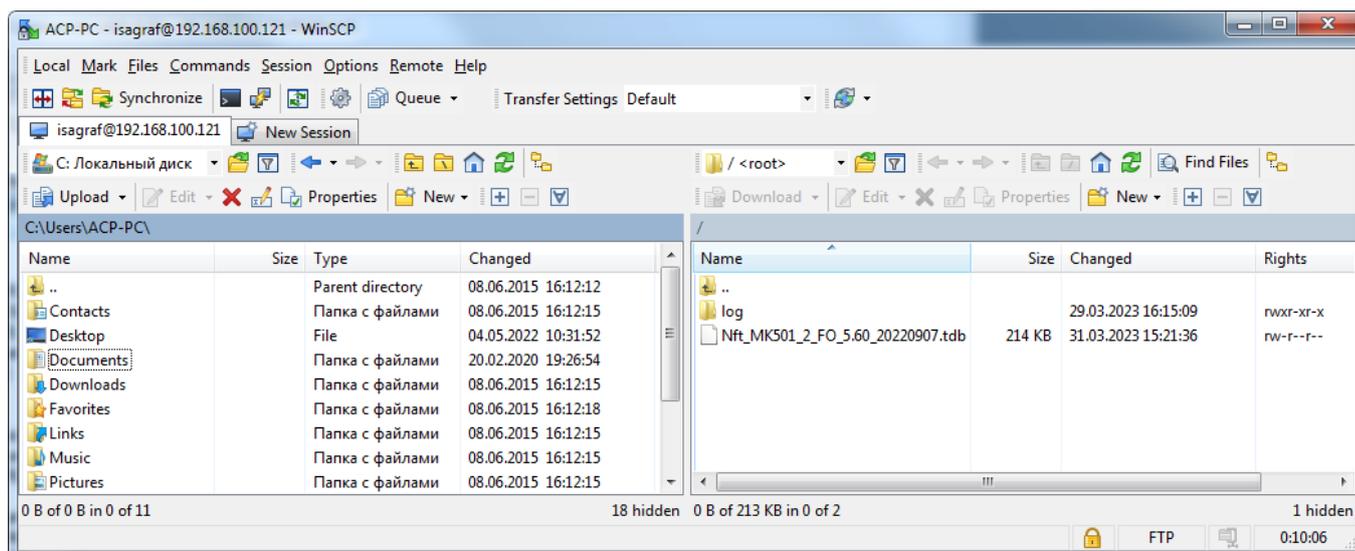


Рис. 2.11 – Окно WinSCP при обращении к FTP модуля CPU

В открывшемся окне приложения WinSCP можно найти актуальные tdb-файлы для данного модуля CPU, в него же можно скопировать необходимые файлы ([Рис. 2.11](#)).

⚠ ВНИМАНИЕ

Логин к папке FTP – «isagraf». Пароль на доступ к папке FTP по умолчанию – «isagraf»

⚠ ВНИМАНИЕ

Пользователю разрешено копировать в папку FTP файлы суммарным объёмом не более 50 МБ.

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.

Если модуль CPU защищён паролем и пароль не пустой, появится окно ввода пароля для доступа к модулю CPU ([Рис. 2.12](#)). Нажатие на символ  позволяет увидеть/скрыть вводимый пароль. При успешном вводе пароля последующего ввода пароля не требуется до перезапуска плагина MK500 IODevice или повторного нажатия кнопки «Поиск устройств». Исходно модули CPU поставляются без пароля.

Число попыток ввода пароля не ограничено.

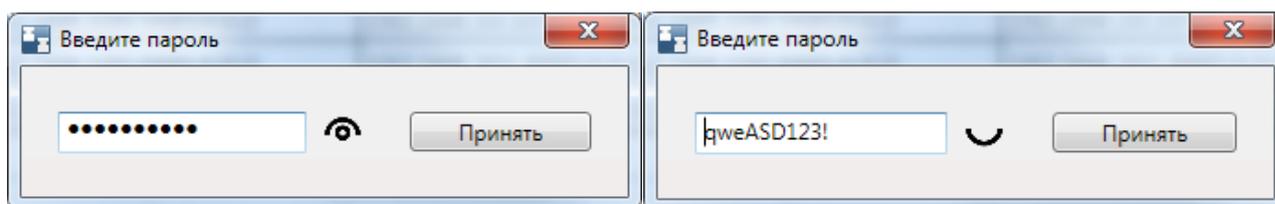


Рис. 2.12 – Окно ввода пароля доступа к модулю CPU

После ввода верного пароля будет выведено окно настройки параметров модуля CPU (см. [п.2.3.2](#)).

Если модуль CPU не поддерживает парольную защиту, или если авторизация к модулю уже выполнялась, окно настройки параметров модуля CPU будет открыто немедленно.

2.3.2 Настройка и диагностика модуля CPU с помощью плагина MK500 IODevice

Открывшееся окно настроек модуля CPU имеет 6 разделов, открывающихся при нажатии соответствующего пункта списка в левой части окна ([Рис. 2.13](#)). Заголовок окна содержит тип CPU и его серийный номер. Общими для всех разделов являются кнопки «Прочитать из CPU» и «Заккрыть» в нижнем правом углу окна. Кнопку «Прочитать из CPU» следует использовать для обновления значений в текущем разделе окна настроек.

Раздел «Сетевые настройки» предназначен для настройки сетевых интерфейсов и маршрутов модуля CPU для режимов работы Primary и Secondary, настройки списка «Белых IP-адресов» а также для назначения ролей сетевым интерфейсам ([п. 2.4](#)).

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 ISaGRAF 5.60, 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-503-120.

Раздел «Настройки безопасности» отображается только для модулей CPU, поддерживающих парольную авторизацию, и предназначен для настройки парольного доступа к модулю CPU, смены пароля по умолчанию для доступа к FTP, и управления режимом скрытия IP-адресов модуля CPU.

Ниже будут более подробно разобраны особенности работы во всех шести разделах.

2.3.2.1 Сетевые настройки

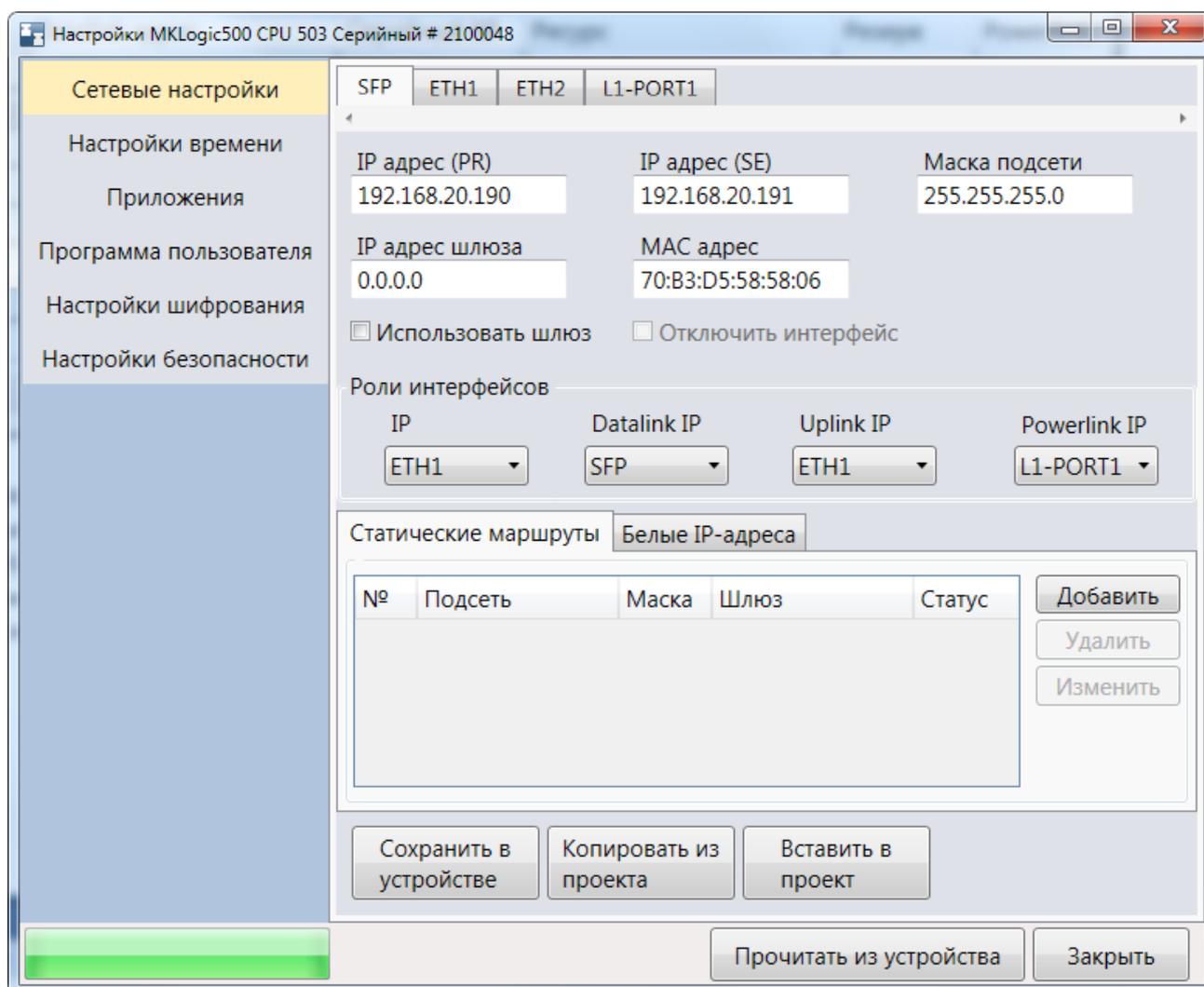


Рис. 2.13 – Окно сетевых настроек модуля CPU

В верхней части окна настроек модуля CPU в разделе «Сетевые настройки» (Рис. 2.13) 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 в позиции SE (все остальные варианты размещения). Два варианта настроек нужны для того, чтобы модуль CPU мог передать корректные сетевые настройки резервному модулю CPU без его предварительной настройки.

 **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.

Начиная с версии системного ПО 1.3.8.2 от 20.02.2023 можно отключать неиспользуемые сетевые интерфейсы. Отключенный интерфейс полностью исключается из работы операционной системой модуля CPU. На индикации модуля CPU для такого интерфейса вместо IP-адреса выводится «OFF».

Флаг отключения сетевого интерфейса находится рядом с флагом использования шлюза ([Рис. 2.13](#)). Собственно включение/отключение интерфейсов выполняется в ходе применения сетевых настроек. Нельзя отключать сетевые интерфейсы, задействованные в ролях IP и Datalink IP (см ниже).

Роли сетевых интерфейсов

В средней части окна расположены выпадающие списки для задания ролей сетевых интерфейсов модуля CPU ([Рис. 2.14](#)). Необходимо выбрать интерфейсы для роли IP-интерфейса (для связи со средой разработки АСР) и для роли Datalink IP-интерфейса (связь с резервным модулем CPU в режиме резервирования Failover).



Рис. 2.14 – Раздел настройки ролей интерфейсов модуля CPU

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). Также роль Uplink IP используется устройством ping_diag ([п. 3.7.9](#)). Роль Uplink IP является опциональной.

Роль Powerlink IP интерфейса реализована для CPU МК-503-120, CPU МК-504-120 и МК-545-010. Назначение сетевого интерфейса на роль Powerlink IP необходимо для корректной работы Powerlink ([п. 3.7.5](#)).

Статические маршруты

Под настройками ролей расположены вкладки настройки статических маршрутов и «белых IP-адресов» модуля CPU ([Рис. 2.15](#)).

Статические маршруты настраиваются сразу для всех сетевых интерфейсов модуля. При нажатии кнопки «Добавить» появляется окно настроек параметров маршрута ([Рис. 2.16](#)).

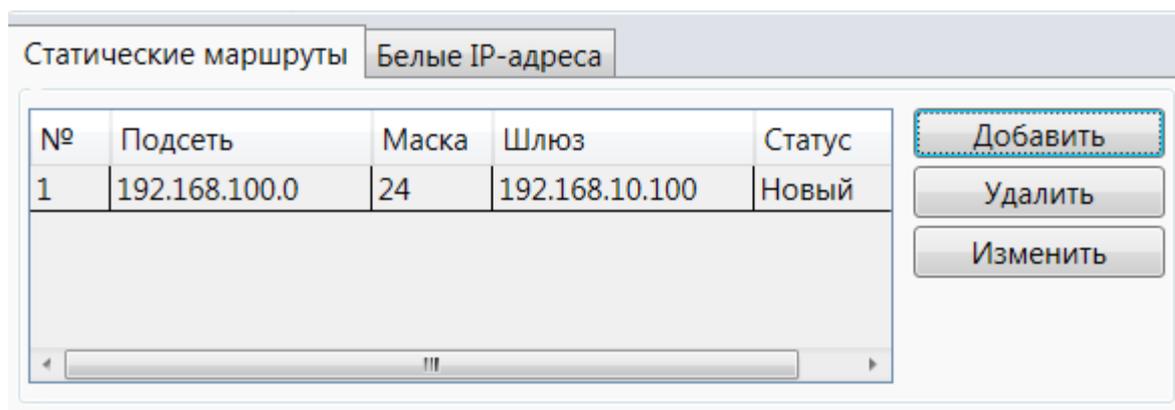


Рис. 2.15 – Вкладка настроек статических маршрутов модуля CPU

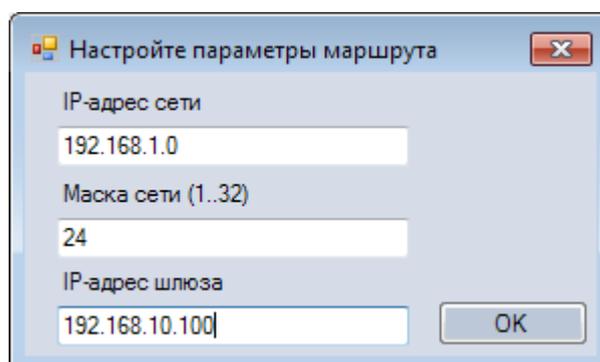


Рис. 2.16 – Окно настроек параметров маршрута модуля CPU

Следует отметить, что IP-адрес сети и маска сети в окне настройки параметра маршрута должны соответствовать друг другу, в IP-адресе все биты после маски должны быть равны 0.

Новые маршруты запишутся в модуль CPU при сохранении всех сетевых настроек.

Белые IP-адреса

В соседней с настройками маршрутов вкладке расположены настройки «Белых IP-адресов» ([Рис. 2.17](#)).

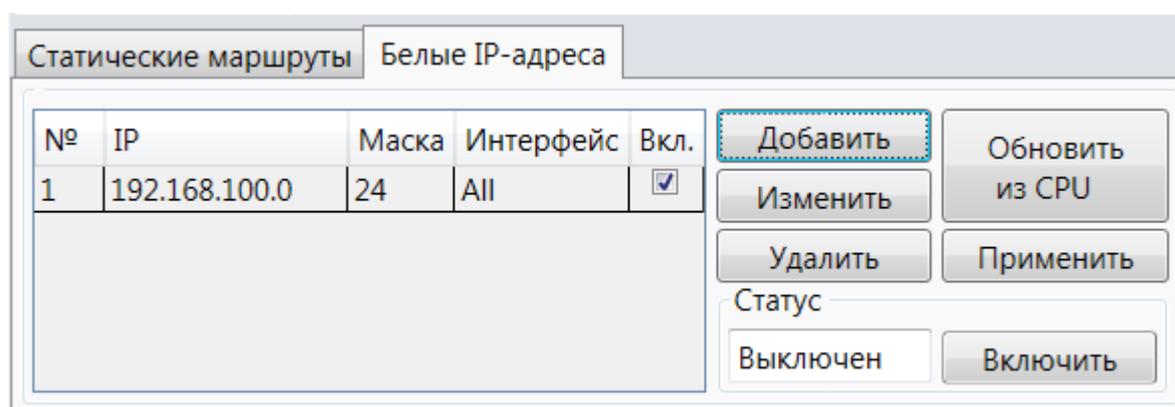


Рис. 2.17 – Вкладка настроек «белых IP-адресов» модуля CPU

При включённом режиме «Белых IP-адресов» встроенный сетевой фильтр модуля CPU пропускает TCP-пакеты только с тех адресов и с тех интерфейсов, которые добавлены в список. При этом пакеты канального уровня (Ethernet) не блокируются.

При нажатии кнопки «Добавить» появляется окно настройки «Белых IP-адресов» (Рис. 2.18). В нём можно выбрать IP-адрес или группу IP-адресов, с которой будет разрешено прохождение пакетов на один выбранный сетевой интерфейс или на все сетевые интерфейсы.

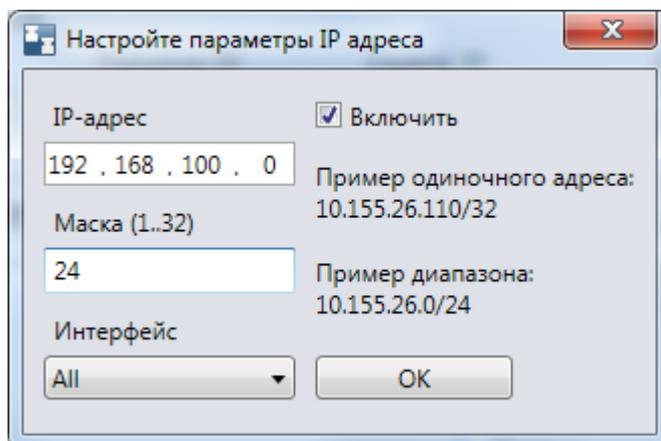


Рис. 2.18 – Окно настроек «белых IP-адресов» модуля CPU

Применить выбранные «Белые IP-адреса» и включить/выключить фильтрацию можно по нажатию кнопок «Применить» и «Включить/Выключить» соответственно. Сохранять при этом все сетевые настройки не обязательно.

Применение сетевых настроек к модулю CPU

В нижней части окна расположены настройки режима резервирования Failover (для модулей CPU с поддержкой этого режима) и кнопки «Сохранить в устройстве», «Копировать из проекта» и «Вставить в проект» (Рис. 2.19).

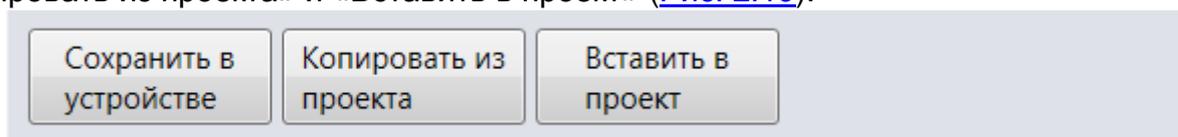


Рис. 2.19 – Кнопки применения сетевых настроек

Кнопка «Копировать из проекта» копирует текущие сетевые настройки проекта в поля IP-адреса интерфейса в роли IP и в поля настроек режима резервирования Failover, что упрощает конфигурацию модуля CPU при уже настроенном проекте.

Кнопка «Вставить в проект» копирует значения поля IP-адреса интерфейса в роли IP и настройки режима резервирования Failover в сетевую конфигурацию проекта, что может быть полезным при адаптации уже существующих проектов под новые сетевые настройки.

Кнопка «Сохранить в устройстве» служит для применения введённых 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 1.2](#)). Поэтому для успешного применения сетевых настроек следует отключить второй модуль CPU.

2.3.2.2 Настройки времени

В верхней части окна настроек модуля CPU в разделе «Настройки времени» (Рис. 2.20) расположены поля ввода и кнопки для настройки временных параметров модуля CPU.

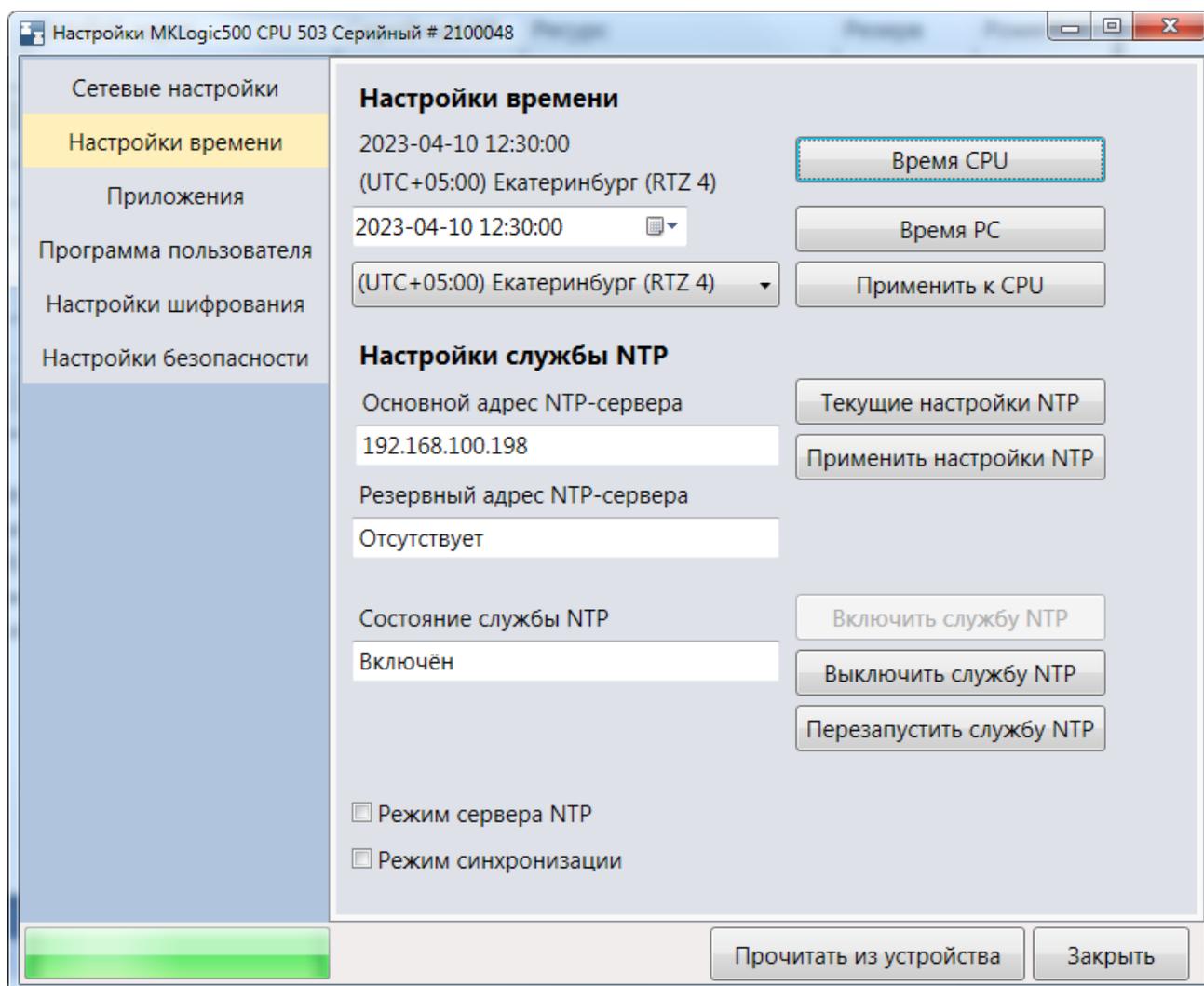


Рис. 2.20 – Окно настроек времени модуля CPU

Текущее 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 установлена среда разработки АСР. 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 службы NTP модуля CPU и кнопки управления службой NTP. Там же расположено поле статуса службы NTP и флаги включения режима сервера NTP и режима синхронизации.

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.

В режиме NTP-сервера модуль CPU является NTP-сервером для сторонних NTP-клиентов, но только в том случае, если сам получает точное время со своих NTP-серверов. При этом значение *stratum* у NTP-сервера на модуле CPU на единицу хуже, чем у исходного NTP-сервера.

Режим синхронизации позволяет резервному CPU синхронизировать своё время по основному CPU. При включённом режиме синхронизации в настройки службы NTP резервного модуля CPU автоматически добавляется IP-адрес интерфейса в роли IP Datalink на основном CPU. При потере резервным модулем связи с NTP-серверами служба NTP автоматически переключается на ведущий CPU как источник времени.

Для работы режим синхронизации требует включённого режима NTP-сервера.

2.3.2.3 Приложения

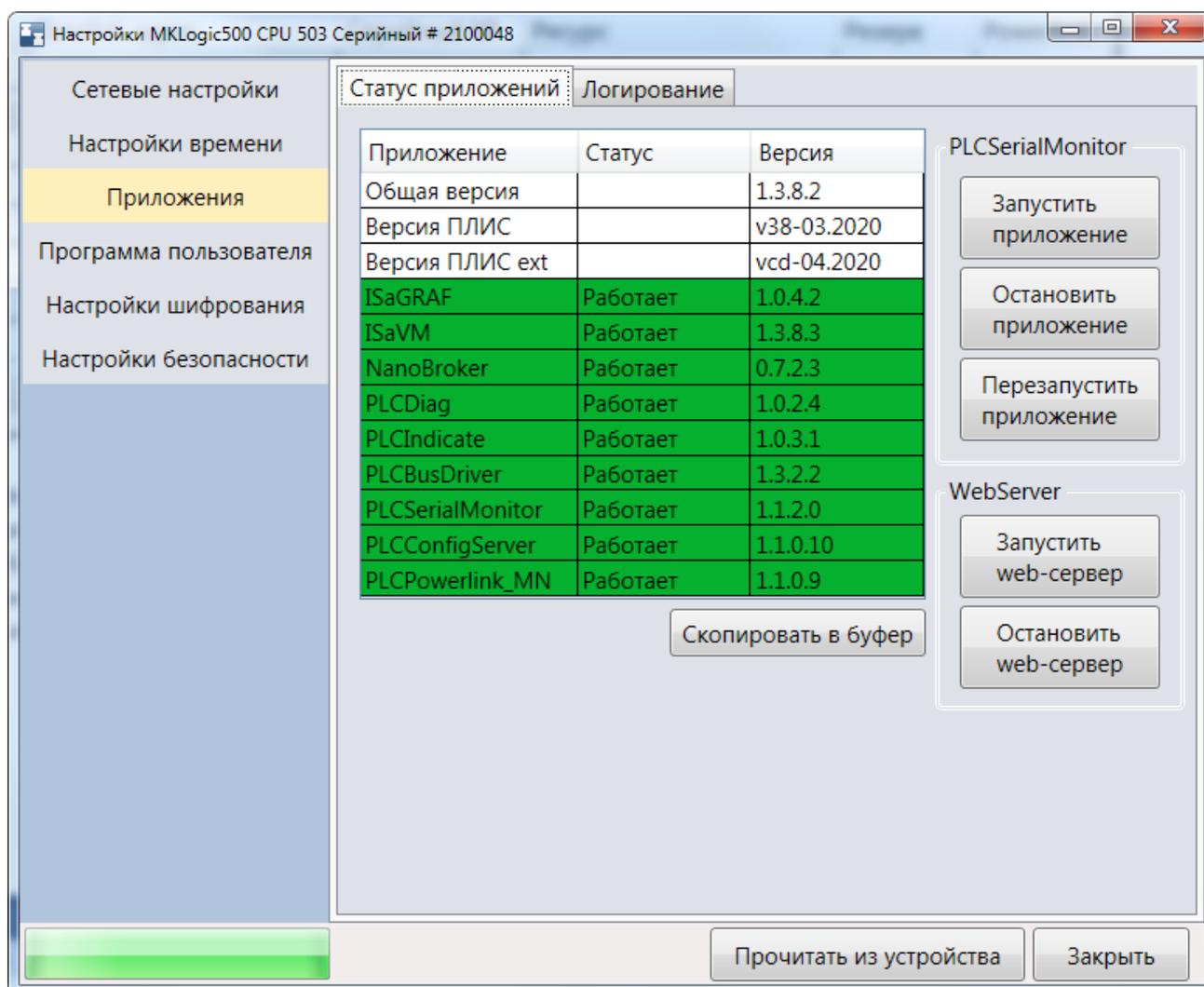


Рис. 2.21 – Окно статуса приложения модуля CPU

В верхней части окна настроек модуля CPU в разделе «Статус приложений» (Рис. 2.21) 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-503-120, МК-504-120 и МК-545-010 в этом списке также указываются версии прошивок микросхем ПЛИС.

Вся эта информация служит для диагностических целей, и должна по запросу передаваться сервисным службам организации-производителю изделия.

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).

2.3.2.4 Логирование во внешнее приложение

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

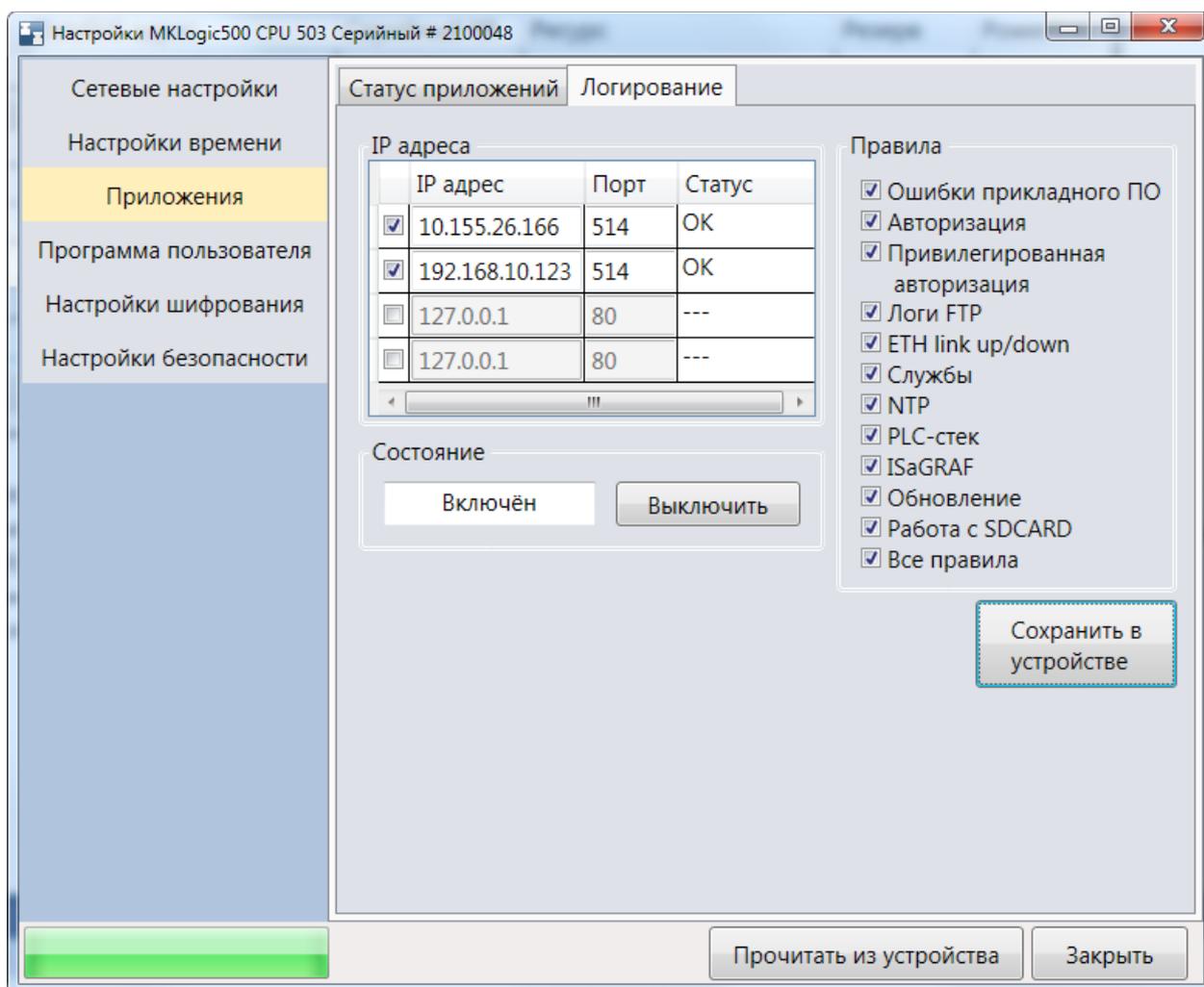


Рис. 2.22 – Окно настройки логирования системных сообщений

В верхней части вкладки «Логирование» в разделе «Приложения» (Рис. 2.22) 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.

2.3.2.5 Программа пользователя

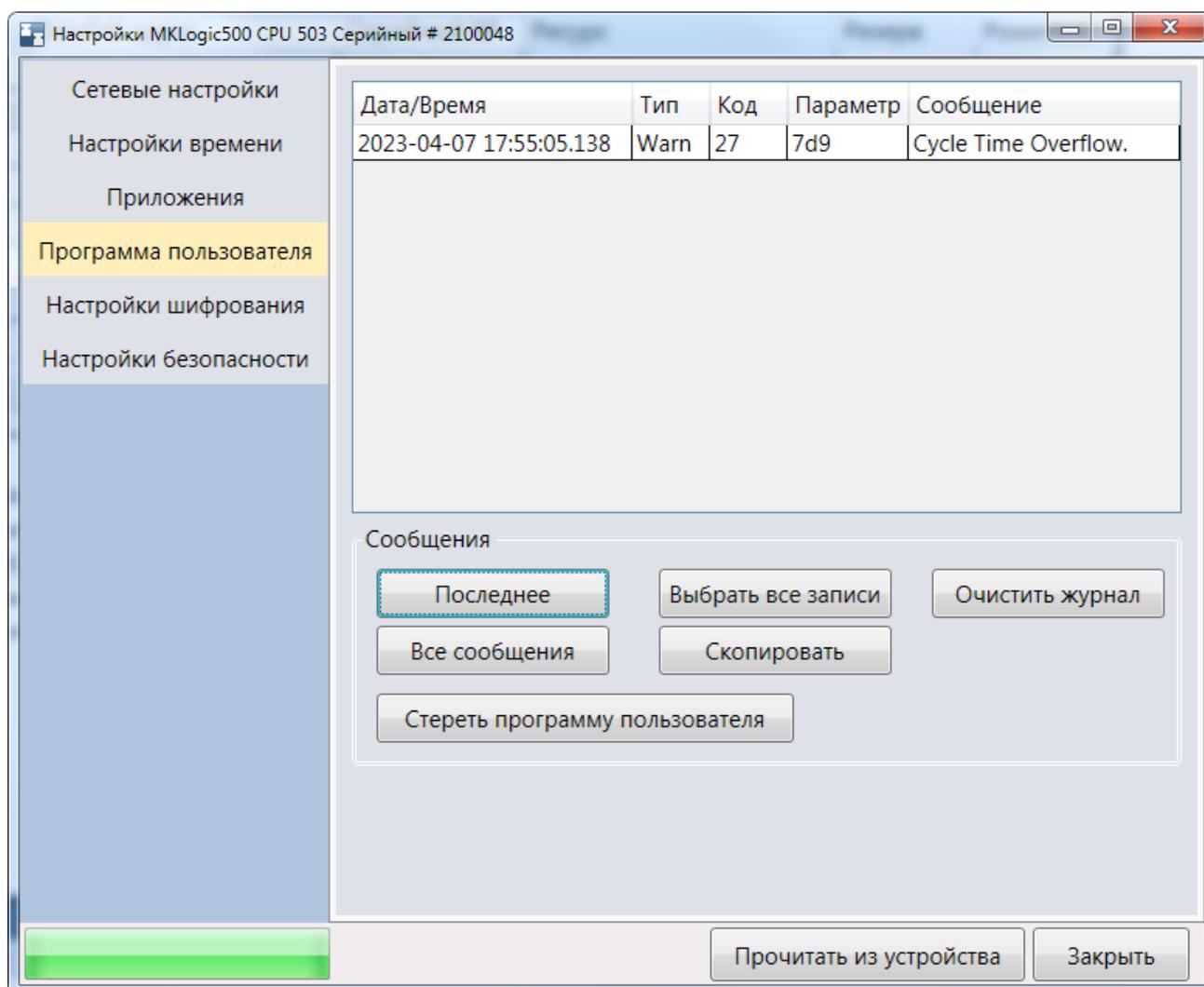


Рис. 2.23 – Окно программы пользователя модуля CPU

В верхней части окна настроек модуля CPU в разделе «Программа пользователя» ([Рис. 2.23](#)) расположен журнал сообщений программы пользователя. По умолчанию при открытии окна показывается только самое последнее сообщение журнала. Расшифровка сообщений приводится в [разделе 5.3](#).

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 (см. [п. 2.7](#)).

2.3.2.6 Настройки шифрования

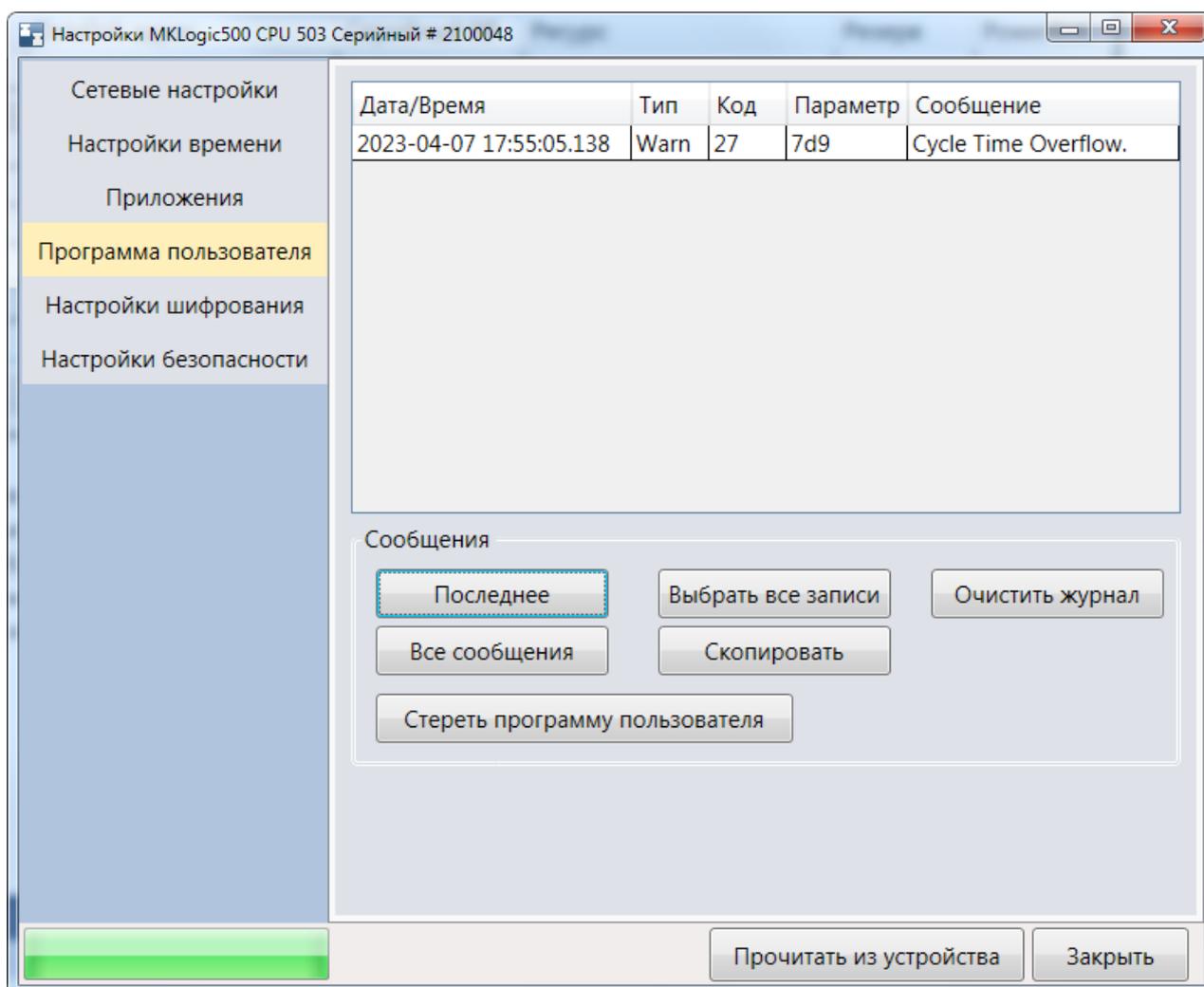


Рис. 2.24 – Окно настроек шифрования модуля CPU

Модули CPU 503 (МК-503-120) поддерживают шифрование обмена по протоколу OPC UA по стандарту X.509 (п. 3.7.4). Настроить параметры шифрования OPC UA можно в разделе «Настройки шифрования».

В верхней части окна настроек модуля CPU в разделе «Настройки шифрования» (Рис. 2.24) 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 `opcua_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 2.1](#).

```
# 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 -crl days 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 2.1 – 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.

2.3.2.7 Настройки безопасности

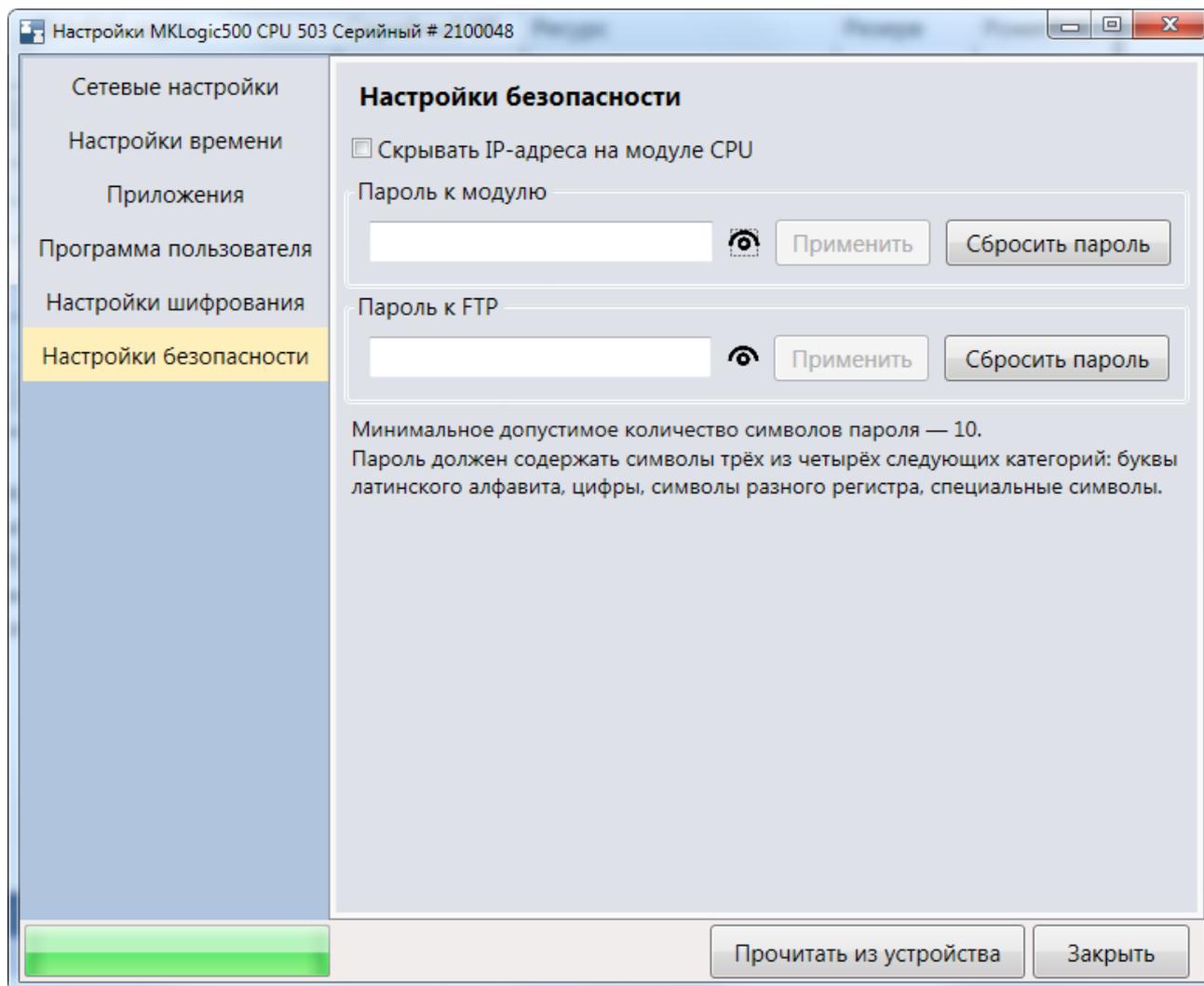


Рис. 2.25 – Окно настроек безопасности модуля CPU

Начиная с версии системного ПО 1.3.8.2 от 20.02.2023 (также требуется версия плагина MK500 IODevice не ниже 5.1.0.4) модулями CPU поддерживается парольная защита функций настройки, диагностики, доступа и обновления по FTP. Также поддерживается смена пароля по умолчанию на доступ к FTP, а также скрывание IP-адресов модуля CPU на его панели индикации.

Управление паролями выполняется в разделе «Настройки безопасности» ([Рис. 2.25](#)).

Требования к паролю описаны в центральной части рабочего окна. Кнопка «Применить к CPU» станет доступна после ввода корректного пароля. Нажатие на символ  позволяет увидеть/скрыть вводимый пароль.

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

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 ([см. п.2.3.1](#)).

В случае, если пароль забыт, следует обратиться в организацию-разработчик модулей CPU (ОП ИПЦ АО «Нефтеавтоматика») за получением инструкции по сбросу пароля.

Аналогичным образом в разделе «Настройки безопасности» можно сменить пароль на доступ к FTP. Логин к FTP изменить нельзя. При сбросе пароля на доступ к FTP пароль сбрасывается в значение по умолчанию.

**ВНИМАНИЕ**

Логин к папке FTP – «isagraf». Пароль на доступ к папке FTP по умолчанию – «isagraf»

При сокрытии сетевых интерфейсов на индикации модуля CPU вместо IP-адресов интерфейсов выводятся «- . - . -».

Режим скрытия адресов IP на панели индикации CPU включается и выключается соответствующим флагом на окне настроек. При этом режим применяется немедленно при установке/снятии флага.

2.4 Настройка сетевых параметров проекта

При работе с модулем CPU с поддержкой режима Failover сетевые настройки проекта включают в себя IP-адрес интерфейса для связи со средой разработки АСР, а также IP-адреса IP-интерфейса и Datalink IP-интерфейса основного и резервного модулей CPU (Primary и Secondary Device соответственно).

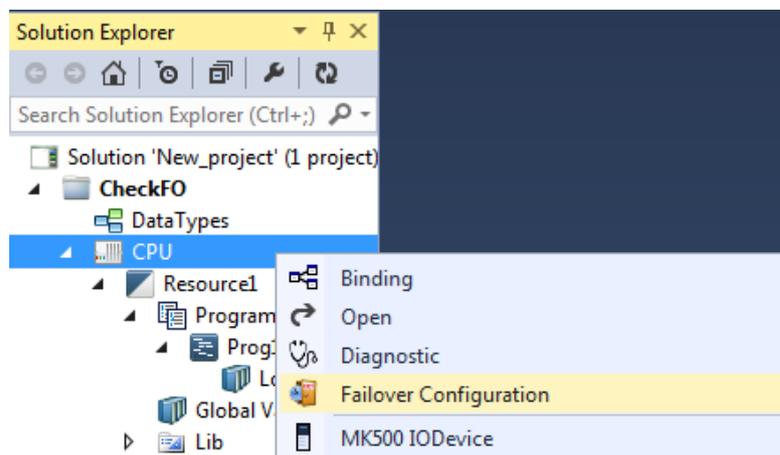


Рис. 2.26 – Вызов окна Failover Configuration

Для настройки IP-адресов следует открыть окно «Failover Configuration», выбрав в дереве проекта устройство и в его контекстном меню выбрав пункт «Failover Configuration» (Рис. 2.26).

В открывшемся окне (Рис. 2.27) следует ввести IP-адреса для основного и для резервного модулей CPU, а также IP-адрес интерфейса для связи со средой разработки АСР (обычно вводится IP-адрес, совпадающий с `FailoverPrimaryIP`).

Также в окне «Failover Configuration» следует включить режим Failover (ввести TRUE в поле `EnableFailover`), а также настроить параметры таймаутов режима Failover. Параметр `FailoverHeartbeatTimeoutMs` определяет задержку перехода с основного на резервный модуль CPU при отсутствии связи между основным и резервными модулями CPU по интерфейсу Datalink IP.

Параметр `FailoverHeartbeatDeactivationTimeMs` определяет время, в течение которого модули CPU решают, кто из них будет основным, а кто – резервным (обычно это происходит при потере и восстановлении связи по интерфейсу Datalink IP).



ВНИМАНИЕ

Значения таймаутов по умолчанию в новом проекте (2000 и 10000) всегда следует уменьшать минимум в 10 раз. Оптимальные значения таймаутов – 100...200 мс для `FailoverHeartbeatTimeoutMs` и 500...1000 для `FailoverHeartbeatDeactivationTimeMs`. Также следует уменьшить до 100...200 мс параметр `FailoverDatalinkTimeoutMs` из дополнительных настроек ресурса (п. 2.5.2, секция Info).

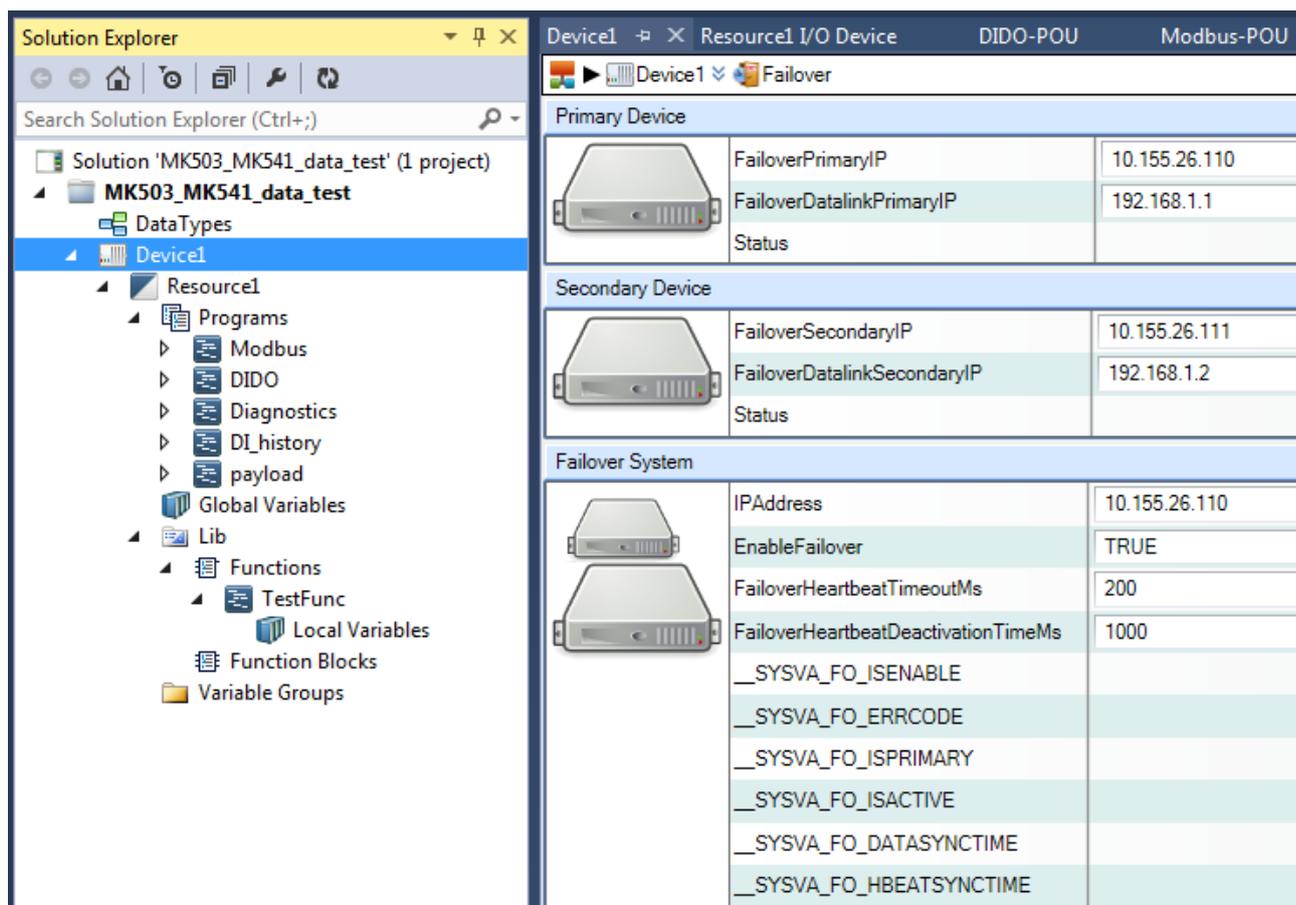


Рис. 2.27 – Настройка сетевых параметров проекта с поддержкой режима Failover из окна Failover Configuration

Если модуль CPU работает без резерва и поддержка режима Failover не нужна, рекомендуется выключить её (ввести FALSE в поле EnableFailover). В этом режиме работы модуль CPU не будет искать резервный модуль CPU, и не будет диагностировать ошибку работы в режиме Failover.

Альтернативный вариант задания IP-адресов проекта (путём копирования из модуля CPU) был описан в [п. 2.3.2.1](#).

Также следует отметить то, что при старте модулей CPU происходит сравнение их сетевых настроек, в котором стартовавший первым модуль CPU применяет свои сетевые настройки в режиме Secondary на стартовавший вторым модуль CPU ([п.1.2](#)).

Для применения сетевых настроек проекта следует его пересобрать, лучше всего с предварительной очисткой проекта. Для этого следует последовательно выполнить команды BUILD→Clean Solution и BUILD→Build Solution, либо BUILD→Rebuild Solution.

2.5 Структура проекта

В рамках среды исполнения ISaGRAF проект (Solution) создаётся как коллекция для устройств ([п. 2.5.1](#)), имеющих общую аппаратную платформу и среду исполнения, и описываемых общим tdb-файлом.

Также общими для всех устройств являются типы данных (массивы, структуры) и константы. Ознакомиться с имеющимися типами данных, а также добавить свои типы данных, можно выбрав в дереве проектов раздел DataTypes.

Tdb-файл содержит в себе описание всех модулей изделия, описание дополнительных констант, типов, функций и функциональных блоков, а также настройки и ограничения создаваемого с его помощью проекта. Каждый модуль CPU содержит в себе актуальный tdb-файл, который можно скопировать из модуля CPU по FTP ([п. 2.3.1](#)).

Tdb-файл в модуле CPU может обновиться после обновления системного ПО модуля CPU.

2.5.1 Устройства (Device) проекта

В рамках среды исполнения ISaGRAF устройством называется представление оборудования (ПЛК), выполняющего виртуальные машины IsaVM (п. 2.5.2).

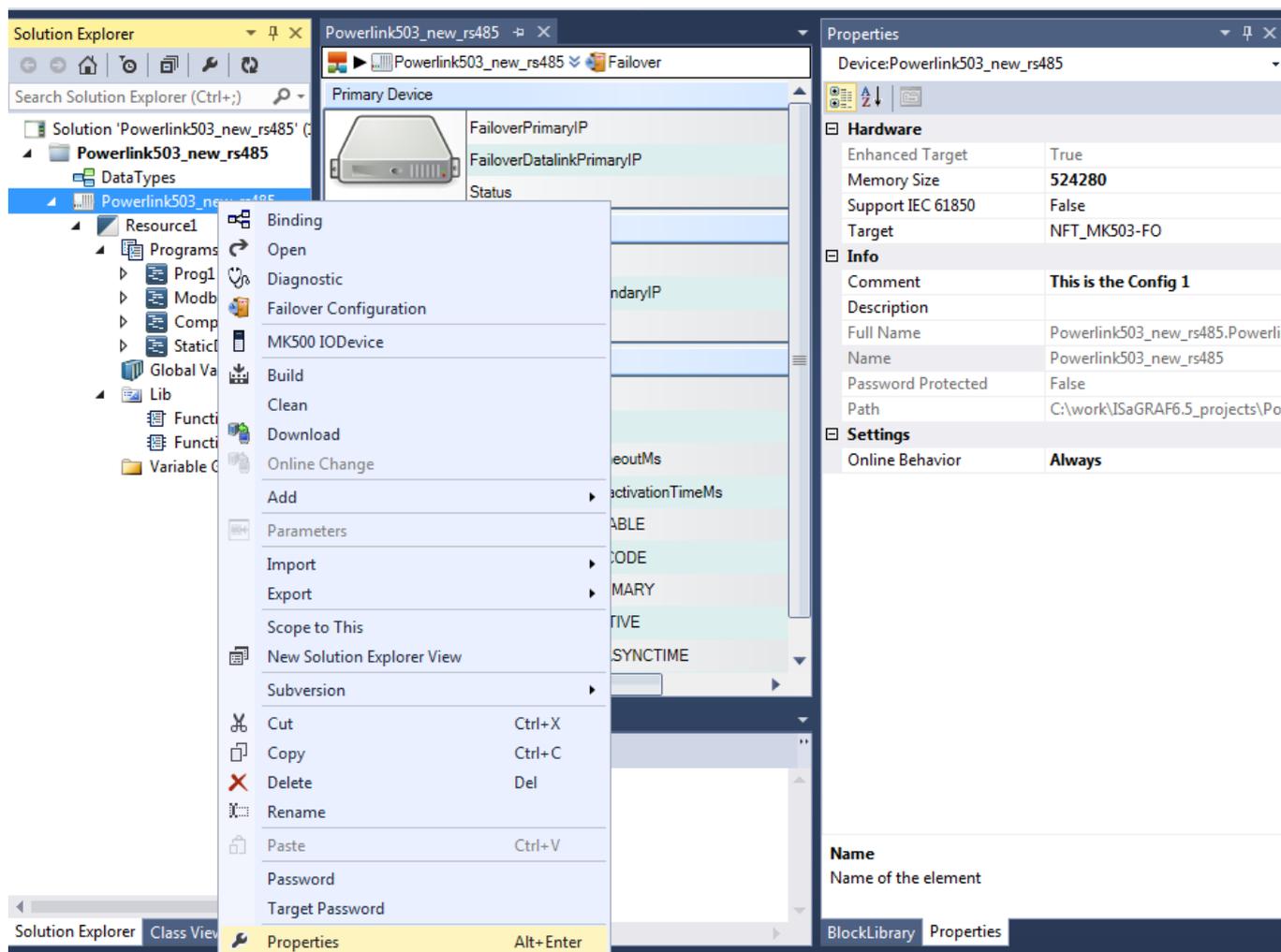


Рис. 2.28 – Настройка параметров устройства

На уровне устройства определяются сетевые параметры (п. 2.4), а также размер оперативной памяти в модуле CPU изделия, отводимый для хранения пользовательских переменных.

Для настройки размера оперативной памяти в модуле CPU для хранения пользовательских переменных следует открыть окно свойств устройства. Для этого необходимо выбрать в дереве проектов интересующее нас устройство и в его контекстном меню выбрать пункт Properties (Рис. 2.28). Далее в окне Properties следует изменить (обычно требуется увеличить) значение Memory Size, по умолчанию равное 524280.

2.5.2 Ресурсы (Resource) проекта

В рамках среды исполнения ISaGRAF ресурсом называется совокупность программных модулей и определений, составляющая виртуальную машину IsaVM. Виртуальные машины IsaVM выполняются в модуле CPU как отдельные процессы под общим контролем и управлением среды исполнения ISaGRAF, и фактически позволяют параллельно выполнять несколько разных независимых программ пользователя.

Каждый ресурс содержит в себе конфигурацию устройств ввода-вывода, переменные и программные модули, не пересекающиеся с конфигурациями, переменными и программными модулями других ресурсов.



ВНИМАНИЕ

В модулях CPU настоящего изделия разрешено выполнять только один ресурс, поэтому создание дополнительных ресурсов в проектах, предназначенных для работы с модулями CPU изделия, настоятельно не рекомендуется. Проекты с более чем одним ресурсом, не будут нормально запускаться в модулях CPU изделия.

Добавление, настройка и удаление модулей ввода-вывода в ресурс будут рассмотрены в разделе [3.1](#).

Для настройки параметров ресурса следует выбрать в дереве проектов интересующий нас ресурс и в его контекстном меню выбрать пункт «Properties» в открывшемся окне «Properties» ([Рис. 2.29](#), [Рис. 2.31](#)). Далее будут перечислены самые значимые параметры ресурса и даны рекомендации по их настройке.

Секция **Code** ([Рис. 2.29](#)):

- Code for simulation – отвечает за сборку версии проекта для запуска на локальном симуляторе. В рабочем проекте настоятельно рекомендуется отключать (false), это ускорит сборку и позволит избежать сообщений компилятора о нехватке памяти при использовании в проекте больших объёмов пользовательских данных (у симулятора небольшой объём памяти для хранения пользовательских данных).
- Check Array Index – отвечает за добавление в генерируемый код проверок на выход за пределы массивов. Если установить в true, при выходе за пределы массива рабочая программа останавливается с ошибкой «21. Kernel TIC: Boundary check error.» (см. [раздел 5.3](#)).
- Dump Configuration Files – отвечает за генерацию отладочных файлов конфигурации ресурса. Если установить его в true, то при сборке проекта в папке ресурса будут созданы файлы-дампы конфигурации ресурса с именами вида RESOURCE1_Conf.ttc, RESOURCE1_Constants.ttc и RESOURCE1_DwlOrder.ttc.
- Dump Network – отвечает за генерацию отладочных файлов сетевой конфигурации проекта. Если установить его в true, то при сборке проекта в папке проекта будет создан файл-дамп сетевой конфигурации NetworkConf.ttc.
- Dump POU Files – отвечает за генерацию отладочных файлов POU (программных модулей) ресурса. Если установить его в true, то при сборке проекта в папке ресурса будут созданы файлы-дампы программных модулей ресурса с именами вида RESOURCE1_Pou_PROG1.ttc. Число создаваемых

- файлов определяется числом программных модулей (программы, функции, функциональные блоки) в составе ресурса.
- Enable Code Optimization – отвечает за оптимизацию результирующего TIC-кода по скорости. Рекомендуется установить в true, так как его включение приводит к существенному (до 2 раз) ускорению времени исполнения программы пользователя.
 - Function Internal State Enable – отвечает за сохранение функциями значений своих локальных переменных между вызовами. Подробнее см. [п. 5.1.2.](#)
 - Generate Map File – отвечает за генерацию отладочных файлов пользовательских и системных переменных ресурса. Если установить его в true, то при сборке проекта в папке ресурса будут созданы файлы-дампы переменных с именами RESOURCE1_SymbolsDebug.ttc и RESOURCE1_SymbolsTarget.ttc.
 - Indirect Bit Access Validation – отвечает за добавление в генерируемый код проверок на ошибки при косвенной адресации отдельных битов переменной. Если установить в true, при выходе за пределы переменной рабочая программа останавливается с ошибкой.

Code	
Code For Simulation	True
Compiler Options	
Check Array Index	False
Dump Configuration Files	False
Dump Network	False
Dump POU Files	False
Enable Code Optimization	False
Function Internal State Enable	True
Generate Map File	False
Indirect Bit Access Validation	True
Reduce Boolean Expression Evaluation	False
Target Supports Optimized TIC Code	False
Embed Symbol Table	True
Embedded Table Type	Complete
Embedded Zip Source	None
Structured C Source Code	False
TIC Code	True
Custom Fields	
TDB Version	2019-04-01
Hardware	
Target	NFT_MK501_2-FO
Info	
Comment	Resource Number 1
Description	
Extended Parameters	(Collection)
Full Name	New_project.Device1.Resource1
Name	Resource1
Number	1
Password Protected	False
Path	C:\work\ISaGRAF6.5_projects\New_pr
Task Name	Task

Рис. 2.29 – Настройка параметров ресурса

- Reduce Boolean Expression Evaluation – отвечает за сокращённое вычисление булевых выражений. Если установить его в true, то булевы выражения будут вычисляться не полностью, а только до момента, когда становится очевидным конечный результат. Рекомендуется установить значение в true.
- Target Supports Optimized TIC Code – отвечает за поддержку оптимизированного TIC-кода средой исполнения. Рекомендуется установить значение в true.
- Embedded Symbol Table – отвечает за встраивание таблицы символов в проект. Всегда принимает значение true, изменение запрещено на уровне среды разработки.
- Embedded Table Type – отвечает за объём встраиваемой таблицы символов в проект. Всегда принимает значение Complete, изменение запрещено на уровне среды разработки.
- Embedded Zip Source – отвечает за генерацию файла--архива ресурса, устройства или всего проекта (в зависимости от значения). Если установить его в Project, Device или Resource, то при сборке проекта в папке ресурса будет создан файл IDS00103, представляющий собой 7z-архив проекта, устройства или ресурса. В рабочем проекте настоятельно не рекомендуется использовать для этого параметра значение, отличное от None, так как среда разработки в ходе загрузки проекта в модуль CPU устройства пытается загрузить и этот файл, что в версии среды разработки ACP 6.50 всегда заканчивается неудачей.
- Structured C Source Code – отвечает за генерацию проекта в программу на языке C. Если установить его в true, то при сборке проекта в папке ресурса будет создан набор файлов с расширениями *.h и *.c, которые теоретически можно скомпилировать непосредственно в машинный код и запустить в модуле CPU изделия. Значение данного параметра всегда противоположно значению параметра TIC Code. Рекомендуется установить значение параметра в false.
- TIC Code – отвечает за генерацию проекта в исполняемый средой isagraf TIC-код. Если установить его в true, то при сборке проекта в папке ресурса будет создан набор файлов, предназначенных для загрузки и исполнения в модуле CPU изделия. Значение данного параметра всегда противоположно значению параметра Structured C Source Code. Для нормальной работы настоятельно рекомендуется установить значение параметра в true.

Секция **Hardware** ([Рис. 2.29](#)):

- Target – отвечает за тип целевого устройства, для которого собирается ресурс. Для работы с модулями CPU изделия должен быть установлен в «NFT_MK501_2-FO» или «NFT_MK503-FO».

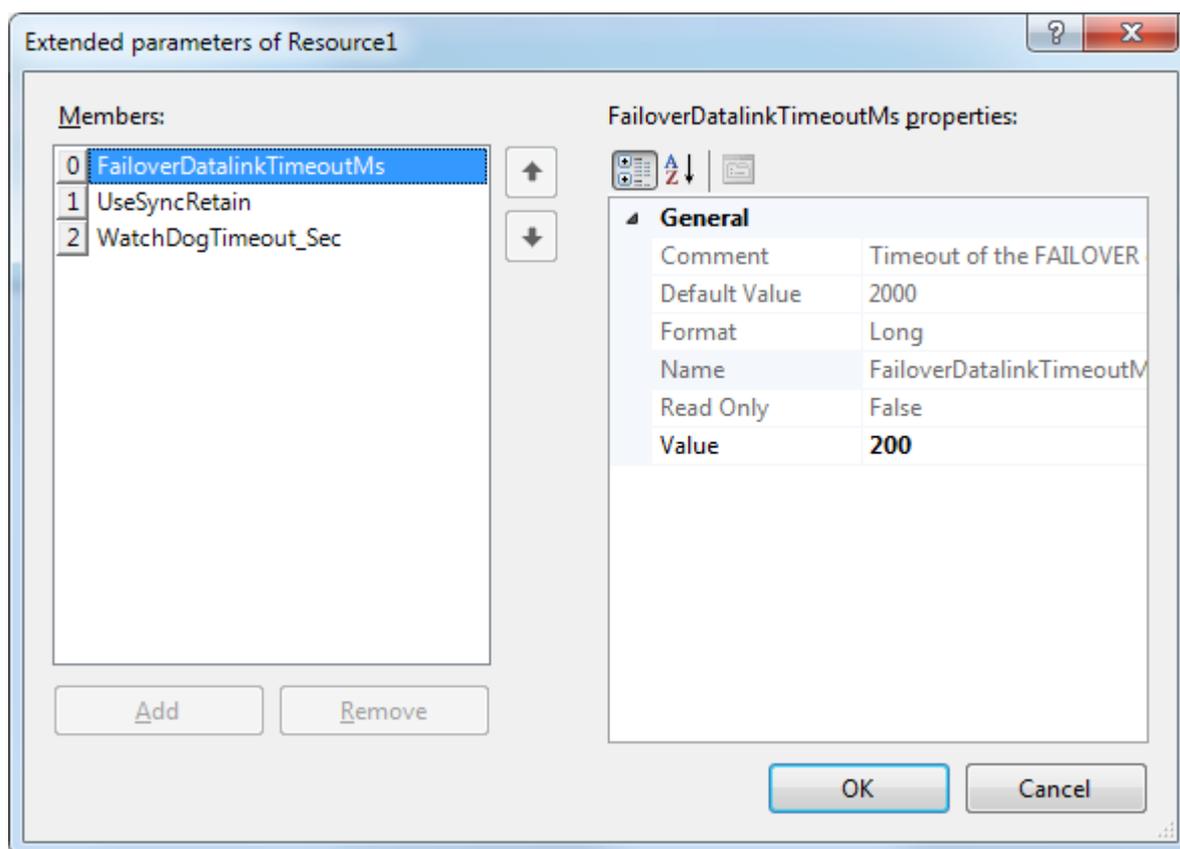


Рис. 2.30 – Настройка дополнительных параметров ресурса

Секция **Info** ([Рис. 2.29](#)):

- Comment – текстовый комментарий к ресурсу.
- Number – порядковый номер ресурса.
- Extended Parameters – дополнительные параметры ресурса. При нажатии на кнопку редактирования появляется окно настройки дополнительных параметров ресурса ([Рис. 2.30](#)), в котором можно настроить три параметра:
 - FailoverDatalinkTimeoutMs - определяет задержку перехода с основного на резервный модуль CPU при таймауте между основным и резервным модулями CPU по интерфейсу Datalink IP. Значение по умолчанию (2000 мс) настоятельно рекомендуется уменьшить до 100..200 мс.
 - UseSyncRetain – включает режим принудительного сброса кэша постоянной памяти модуля CPU после каждого обновления Retain-переменной. Существенно замедляет скорость работы, рекомендуется к использованию только в особых случаях. По умолчанию выключен.
 - WatchDogTimeout_Sec – определяет длительность сторожевого таймера, защищающего программу пользователя от закливания. Если в течении периода сторожевого таймера программа пользователя не закончит цикл и не начнёт следующий, произойдёт переход в режим принудительного останова программы пользователя с ожиданием рестарта/обновлением программы пользователя (подробнее см. [п. 5.4.2](#)). По умолчанию равен 4 секундам.

Memory Size for Online Changes	
Code Size	14000
Maximum Extra POUs	20
SFC States Mem size	4096
User Variable Size	2048
Memory Usage Info	
Biggest Free User Variable Me	2048
Data Space Usage	9868
Memory Usage (Code)	200
Memory Usage (Compressed	
Memory Usage (Data)	536200
Memory Usage (Retain Space)	0
Memory Usage (Symbolic Tab	17086
Memory Usage (Temporary V	4
User Variable Data Space Incre	0
Settings	
Cycle Time	100
Cycle Time Units	ms
Detect Errors	True
Execution Mode	Real Time
Memory For Retain	RETAIN
Nb Stored Errors	16
Online Behavior	Always
Trigger Cycles	True
SFC Dynamic Behavior Limits	
Gain Factor	8
Offset Factor	18

Рис. 2.31 – Настройка параметров ресурса

Секция **Memory Size for Online Changes** (Рис. 2.31):

- Code Size – объём памяти (в байтах), который выделяется при онлайн-обновлении секции кода. Оптимально увеличить исходное значение в 10 раз, чтобы избежать проблем с обновлением проекта без остановки (Online change);
- Maximum Extra POUs – максимальное число программных модулей, которые могут быть добавлены при онлайн-обновлении. Оптимально увеличить исходное значение в 10 раз, чтобы избежать проблем с обновлением проекта без остановки (Online change);
- SFC States Mem Size – объём памяти (в байтах), который выделяется при онлайн-обновлении программных модулей на языке SFC. Оптимально увеличить исходное значение в 10 раз, чтобы избежать проблем с обновлением проекта без остановки (Online change);
- User Variable Size – объём памяти (в байтах), который выделяется при онлайн-обновлении переменных. Оптимально увеличить исходное значение в 10 раз, чтобы избежать проблем с обновлением проекта без остановки (Online change);

Секция **Settings** (Рис. 2.31):

- Cycle Time – заданное время цикла выполнения программы пользователя, в единицах Cycle Time Units.



ВНИМАНИЕ

Установка заданного времени цикла, равного или меньшего реального времени выполнения программы пользователя, приводит к существенному росту загрузки процессора модуля CPU изделия.

- Cycle Time Units – единицы времени Cycle Time, для модулей CPU изделия могут принимать только значение ms (миллисекунды).
- Detect Errors – отвечает за хранение ошибок средой исполнения.
- Execution Mode – указывает на режим выполнения программы пользователя. Для непрерывной работы (нормальный режим работы модуля CPU) должен принимать значение «Real Time», для пошагового выполнения программы пользователя цикл за циклом должен принимать значение «Cycle to cycle».
- Memory For Retain – указывает на место хранения сохраняемых переменных. Для работы с модулями CPU изделия должен быть установлен в «RETAIN».
- Nb Stored Errors – число ошибок, которые хранятся в среде исполнения при параметре Detect Errors равном true.
- Online Behavior – указывает на режим работы программы пользователя (Design, Simulator, Online или Always, то есть в любом режиме), в котором допускается подключение к программе пользователя отладчиком («DEBUG»→«Start Debugging»). Рекомендуется установить в Always.
- Trigger Cycles – указывает на режим выполнения программы пользователя. Если параметр установлен в false, то следующий цикл выполнения программы пользователя стартует немедленно после завершения предыдущего цикла; если параметр установлен в true, то если реальное время выполнения программы пользователя меньше значения Cycle Time, очередной цикл выполнения программы будет запущен только по истечении Cycle Time. В большинстве случаев этот параметр должен быть установлен в true.

Секция **SFC Dynamic Behavior Limits** ([Рис. 2.31](#)):

- Gain Factor и Offset Factor – параметры, определяющие объём выделяемой памяти при добавлении одного программного модуля на языке SFC.

2.6 Сборка проекта

Сборка проекта (Solution) выполняется с помощью команд BUILD→Build Solution. Собрать отдельный программный блок либо устройство можно, выделив его в дереве проекта и выполнив команду BUILD→Build Selection.

В случае внесения изменений в конфигурацию ввода-вывода проекта (см. п. 3.1) либо в сетевые параметры проекта настоятельно рекомендуется перед сборкой выполнить команду очистки BUILD→Clean Solution либо выполнить команду BUILD→Rebuild Solution.

Если в ходе сборки возникли ошибки, они будут выведены в окно Error List (Рис. 2.32). Проект при этом собран не будет.

Следует отметить, что компилятор среды разработки ACP Workbench 6.5 не всегда указывает на одну лишь первопричину ошибки, что может затруднить локализацию проблемы. В примере, приведённом на Рис. 2.32, синтаксическая ошибка (лишние символы `aaa` перед идентификатором массива) порождает сразу 4 ошибки сборки, из которых следует обратить внимание только на самую первую («AAA: undeclared identifier»). С её устранением исчезают и остальные три ошибки.

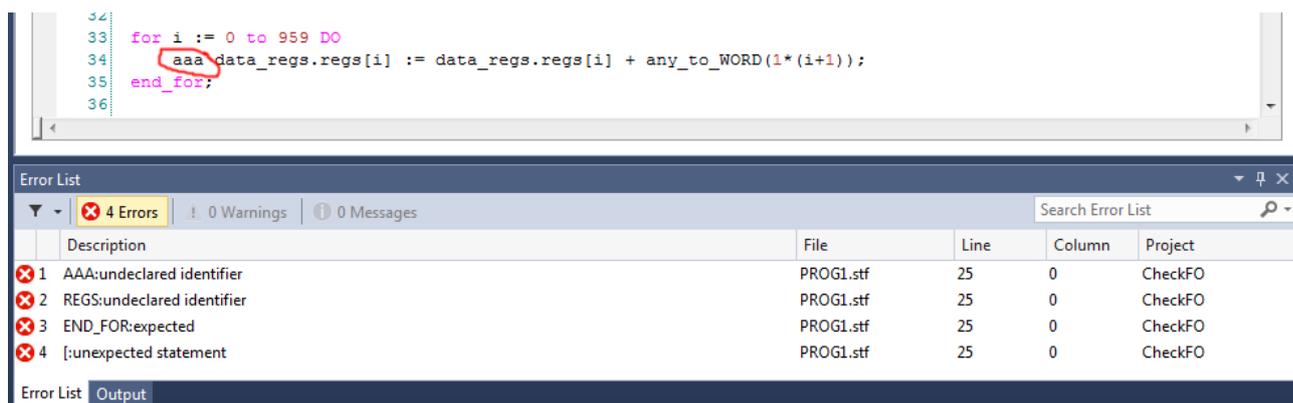


Рис. 2.32 – Пример вывода ошибок при сборке проекта

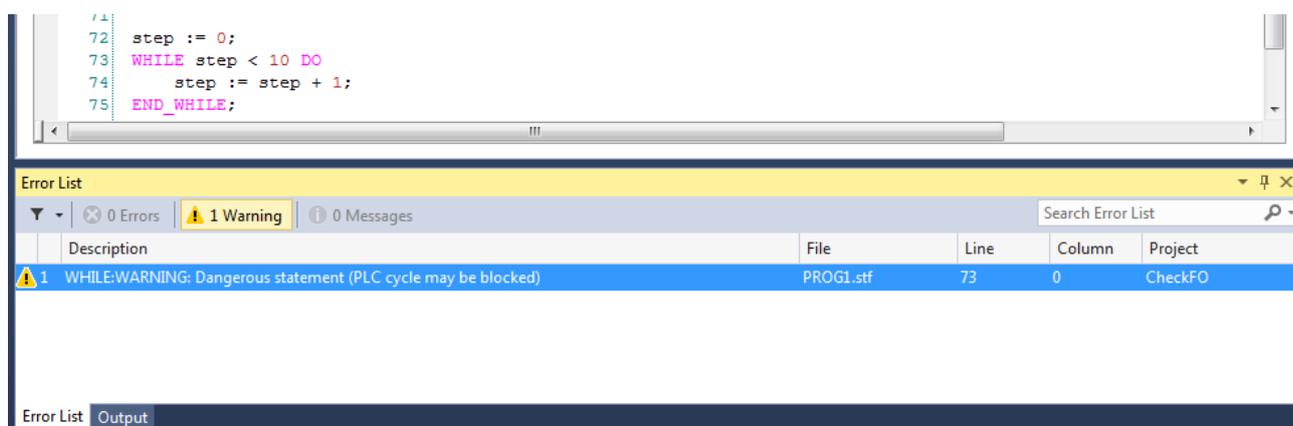


Рис. 2.33 – Пример вывода предупреждений при сборке проекта

Если в ходе сборки возникли предупреждения, они также будут выведены в окно Error List. В примере на Рис. 2.33 компилятор предупреждает об использовании функции,

способной вызвать зависание программы пользователя. Проект при этом будет собран и готов к загрузке в модуль CPU.

2.7 Загрузка проекта в модули CPU

После успешной сборки проекта появляется возможность загрузить его в модуль CPU (в случае проекта с поддержкой Failover – в пару модулей CPU). Для этого следует в дереве проектов выбрать соответствующее устройство и в его контекстном меню (либо на панели инструментов) выбрать пункт «Download» ([Рис. 2.34](#)).

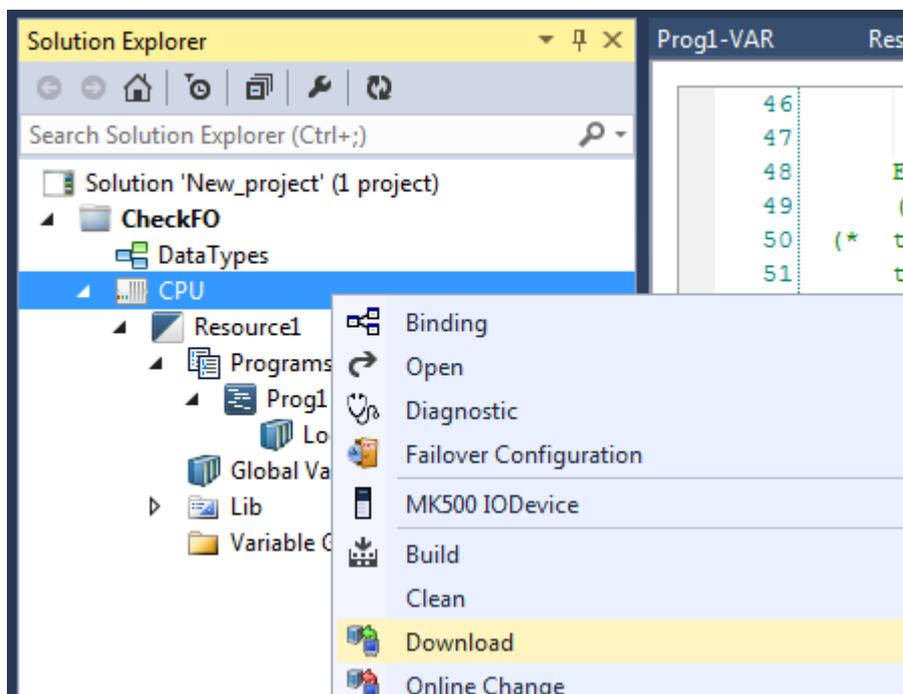


Рис. 2.34 – Загрузка собранного проекта в модуль CPU

Если в момент загрузки проекта в модуле CPU уже выполняется программа пользователя, будет выдано окно подтверждения ([Рис. 2.35](#)). Для продолжения загрузки проекта следует нажать кнопку «Да», нажатие кнопки «Нет» прекратит загрузку и никак не повлияет на выполнение уже запущенной в модуле CPU программы.

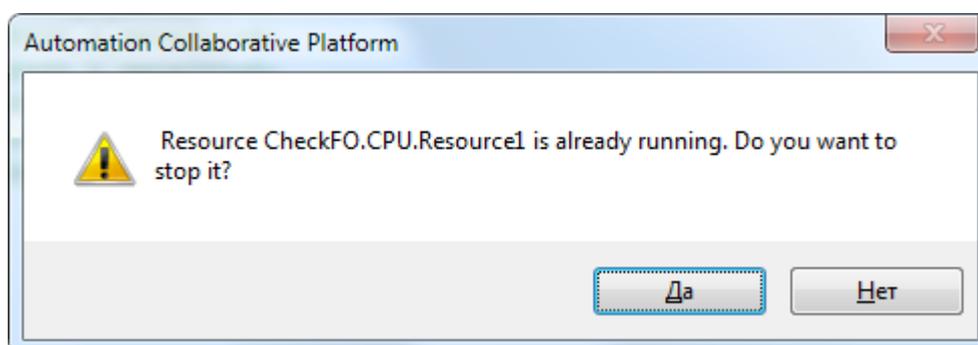
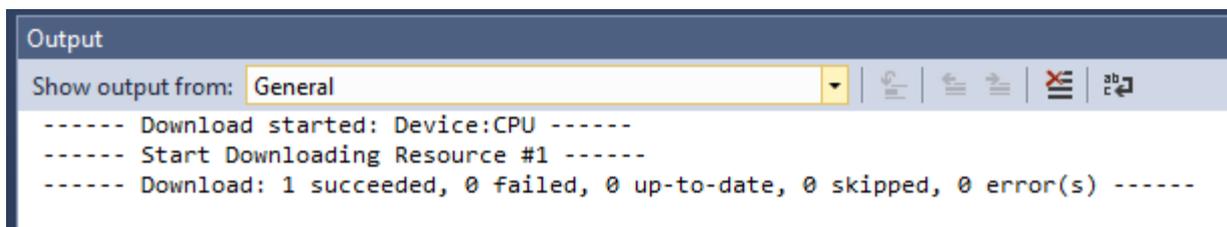


Рис. 2.35 – Окно подтверждения загрузки проекта в модуль CPU с уже запущенной программой пользователя

Если загрузка проекта завершилась успешно, в окно «Output» будет выведено соответствующее сообщение ([Рис. 2.36](#)).

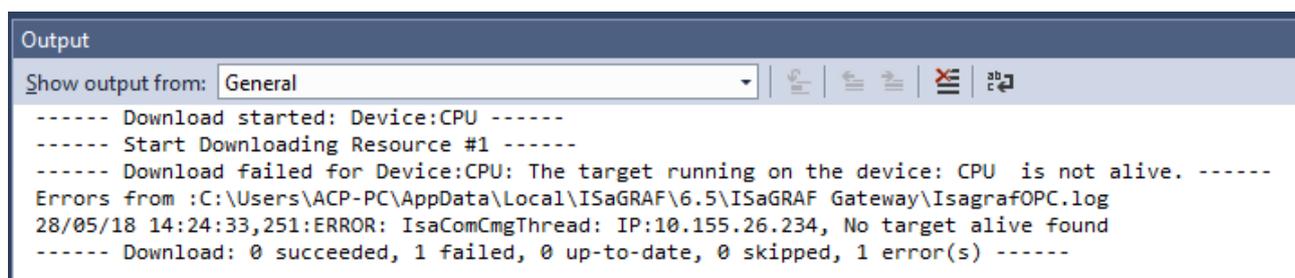


```
Output
Show output from: General
----- Download started: Device:CPU -----
----- Start Downloading Resource #1 -----
----- Download: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped, 0 error(s) -----
```

Рис. 2.36 – Вывод сообщения об успешной загрузке проекта в модуль CPU

Также при старте программы пользователя модуль CPU в течение одной секунды мигает всей индикацией на лицевой панели, если он не Secondary-модуль CPU.

В случае наличия ошибки (несовместимость версии tdb-файла и среды исполнения модуля, отсутствия связи между средой разработки и модулем CPU и т.п.) в окно Output будет выведено сообщение об ошибке. На [Рис. 2.37](#) приведено сообщение, выводимое средой разработки при отсутствии связи с модулем CPU.



```
Output
Show output from: General
----- Download started: Device:CPU -----
----- Start Downloading Resource #1 -----
----- Download failed for Device:CPU: The target running on the device: CPU is not alive. -----
Errors from :C:\Users\ACP-PC\AppData\Local\ISaGRAF\6.5\ISaGRAF Gateway\IsagrafOPC.log
28/05/18 14:24:33,251:ERROR: IsaComCmgThread: IP:10.155.26.234, No target alive found
----- Download: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped, 1 error(s) -----
```

Рис. 2.37 – Вывод сообщения об ошибке при отсутствии связи с модулем CPU

Также в среде разработки ACP могут быть доступны два пункта меню:

- «Upload» – выгрузка проекта пользователя из модуля CPU. В версии среды разработки ACP 6.50 это действие не может быть реализовано, так требует для своей работы значения параметра ресурса Embedded Zip Source отличное от None (см. [п. 2.5.2](#)).
- «Online Change» – применение изменений в проекте без остановки проекта. Позволяет применять без перезапуска проекта изменения, не затрагивающие устройства ввода-вывода и типы данных (см. [п. 2.7.1](#)). Следует использовать этот вариант обновления с осторожностью, так как в случае несовместимости внесённых изменений может потребоваться перезапуск модуля/модулей CPU для возврата их в рабочее состояние.

2.7.1 Загрузка проекта в модули CPU без остановки работы (Online Change)

Для загрузки проекта в CPU без остановки проекта должны быть выполнены следующие условия:

- в проект не должны вноситься изменения, затрагивающие конфигурацию ввода-вывода (добавление, удаление, редактирование параметров);
- в проект не должны добавляться новые типы данных или изменяться существующие типы;

– в проекте не должны создаваться и/или удаляться глобальные переменные.

Также после создания проекта следует увеличить в ресурсе как минимум на порядок значения параметров по умолчанию в секции «Memory Size for Online Changes» (см. [п. 2.5.2](#)). Если этого не сделать при попытке выполнить Online Change средой разработки будет выдано сообщение о нехватке памяти.

Порядок действий для выполнения Online Change:

1. Внести обновления в проект;
2. Выделить в дереве проектов элемент Device (см. [п. 2.5.1](#));
3. Выполнить операцию «Build» (не «Build Solution»!). После успешной сборки должна стать активной команда «Online Change».
4. Выполнить команду «Online Change». При соблюдении всех условий должна начаться загрузка собранного проекта в CPU. По окончании загрузки проекта в CPU средой разработки будет выдано сообщение о результате.

После успешного обновления изменится контрольная сумма проекта в контроллере и число выполненных Online Change (см. [п. 3.7.6](#)).

2.8 Мониторинг и управление работой программы в модуле CPU

Среда разработки АСР 6.50 предоставляет следующие возможности мониторинга и управления работой программы пользователя в модуле CPU:

- диагностика работы программы пользователя;
- просмотр и модификация значения переменных программы пользователя;
- управление работой ресурса программы пользователя.

Все эти возможности доступны только в онлайн-режиме среды разработки. Для перехода в онлайн-режим следует на панели инструментов «Debug» выбрать режим работы «Online» и нажать кнопку «Start Debugging» (Рис. 2.38).

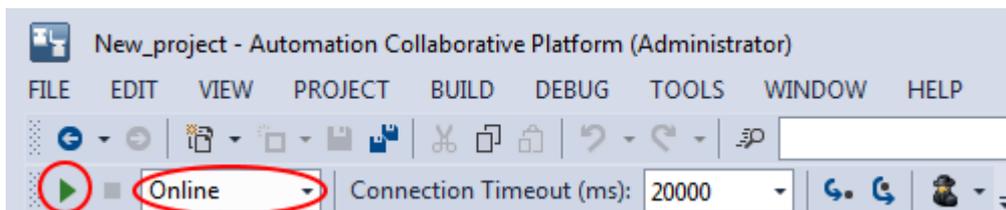


Рис. 2.38 – Перевод среды разработки АСР в онлайн-режим

Для перехода в онлайн-режим, модуль CPU должен быть доступен среде разработки по сети. Таймаут подключения к модулю CPU определяется параметром Connection Timeout (по умолчанию 20000 мс, Рис. 2.38) на панели инструментов «Debug».

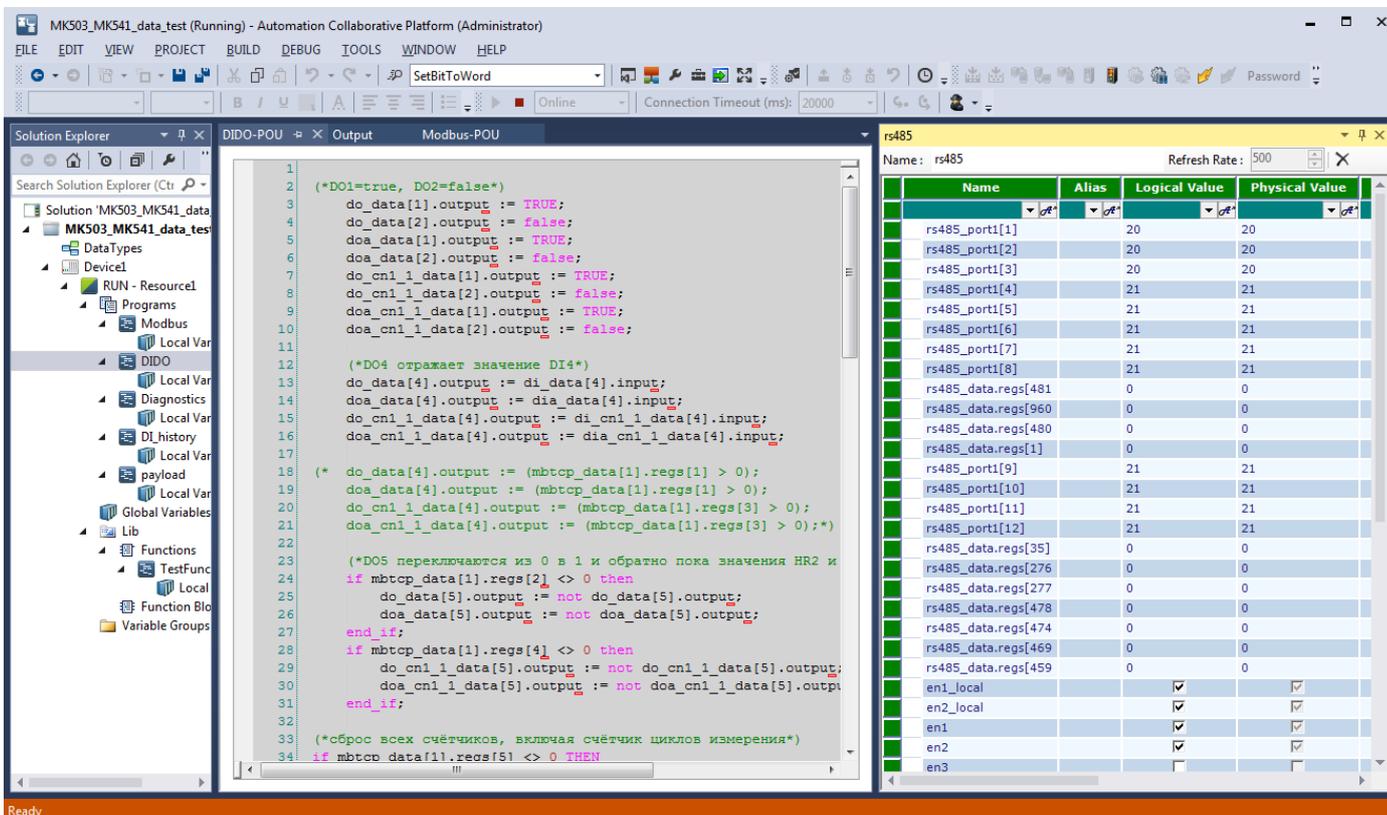


Рис. 2.39 – Внешний вид среды разработки АСР в онлайн-режиме

После успешного перехода в онлайн-режим строка статуса среды разработки становится оранжевого цвета, редактирование программы пользователя становится невозможным ([Рис. 2.39](#)). При этом становятся доступными функции онлайн-мониторинга состояния программы пользователя, функции управления работой ресурсов программы пользователя и функции просмотра и модификации переменных программы пользователя.

2.8.1 Диагностика работы программы пользователя

Для просмотра диагностической информации ресурса или всего устройства следует в онлайн-режиме выбрать в дереве проекта ресурс или устройство, нажать правую кнопку мыши и выбрать в контекстном меню пункт «Diagnostic» (Рис. 2.40).

В открывшемся окне диагностики (Рис. 2.41) выводится следующая информация обо всех ресурсах устройства (в случае, если в дереве проекта был выбран ресурс – только об этом ресурсе):

- число ресурсов устройства, запущены ли они и сохранены ли они в энергонезависимую память модуля CPU;
- контрольные суммы, номера версий (в рамках проекта) и даты загрузки ресурса в модуль CPU;
- использованием ресурсом памяти модуля CPU.

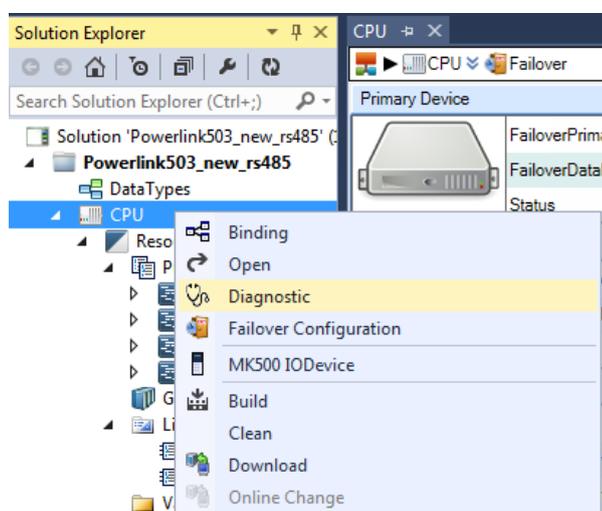


Рис. 2.40 – Открытие окна диагностики

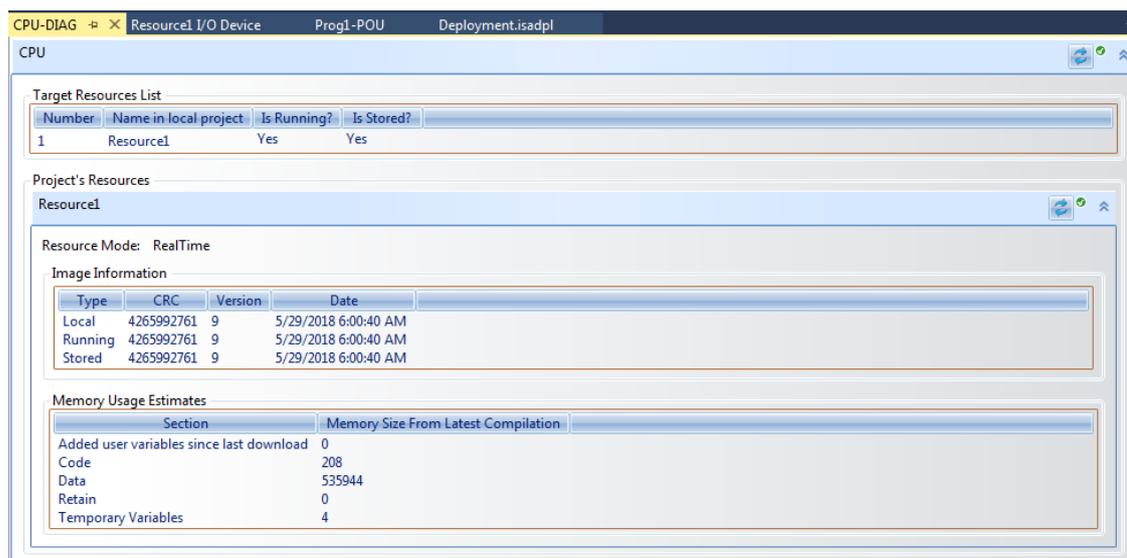
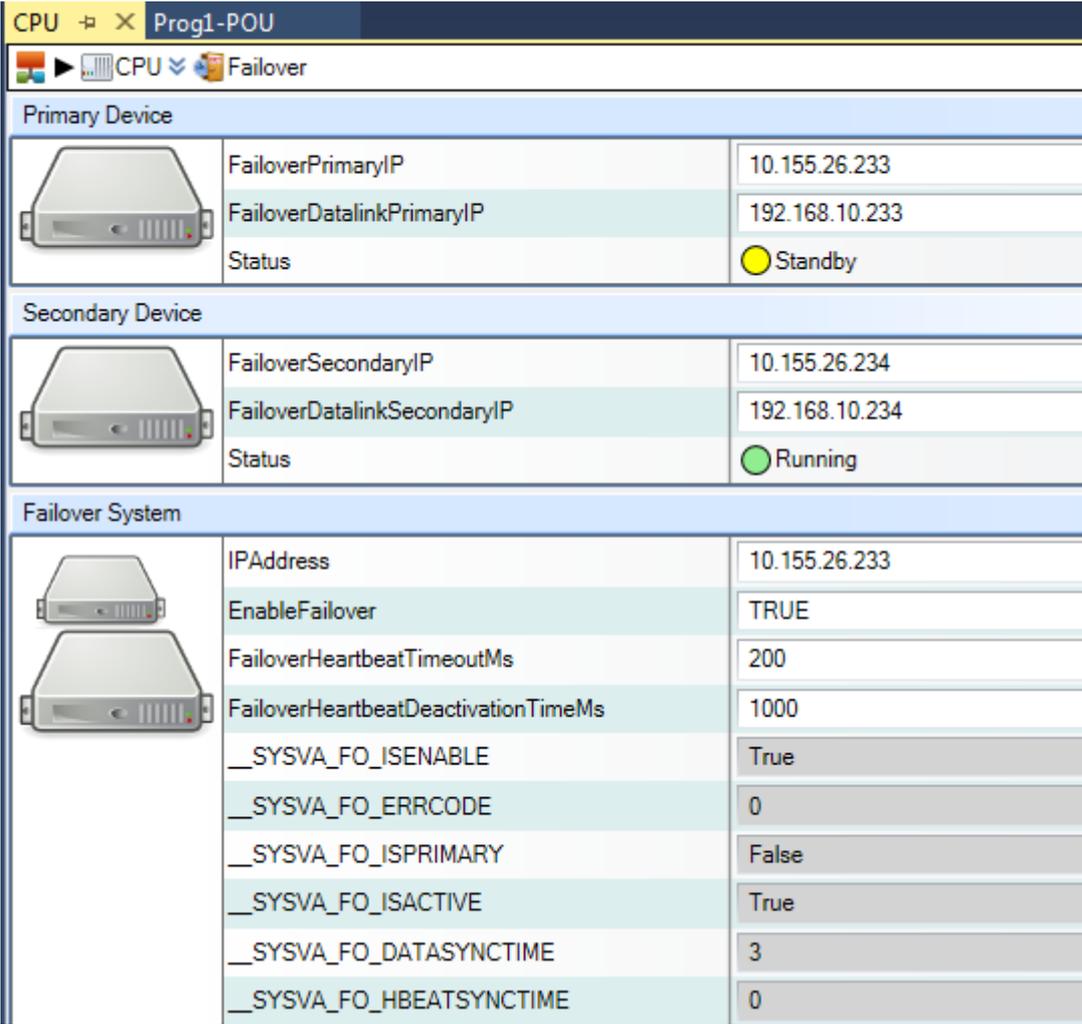


Рис. 2.41 – Окно диагностики устройства

Для модулей CPU с поддержкой режима Failover также доступно окно онлайн-диагностики работы системы резервирования. Для его просмотра следует в онлайн-режиме выбрать в дереве проекта устройство, и в его контекстном меню выбрать пункт «Failover Configuration» (Рис. 2.26). Откроется окно Failover Configuration (п. 2.4)

В онлайн-режиме редактирование сетевых параметров недоступно, вместо этого в окне Failover Configuration (Рис. 2.42) выводится информация о состоянии программы пользователя в ведущем и ведомом модулях CPU резервированной пары, а также подробная информация о текущем состоянии системы резервирования в модуле CPU, назначенном Primary при настройке сетевых параметров.



Primary Device		
	FailoverPrimaryIP	10.155.26.233
	FailoverDatalinkPrimaryIP	192.168.10.233
	Status	Standby
Secondary Device		
	FailoverSecondaryIP	10.155.26.234
	FailoverDatalinkSecondaryIP	192.168.10.234
	Status	Running
Failover System		
	IPAddress	10.155.26.233
	EnableFailover	TRUE
	FailoverHeartbeatTimeoutMs	200
	FailoverHeartbeatDeactivationTimeMs	1000
	__SYSVA_FO_ISENABLE	True
	__SYSVA_FO_ERRCODE	0
	__SYSVA_FO_ISPRIMARY	False
	__SYSVA_FO_ISACTIVE	True
	__SYSVA_FO_DATASYNCTIME	3
	__SYSVA_FO_HBEATSYNCTIME	0

Рис. 2.42 – Окно сетевых параметров проекта с поддержкой режима Failover в онлайн-режиме

Более подробную информацию о параметрах системы резервирования можно узнать из документа «Механизм обеспечения отказоустойчивости» (fvr_ISa6_ru).

2.8.2 Просмотр и модификация значения переменных программы пользователя

Для просмотра и модификации переменных программы пользователя в онлайн-режиме предназначены так называемые «Spy list».

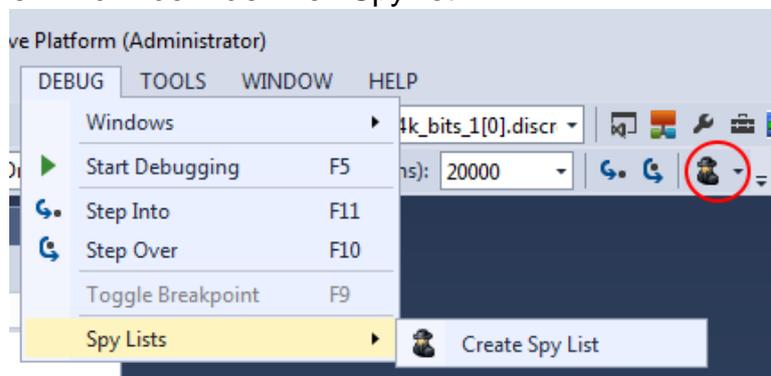


Рис. 2.43 – Создание экземпляра «Spy list»

Для создания нового экземпляра «Spy list» следует в меню среды разработки выбрать «DEBUG»→«Spy Lists»→«Create Spy List» либо нажать кнопку «Create Spy List» на панели инструментов (Рис. 2.43).

Для добавления интересующей нас переменной следует поместить курсор на свободное поле колонки «Name» окна «Spy list» и начать вводить имя переменной. Появится выпадающий список с именами доступных переменных, из которых можно выбрать нужную. Если окно «Spy list» открыто не в онлайн-режиме, в нём можно также указать имя окна (для удобства пользования) и период обновления данных в онлайн-режиме (Рис. 2.44).

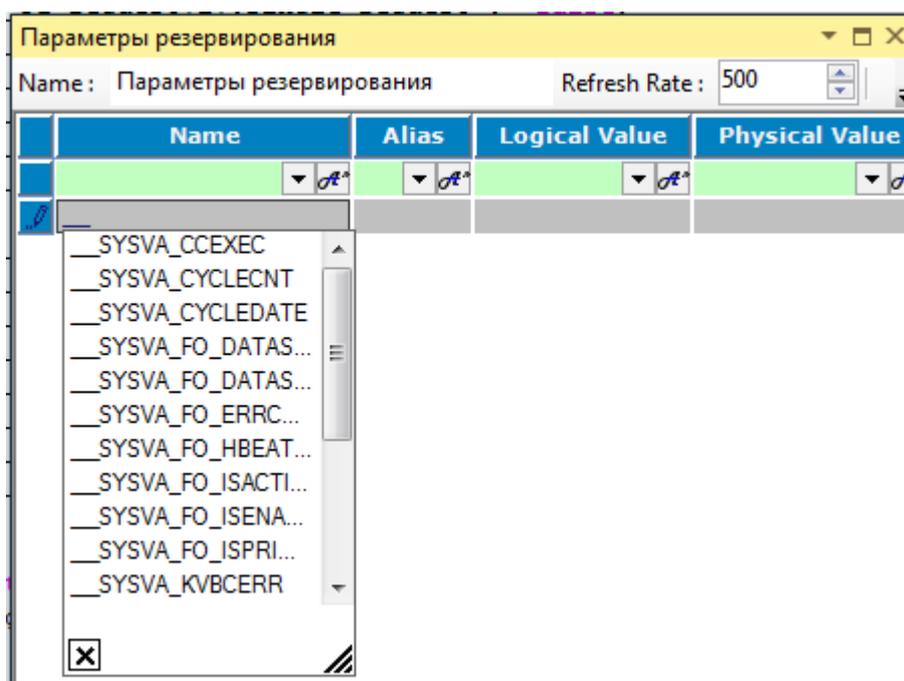


Рис. 2.44 – Настройки «Spy list» и формирование списка переменных

В онлайн-режиме в открытом окне «Spy list»-а выводятся текущие значения переменных программы пользователя (Рис. 2.45).

Параметры резервирования							
Name: Параметры резервирования		Refresh Rate: 500					
Name	Alias	Logical Value	Physical Value	Lock	Comment	Access Path	Data Type
__SYSVA_TCYCURRE		T#7ms	T#0s	<input type="checkbox"/>	Current cycle	CheckFO.CPU.Resource1	TIME

Рис. 2.45 – Просмотр переменных программы пользователя в онлайн-режиме

Для модификации значений переменных программы пользователя следует поместить курсор в колонку «Logical Value» соответствующей переменной в окне «Spy list», ввести необходимое значение и нажать «Enter» (Рис. 2.46).

Параметры резервирования							
Name: Параметры резервирования		Refresh Rate: 500					
Name	Alias	Logical Value	Physical Value	Lock	Comment	Access Path	Data Type
__SYSVA_TCYCURRE		T#13ms	T#0s	<input type="checkbox"/>	Current cycle	CheckFO.CPU.Resource1	TIME
i		11	0	<input type="checkbox"/>		CheckFO.CPU.Resource1.Prog	DINT

Рис. 2.46 – Модификация переменных программы пользователя в онлайн-режиме

Альтернативный вариант модификации переменной – выполнить на строке с интересующей переменной в окне «Spy list» двойной клик либо из контекстного меню выбрать пункт «Write Variable...». Затем в открывшемся окне (Рис. 2.47) можно ввести и записать новое значение переменной, либо защитить переменную от дальнейших изменений (опция «Lock»). Защищать переменную без необходимости не рекомендуется, так как при попытке записи в неё может произойти зависание ресурса, выйти из которого можно только перезапуском модуля CPU.

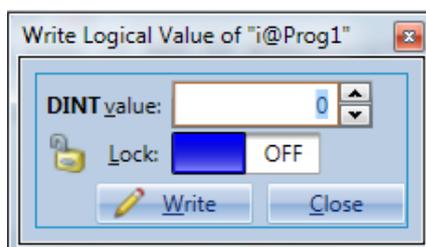


Рис. 2.47 – Окно записи значения переменной программы пользователя в онлайн-режиме

2.8.3 Управление работой ресурса программы пользователя

В онлайн-режиме среда разработки позволяет выполнять следующие операции над исполняемым в модуле CPU ресурсом:

- останавливать и запускать исполнение ресурса;
- переводить режим исполнения ресурса в режим пошагового исполнения и обратно в непрерывный режим;
- включать и отключать мониторинг системных сообщений среды исполнения ISaGRAF.



Рис. 2.48 – Кнопки «Stop Resource(s)» (сверху) и «Start Resource(s)» (снизу) панели инструментов «Target Execution»

Для остановки и запуска ресурса служат кнопки «Stop Resource(s)» и «Start Resource(s)» панели инструментов «Target Execution» ([Рис. 2.48](#)). Их нажатие в онлайн-режиме позволяет остановить и заново запустить исполнение ресурсов в текущем устройстве. Данная операция может быть полезна для выполнения перезапуска программы пользователя без перезагрузки модуля CPU.



Рис. 2.49 – Кнопки «Cycle-to-Cycle Mode» (сверху), «One Cycle» (в центре) и «Real Time Mode» (снизу) панели инструментов «Target Execution»

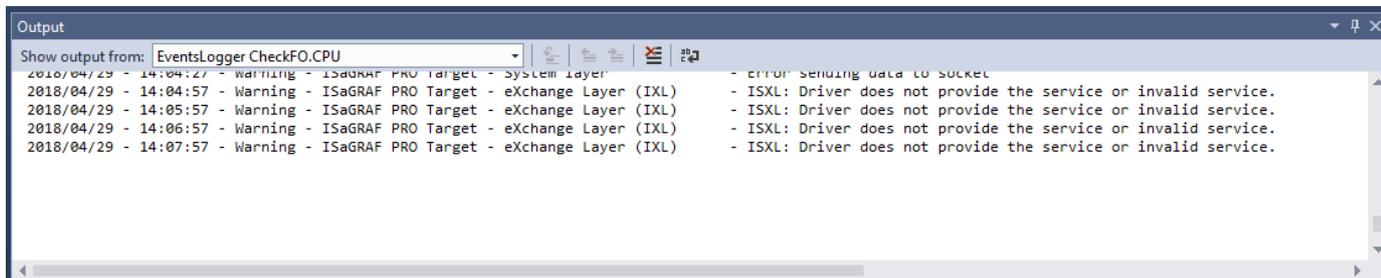
Для перевода режима исполнения ресурса в пошаговый режим, исполнения одного цикла программы пользователя и обратно в непрерывный режим работы служат кнопки «Cycle-to-Cycle Mode», «One Cycle» и «Real Time Mode» панели инструментов «Target Execution» ([Рис. 2.49](#)). В пошаговом режиме удобно отлаживать программу пользователя.



Рис. 2.50 – Кнопки «Start System Events» (сверху) и «Stop System Events» (снизу) панели инструментов «Target Execution»

Для просмотра системных сообщений среды исполнения ISaGRAF служат кнопки «Start System Events» (сверху) и «Stop System Events» (снизу) панели инструментов «Target Execution» ([Рис. 2.50](#)). При включённом режиме вывода системных сообщений в окно Output среды исполнения выводятся в режиме реального времени системные сообщения среды исполнения ISaGRAF ([Рис. 2.51](#)), что может быть полезным для

диагностики системных сбоев среды исполнения (если возникнет подозрение на их наличие).



```
Output
Show output from: EventsLogger CheckFO.CPU
2018/04/29 - 14:04:27 - Warning - ISaGRAF PRO Target - System Layer - error sending data to socket
2018/04/29 - 14:04:57 - Warning - ISaGRAF PRO Target - eXchange Layer (IXL) - ISXL: Driver does not provide the service or invalid service.
2018/04/29 - 14:05:57 - Warning - ISaGRAF PRO Target - eXchange Layer (IXL) - ISXL: Driver does not provide the service or invalid service.
2018/04/29 - 14:06:57 - Warning - ISaGRAF PRO Target - eXchange Layer (IXL) - ISXL: Driver does not provide the service or invalid service.
2018/04/29 - 14:07:57 - Warning - ISaGRAF PRO Target - eXchange Layer (IXL) - ISXL: Driver does not provide the service or invalid service.
```

Рис. 2.51 – Пример вывода системных сообщений среды исполнения ISaGRAF

2.8.4 Парольная защита проекта и целевого устройства

Парольная защита подробно описана в документе «cam_ISa6_ru.pdf». В данном разделе приводятся только краткие комментарии относительно механизма работы защиты.

2.8.4.1 Парольная защита проекта

Парольная защита проекта может быть применена почти к любому элементу проекта. При этом ограничения для неавторизованного доступа для различных элементов проекта заметно отличаются ([Табл. 1.0](#)).

Табл. 1.0– Ограничения для неавторизованного доступа

Защищённый паролем элемент	Ограничения на изменение
Проект	Открытие проекта, управление парольной защитой для всех прочих элементов проекта
Устройство	Добавление, редактирование и удаление ресурса, программы, библиотечных функций или функционального блока
Ресурс	Добавление, редактирование и удаление программы, библиотечных функций или функционального блока
Программа	Просмотр программы
Пользовательская функция	Просмотр функции
Пользовательский функциональный блок	Просмотр функционального блока

Пароль проекта сохраняется внутри проекта. При копировании, вставке, импорте и экспорте элементов с настройками парольной защиты, заданные пароли сохраняются.

2.8.4.2 Парольная защита целевого устройства

Парольная защита целевого устройства (Target Password) блокирует неавторизованный доступ к целевому устройству по протоколу IXL. Это означает, что при попытке среды разработки АСР загрузить проект в защищённый модуль CPU ([п. 2.7](#)) либо при попытке подключиться к защищённому модулю CPU Online ([п. 2.8](#)) будет выведено приглашение на ввод пароля. После успешного ввода пароля дальнейшая работа среды разработки с защищённым модулем CPU выполняется без постоянных запросов пароля, пока в среду разработки АСР не будет загружен новый проект.

Пароль целевого устройства сохраняется внутри модуля CPU.

2.9 Работа с эмулятором МК-500

Для отладки программы пользователя в отсутствие модулей CPU предназначен эмулятор ISaGRAF МК-500.

Эмулятор ISaGRAF МК-500 представляет собой среду исполнения ISaGRAF, собранную под ОС Windows и предоставляющую существенную часть возможностей реальных контроллеров МК-502-142 и МК-503-120.



ВНИМАНИЕ

На 04.2023 не поддерживается эмуляция контроллера МК-501-022. Также не поддерживается эмуляция обмена данными с модулями ввода-вывода.

2.9.1 Установка эмулятора

Для установки эмулятора ISaGRAF МК-500 следует запустить программу-инсталлятор (файл ISaGRAF-SIM_2023-03-06.exe) и следовать указаниям мастера установки.

После успешной установки эмулятора в меню «Пуск» создаётся папка ISAGRAF-SIM, содержащая следующие ярлыки для контроллеров МК-502-142 и МК-503-120:

- ярлыки запуска эмуляторов контроллеров;
- ярлыки очистки программы пользователя;
- ссылки на эмулируемые папки FTP;
- ярлык на деинсталлятор эмулятора.

Для удаления эмулятора ISaGRAF МК-500 следует запустить ярлык-деинсталлятор.

2.9.2 Работа с эмулятором

Для работы с эмулятором необходимо:

- если эмулятор запускается не в первый раз, при необходимости – удалить предыдущий проект кликом по ярлыку из папки «Пуск→ISAGRAF-SIM», Стереть программу пользователя МК501_2 или МК503;
- запустить эмулятор кликом по ярлыку из папки «Пуск→ISAGRAF-SIM» (МК501_2 или МК503);
- при необходимости – разрешить брэндмауэру Windows работу программ ETCP.exe, ISaGRAF.exe и IsaVM.exe;
- для нормальной работы с эмулятором также следует разрешить доступ к портам 1131 (для загрузки программы пользователя) и 502 (для работы ModbusTCP)
- скопировать tdb-файл эмулируемого контроллера из соответствующей папки FTP эмулятора («Пуск→ISAGRAF-SIM→Папка FTP ISaGRAF-SIM», МК501_2 или МК503);
- импортировать tdb-файл эмулируемого контроллера в проект в АСР (см. [п. 2.2](#));
- выбрать соответствующий «Target» (NFT_MK501_2-SIM или NFT_MK503-SIM) в свойствах устройства (см. [п. 2.2](#));

- скопировать необходимые конфигурационные файлы в соответствующую папку FTP эмулятора;
- в сетевых настройках проекта следует установить в качестве IPAddress и FailoverPrimaryIP IP-адрес компьютера, на котором запущен эмулятор (см. [п. 2.4](#));
- также в сетевых настройках проекта рекомендуется отключить поддержку резервирования (см. [п. 2.4](#));
- пересобрать и загрузить проект в эмулятор.

Эмулятор имеет следующие особенности, отличающие его поведение от работы среды исполнения ISaGRAF в реальном модуле CPU:

- так как программа пользователя фактически выполняется на персональном компьютере, с более мощным центральным процессором, чем в модулях CPU, производительность программы под эмулятором заметно выше её же производительности на модуле CPU.

2.9.3 Ограничения эмулятора

На 04.2023 эмулятор ISaGRAF МК-500 имеет следующие ограничения:

- не поддерживается работа с модулем CPU МК-501-022;
- не поддерживается работа CPU в режиме резервирования (резервный модуль CPU всегда считается отсутствующим);
- отсутствует эмуляция обмена данными с модулями ввода-вывода (все входные данные модулей ввода-вывода - нулевые);
- не реализована поддержка работы по протоколу ModbusRTU;
- не реализована поддержка работы по протоколу Powerlink;
- поддержка работы по протоколу IEC 60870-5-104 реализована частично.

Глава 3 Работа с модулями изделия в среде разработки ACP Workbench ISaGRAF 6.5

3.1 Общие принципы работы с модулями изделия

Далее в РП под модулями изделия в среде разработки ACP Workbench ISaGRAF 6.50 подразумеваются устройства ввода-вывода (I/O Device), соответствующие реальным модулям контроллера программируемого логического МКLogic-500 (см. разделы 2..9 руководства по эксплуатации КДСА.426471.004 РЭ_2.0_00).

Согласно документа «Монтаж ввода-вывода» (*iow_ISa6_ru*), все модули изделия являются комплексными устройствами ввода-вывода, т.е. состоят их нескольких простых устройств ввода-вывода.

Все прочие устройства ввода-вывода, доступные для добавления в проект, далее именуются дополнительными устройствами ввода-вывода.

3.1.1 Добавление и удаление модулей изделия

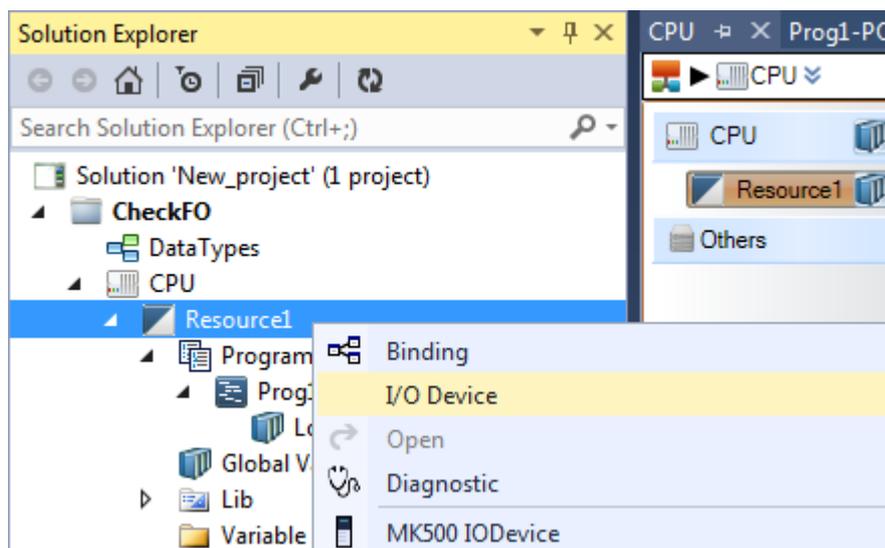


Рис. 3.1 – Вызов окна I/O Device

При работе в среде разработки ACP Workbench ISaGRAF 6.50 добавление, конфигурация и удаление модулей изделия выполняется в окне I/O Device. Для его открытия следует в дереве проекта выбрать требуемый ресурс, нажать правую кнопку мыши и из контекстного меню выбрать пункт «I/O Device» (Рис. 3.1).

В открывшемся окне (Рис. 3.2) можно увидеть панель инструментов (слева), окно устройств (в центре), окно каналов (справа сверху) и окно параметров каналов (справа снизу).

Добавление и удаление модулей изделия в проекте выполняется с помощью кнопок «Add Device» и «Delete» панели инструментов (Рис. 3.3).

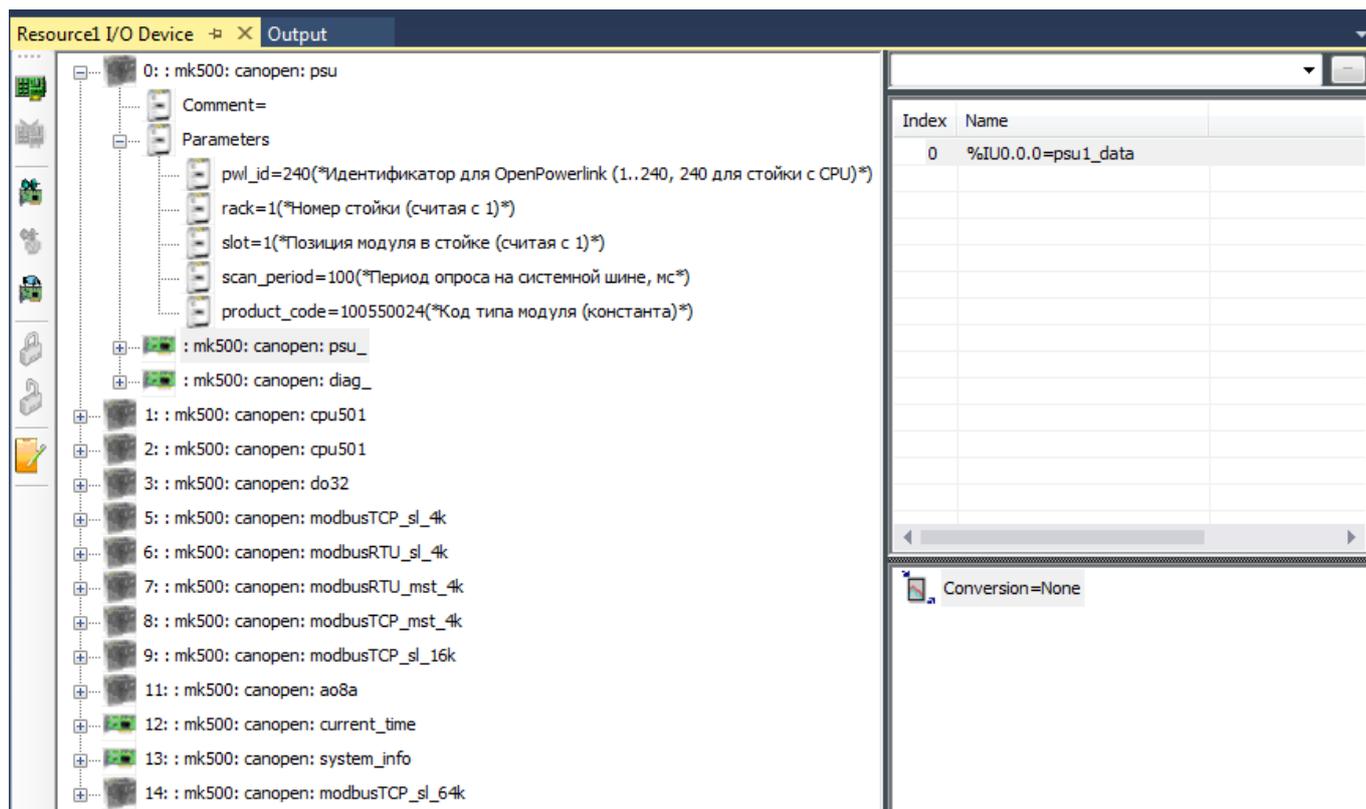


Рис. 3.2 – Пример окна I/O Device



Рис. 3.3 – Кнопки «Add Device» и «Delete» панели инструментов окна I/O Device

Нажатие на кнопку «Add Device» вызывает окно «Device Selector» ([Рис. 3.4](#)), в котором следует выбрать необходимый модуль изделия, задать ему индекс в списке устройств, выбрать (если это возможно) число каналов ввода-вывода, добавить псевдоним (Alias), описание (Description) и комментарий, после чего нажать кнопку «ОК» для добавления устройства.

**ВНИМАНИЕ**

В окне «I/O Device» комбинация клавиш Ctrl+S не приводит к сохранению проекта. Вместо этого она убирает текстовые комментарии в окне устройств. Для возврата текстовых комментариев обратно следует повторно нажать Ctrl+S. Для сохранения проекта во время работы в окне «I/O Device» следует пользоваться иконкой сохранения на панели инструментов среды разработки либо соответствующим пунктом меню «FILE».

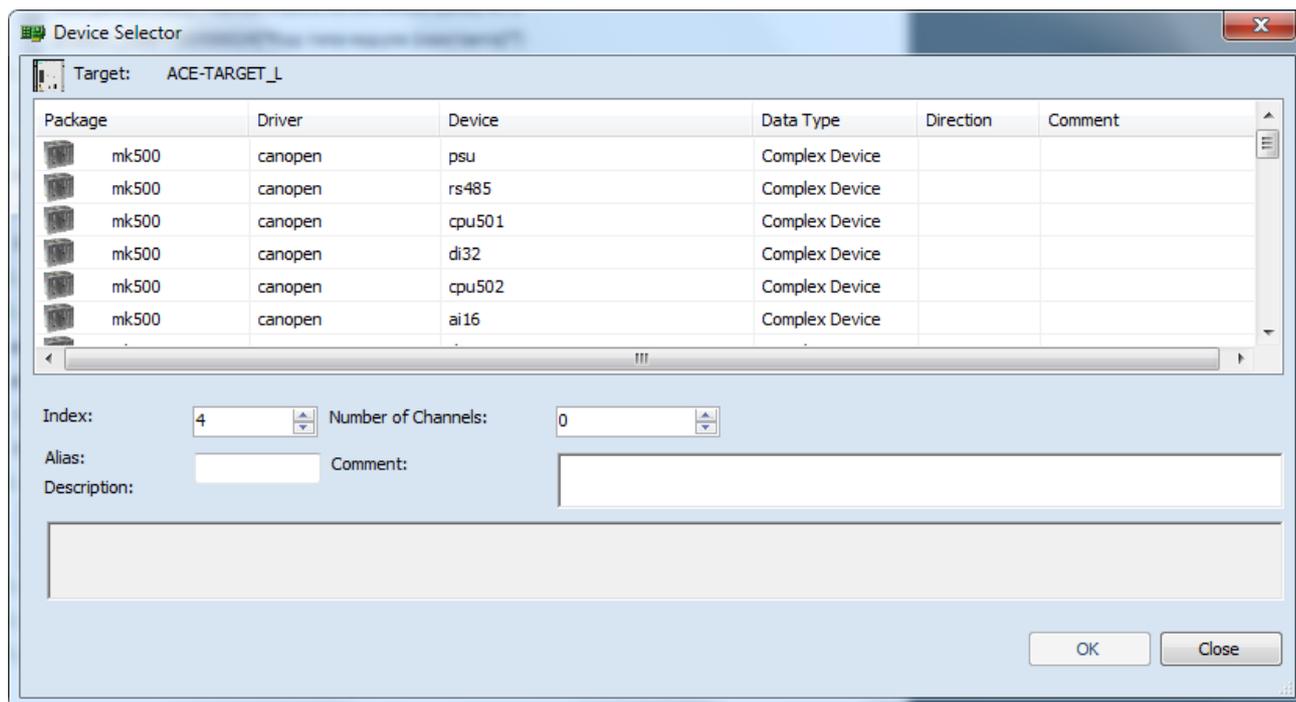


Рис. 3.4 – Окно «Device Selector»

Табл. 1.1– Обозначения модулей изделия в окне «Device Selector» и их соответствия модулям контроллера

Обозначение модулей изделия в окне «Device Selector»	Название модуля контроллера	Назначение
psu	МК-550-024	Модуль питания напряжения постоянного тока с интерфейсом CAN
cpu501	МК-501-022	Модуль центрального процессора с 2 интерфейсами Ethernet 100/1000 Base-T и 2 интерфейсами RS-485
cpu502	МК-502-142	Модуль центрального процессора с 1 оптоволоконным интерфейсом резервирования, 4 интерфейсами Ethernet 100/1000 Base-T и 2 интерфейсами RS-485
cpu503	МК-503-120	Модуль центрального процессора с 1 оптоволоконным интерфейсом резервирования, 2 интерфейсами Ethernet 100/1000 Base-T
cn545	МК-545-010	Коммуникационный модуль с 1 портом Ethernet 100/1000 Base-T с поддержкой Powerlink с двумя хаб-выходами
ai8 ai8a	МК-516-008 МК-516-008 A	Модуль аналогового ввода с 8 изолированными аналоговыми входами 0-20 (4-20) мА
ai8ahart	МК-576-008 A	Модуль аналогового ввода с 8 изолированными аналоговыми входами 0-20 (4-20) мА с поддержкой HART
ai16 ai16a	МК-513-016	Модуль аналогового ввода с 16 аналоговыми входами 0-20 (4-20) мА

Обозначение модулей изделия в окне «Device Selector»	Название модуля контроллера	Назначение
ai16ahart	МК-576-016 А	Модуль аналогового ввода с 16 изолированными аналоговыми входами 0-20 (4-20) мА с поддержкой HART
ao8 ao8a	МК-514-008 МК-514-008 А	Модуль аналогового вывода с 8 аналоговыми выходами 0-20 (4-20) мА
ao8ahart	МК-574-008 А	Модуль аналогового вывода с 8 аналоговыми выходами 0-20 (4-20) мА с поддержкой HART
di32 di32history di32a di32ahistory	МК-521-032 МК-521-032 А	Модуль дискретного ввода напряжения постоянного тока с 32 дискретными входами
di32anamur	МК-523-032 А	Модуль дискретного ввода напряжения постоянного тока с 32 входами NAMUR (согласно IEC 60947-5-6)
do32anamur	МК-532-032 А	Модуль дискретного вывода напряжения постоянного тока с 32 выходами NAMUR (согласно IEC 60947-5-6)
do32 do32a	МК-531-032 МК-531-032 А	Модуль дискретного вывода напряжения постоянного тока с 32 дискретными выходами
rs485new	МК-541-002	Коммуникационный модуль с 2 интерфейсами RS-485

Нажатие на кнопку «Delete» панели инструментов или клавиатуры удалит выбранный модуль изделия из списка устройств.

 **ВНИМАНИЕ**

Нажатие кнопки «Delete» удаляет выбранный модуль изделия без дополнительных подтверждений, поэтому следует быть внимательным при использовании этой кнопки, и в случае ошибочного удаления использовать команду «Undo» панели инструментов среды разработки, или комбинацию клавиш «Ctrl+Z».

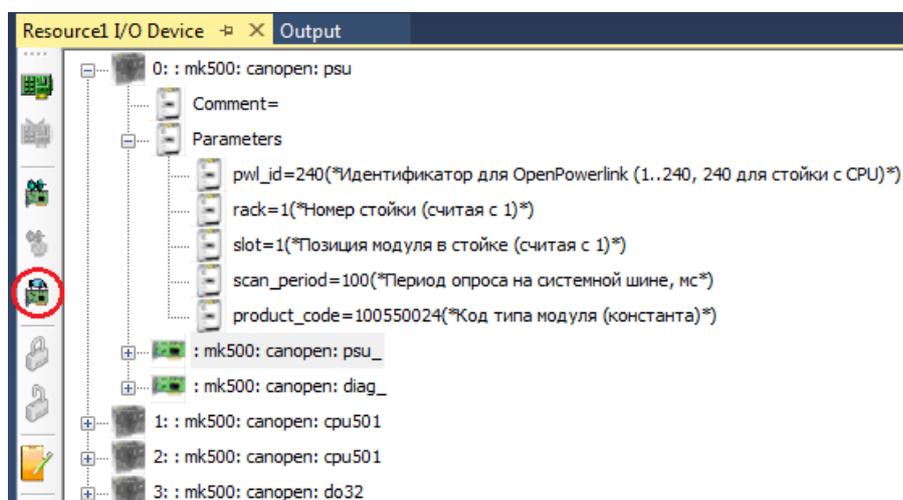


Рис. 3.5 – Кнопка «Real/Virtual I/O Device»

Если необходимо временно исключить модуль изделия из проекта, но не удалять, можно перевести его в виртуальное состояние. Для этого следует выбрать в окне устройств необходимый модуль изделия и нажать на панели инструментов окна кнопку «Real/Virtual I/O Device». В результате ([Рис. 3.5](#)) выбранный модуль изделия «побледнеет» в окне устройств, и не будет обрабатываться средой исполнения ISaGRAF.

Для вывода модуля изделия из виртуального состояния следует выбрать его в окне устройств и повторно нажать кнопку «Real/Virtual I/O Device».

**ВНИМАНИЕ**

Перевод модуля в виртуальное состояние не обрабатывается средой исполнения, но его данные так же синхронизируются между модулями CPU, внося свою долю в увеличение времени цикла.

3.1.2 Ранжирование модулей изделия

Под ранжированием модулей изделия понимается изменение положения модулей изделия в списке путём изменения индекса модулей. Индекс модуля изделия в списке (отображается справа от иконки устройства в окне устройств, [Рис. 3.5](#)) важен, так как он определяет порядок (от меньшего индекса к большему) инициализации, вызова функций ввода-вывода и деинициализации модуля в ходе работы среды исполнения.

При проектировании рекомендуется назначать индексы модулям не подряд, а разделяя модули на группы по принадлежности к стойкам (например, первая стойка – индексы с 0 по 99, вторая стойка – индексы с 200 по 299 и т.п.), оставляя интервалы, как между отдельными модулями изделия, так и между стойками. Этот подход позволит относительно легко добавить новые либо ранжировать ранее добавленные модули изделия.

Далее по тексту выражения типа «в ходе ранжирования модуль А в конфигурации следует размещать ниже, чем модуль В» следует понимать как «в ходе ранжирования индекс модуля А в конфигурации следует задать больше, чем у модуля В». При этом визуально в окне устройств (см. [Рис. 3.5](#)) устройство А будет находиться ниже устройства В.



ВНИМАНИЕ

При назначении индексов модулям нельзя превышать значение 25391, возможно неопределённое поведение CPU. Например, ошибка при запуске программы пользователя с кодом 17 (см. [п. 5.3.1](#)).

Для ранжирования модуля изделия (а также для редактирования псевдонима и/или комментария к модулю) следует дважды кликнуть левой кнопкой мыши на требующем корректировки модуле. Откроется окно «Device Selector», без возможности добавлять новые модули, но с возможностью изменить индекс модуля, псевдоним либо комментарий. После введения нового индекса (не совпадающего с уже существующим) следует нажать «ОК».

3.1.3 Настройка параметров модуля изделия

Все модули изделия имеют унифицированные параметры ([Рис. 3.6](#)):

- `pwl_id` – идентификатор на шине Powerlink. Для модулей ввода-вывода, находящихся на одной шине CAN с модулями CPU, следует оставлять значение 240. Подробнее см. в [п. 3.7.4](#).
- `rack` – номер стойки модуля (считая с 1), должен совпадать с положением переключателя «ADDRESS» на блоке питания стойки;
- `slot` – номер слота модуля в стойке (считая с 1), следует указывать порядковый номер модуля в стойке, учитывая особые случаи, описанные в разделе 15.3 руководства по эксплуатации КДСА.426471.004 РЭ_2.0_00;



ВНИМАНИЕ

Под модули питания МК-550-024 резервируются первые два значения `slot` в любой корзине. При этом ожидается установка не более двух модулей питания в начало стойки модуля. Это значит, что значение `slot` для первого модуля в стойке после блоков питания всегда будет 3.

- `scan_period` – период опроса устройства на шине CAN, в мс, следует выбирать согласно требованиям к скорости обновления данных в программе пользователя. Не рекомендуется устанавливать период меньше 10 мс;
- `product_code` – константа, соответствует коду модуля устройства.

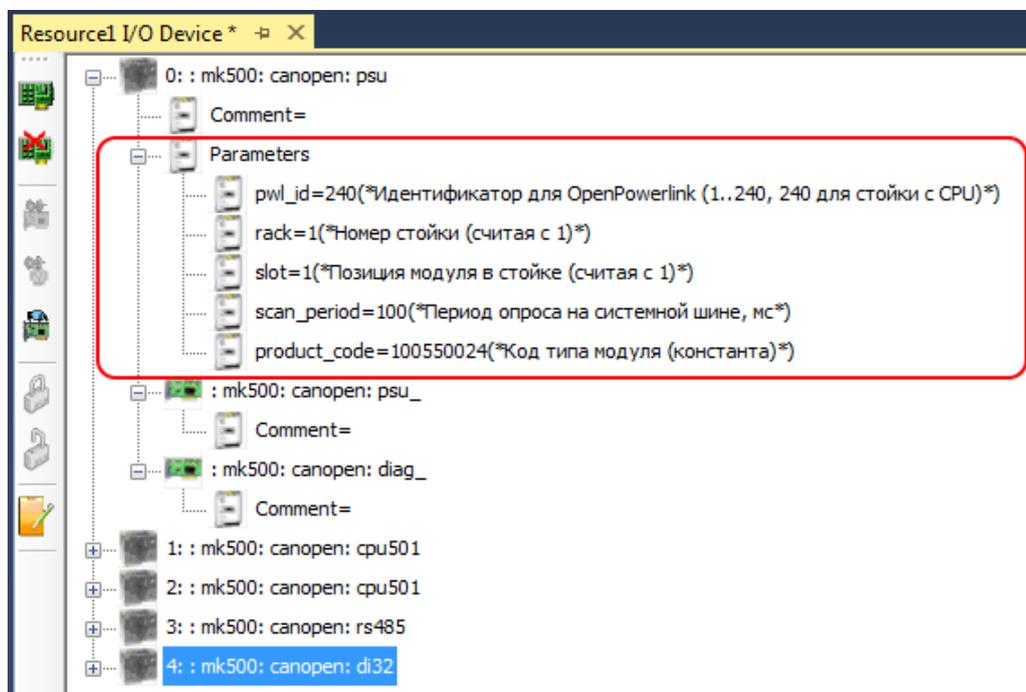


Рис. 3.6 – Параметры модуля изделия

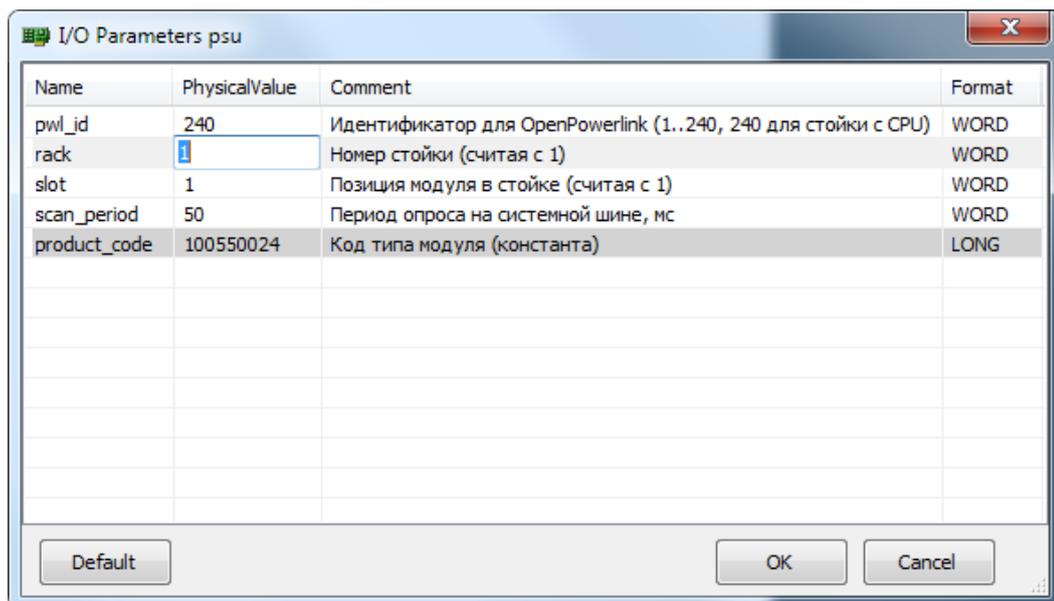


Рис. 3.7 – Окно редактирования параметров модуля изделия

Для настройки параметров модуля изделия следует после добавления устройства в «I/O Device» дважды кликнуть левой кнопкой мыши на любом из параметров модуля. В открывшемся окне редактирования ([Рис. 3.7](#)) следует настроить параметры модуля изделия, затем нажать кнопку «OK».

3.1.4 Привязка каналов модулей изделия

После добавление модуля изделия в проект и настройки его параметров следует связать его каналы с пользовательскими переменными проекта.

ВНИМАНИЕ

У абсолютно всех модулей изделия проекта (включая дополнительные устройства ввода-вывода) следует обязательно связать с пользовательскими переменными все их каналы, особенно в случае проекта с поддержкой *Failover*. Пренебрежение этим правилом приводит к постоянному выполнению циклов синхронизации и, как следствие, к завышенному и негарантированному времени выполнения программы пользователя и к повышенной на 30-40% нагрузке на процессоры модулей CPU.

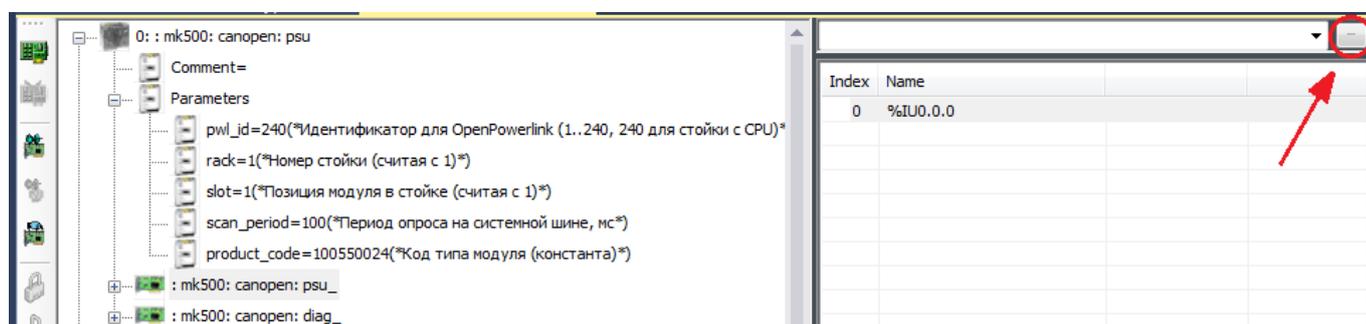


Рис. 3.8 – Кнопка «Variable Selector» (правый верхний угол)

Для привязки канала ввода-вывода модуля изделия следует выбрать в окне устройств интересующее нас простое устройство и в окне каналов дважды кликнуть левой кнопкой мыши на интересующем нас канале либо нажать кнопку «Variable Selector» ([Рис. 3.8](#)).

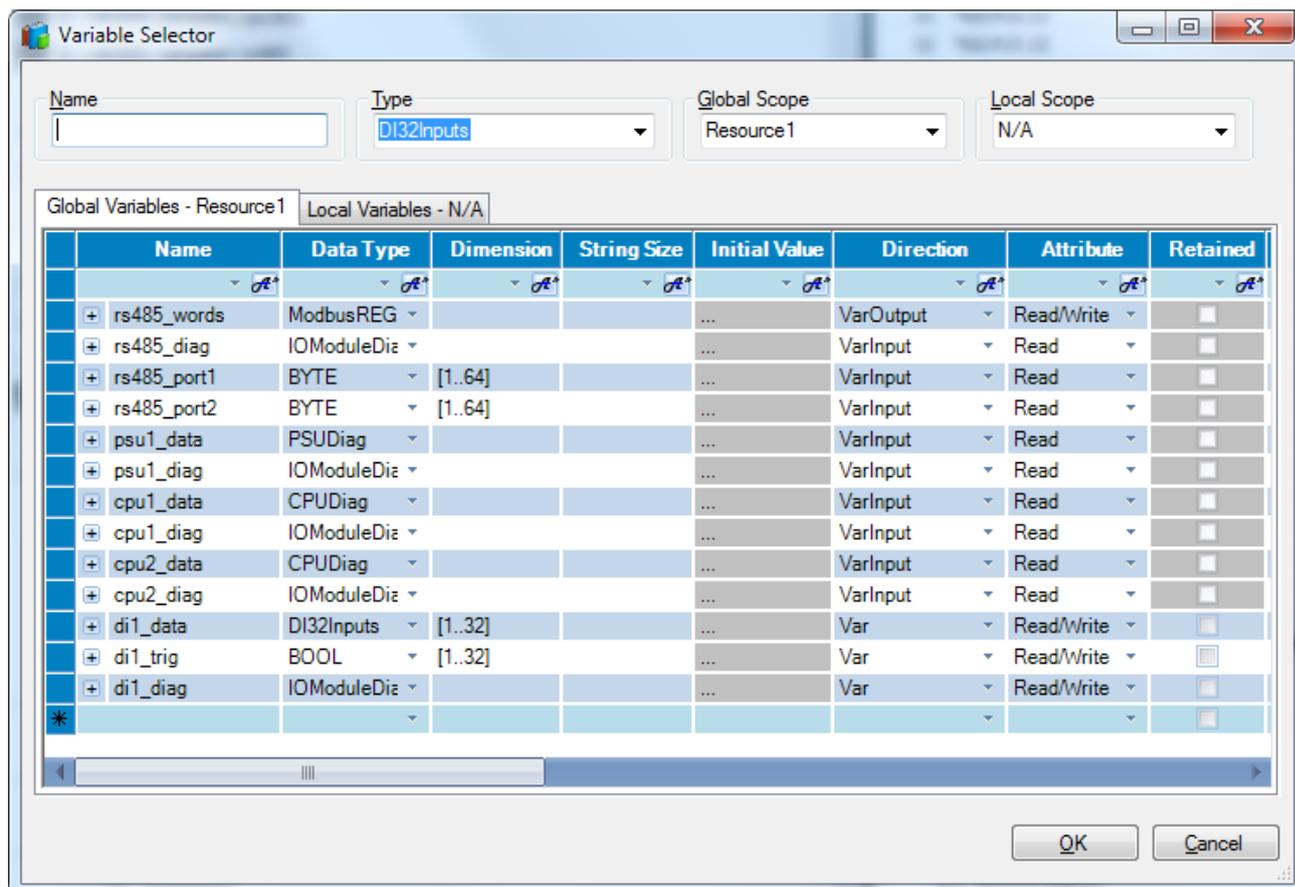


Рис. 3.9 – Окно «Variable Selector»

В открывшемся окне выбора переменной (Рис. 3.9) следует либо выбрать уже созданную переменную из списка, либо создать свою переменную, после чего нажать кнопку «ОК». Допускается выбор массива переменных, в этом случае массив будет спроецирован на последовательность каналов, начиная с выбранного.

В случае несоответствия типа и/или направления (VarInput/VarOutput) переменной типу и направлению переменной канала, будет выдано окно с предложением скорректировать тип и/или направление выбранной переменной (Рис. 3.10).

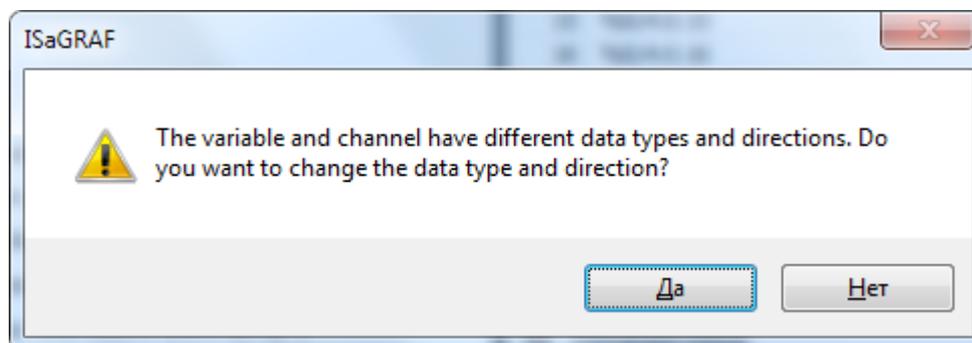


Рис. 3.10 – Предложение скорректировать тип и направление выбранной в «Variable Selector» переменной

Надо отметить, что выходные переменные каналов с направлением VarOutput создаются с атрибутом доступа Write Only. Однако для них в окне «Variable Selector» можно переключить атрибут доступа из Write only в Read/Write (Рис. 3.11), и в

дальнейшем не только записывать в канал выходные данные, но и читать из него возвращаемые значения. Тем самым реализуется отсутствующее в ISaGRAF направление VarInputOutput. Подробнее см. [п. 3.13](#).

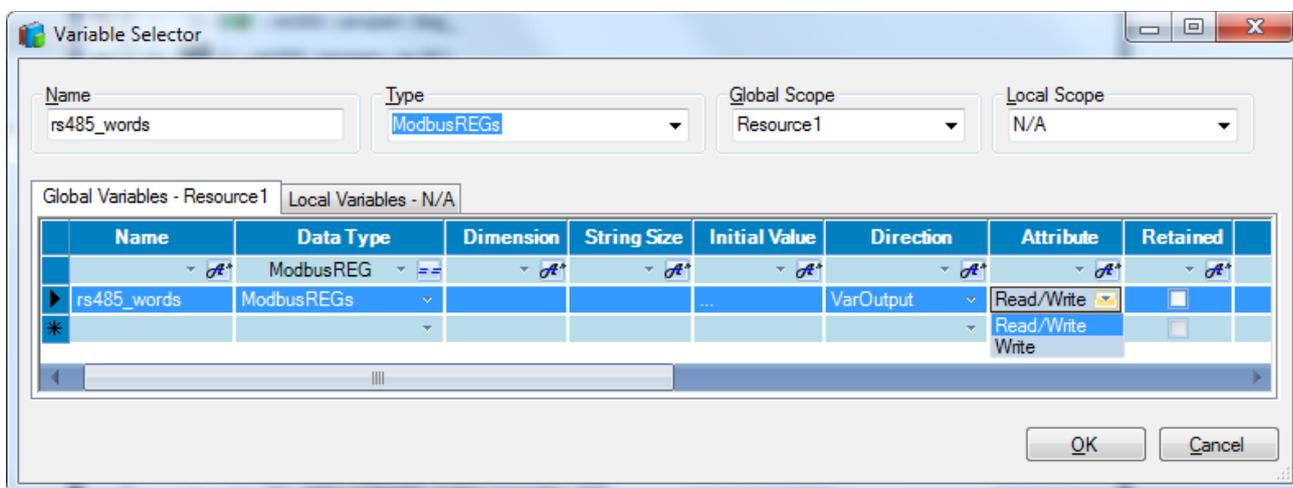


Рис. 3.11 – Смена атрибута доступа для переменной с направлением VarOutput

При необходимости отвязать переменную от канала следует выбрать канал и нажать кнопку «Free channel» на панели инструментов ([Рис. 3.12](#)).

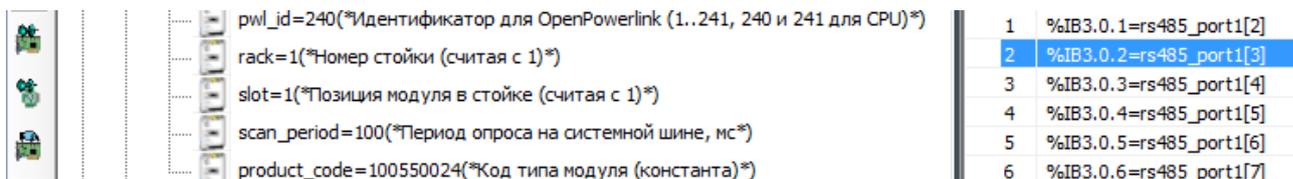


Рис. 3.12 – Кнопки «Free all channels of selected device» (слева сверху) и «Free channel» (слева посередине) панели инструментов окна I/O Device

При необходимости отвязать все переменные от всех каналов модуля следует выбрать канал и нажать кнопку «Free all channels of selected device» на панели инструментов ([Рис. 3.12](#)).

3.1.5 Настройки параметров каналов модулей изделия

Среда разработки ACP позволяет задать для каждого канала модуля изделия (в зависимости от типа данных канала) функцию преобразования значения, коэффициенты линейного масштабирования, режим инверсии, а также параметры, если они предусмотрены для этого модуля изделия. Текущие значения параметров выбранного канала отображаются в окне параметров канала ([Рис. 3.13](#)).

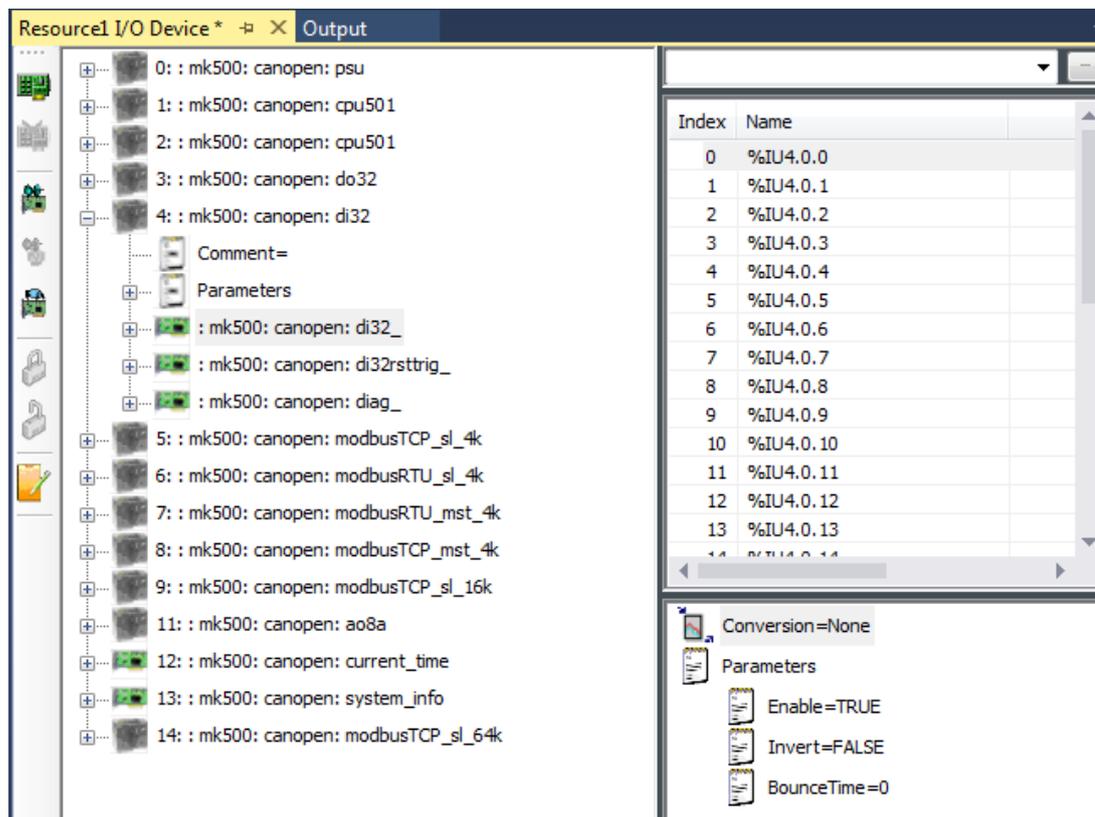


Рис. 3.13 – Пример содержимого окна параметров канала модуля дискретных входов

Функция преобразования (Conversion) присутствует для всех типов каналов, но ни для одного из модулей изделия она не определена (всегда None), поэтому в дальнейшем не упоминается.

Коэффициенты линейного масштабирования Gain и Offset могут быть заданы для каналов с целыми типами данных (BYTE, WORD). Для редактирования коэффициентов следует дважды кликнуть левой кнопкой мыши на поле Gain или Offset в окне параметров канала, затем в окне «I/O Filter» задать необходимые значения в полях Gain и Offset и нажать кнопку «ОК» ([Рис. 3.14](#)).

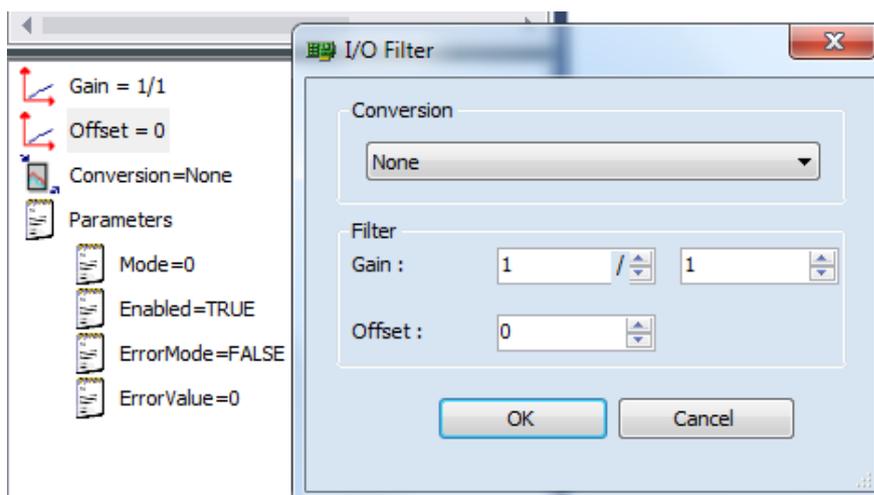


Рис. 3.14 – Окно изменения коэффициентов линейного масштабирования канала модуля аналоговых выходов

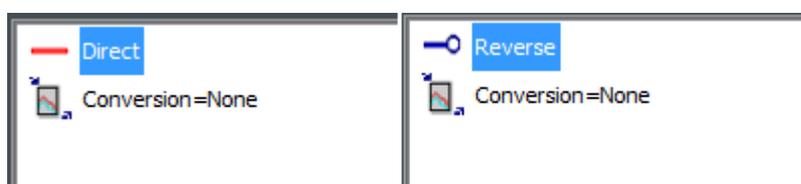


Рис. 3.15 – Применение режима инверсии к каналу модуля дискретных входов

Режим инверсии может быть применён для каналов с булевым типом данных (BOOL). Для применения режима инверсии следует в окне параметров канала дважды кликнуть левой кнопкой мыши на поле Direct, которое должно поменяться на Invert ([Рис. 3.15](#)). Повторным двойным кликом канал переводится в режим работы без инверсии.

Редактирование параметров каналов рассматривается ниже, в разделах, посвящённых работе с конкретными модулями ввода-вывода изделия.

Табл. 1.2– 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 в конфигурации с двумя модулями 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).
6	Мигает	Недопустимые положения переключателей адреса или скорости модулей питания (для всех модулей).
7	Горит	Внутренний отказ модуля.

IOModuleDiag		IOModuleDiag		
	vendorID	DWORD	▼	Идентификатор организации-производителя модуля
	productCode	DWORD	▼	Код модуля
	revisionNumber	DWORD	▼	Номер ревизии модуля
	serialNumber	DWORD	▼	Серийный номер модуля
	deviceType	DWORD	▼	Код типа модуля
	manufStatus	DWORD	▼	Регистр статуса (версия производителя)
	deviceName	STRING	▼ 32	Наименование модуля
	swVersion	STRING	▼ 16	Версия программного обеспечения модуля
	hwVersion	STRING	▼ 16	Версия аппаратного обеспечения модуля
	currentCANbus	USINT	▼	Номер текущей рабочей CAN-шины
	CAN1heartbeat	USINT	▼	Heartbeat модуля на шине CAN1
	CAN2heartbeat	USINT	▼	Heartbeat модуля на шине CAN2
	isCompatible	BOOL	▼	TRUE - если модуль на шине совместим с требуемым
	state	USINT	▼	Код состояния модуля
	crc32	DWORD	▼	Контрольная сумма метрологически значимой части ПО модуля, если она есть
	isPresent	BOOL	▼	TRUE - если модуль присутствует на шине
	isDataReady	BOOL	▼	TRUE - если данные модуля готовы

Рис. 3.17 – Описание структуры диагностики IOModuleDiag модуля изделия

3.2 Общие принципы работы с дополнительными модулями ввода-вывода

К дополнительным модулям ввода-вывода в данном документе относятся устройства ввода-вывода ISaGRAF, доступные к добавлению в окно «I/O Device» и не относящиеся к модулям изделия ([Табл. 1.3](#)). Их общая черта – отсутствие привязки к физическому устройству в составе стойки.

Табл. 1.3– Обозначения дополнительных модулей ввода-вывода в окне «Device Selector» и их функциональное назначение

Дополнительные модули ввода в окне «Device Selector»	Функциональное назначение модуля ввода-вывода
modbusRTU_mst*	Поддержка протокола Modbus RTU (ведущий) в модуле CPU (п. 3.7.1)
modbusRTU_sl*	Поддержка протокола Modbus RTU (ведомый) в модуле CPU (п. 3.7.2)
modbusTCP_mst*	Поддержка протокола Modbus TCP (ведущий) в модуле CPU (п. 3.7.1)
modbusTCP_sl*	Поддержка протокола Modbus TCP (ведомый) в модуле CPU (п. 3.7.2)
SerialPort_	Используется в составе модулей modbusRTU_mst* и modbusRTU_sl*, не рекомендуется к использованию отдельно
EthernetPort_	Используется в составе модулей modbusTCP_mst*, modbusTCP_sl* и opcua_server. Отдельно – поддержка протокола IEC 60870-5-104 (п. 3.7.3) в модуле CPU
iec104*	Поддержка протокола IEC 60870-5-104 (сервер) в модуле CPU (п. 3.7.3)
opcua_server	Поддержка протокола OPC UA (сервер) для модуля CPU МК-503-120 (п. 3.7.4)
powerlinkMN	Поддержка протокола OpenPowerlink (режим MN) для МК-503-120 (п. 3.7.5)
system_info	Системные параметры активного модуля CPU (п. 3.7.6)
mk503ext	Расширенная диагностика для модуля CPU МК-503-120 (SFP-модуль и версия аппаратной реализации CAN-портов) (п. 3.7.7.1)
mk544	Диагностика модуля расширения МК-544-040 для МК-503-120 (состояние 4-х портов Ethernet и версия аппаратной реализации Ethernet-портов) (п. 3.7.7.2)
mk546	Диагностика модуля расширения МК-546-010 для МК-503-120 (состояние 2-х портов Powerlink, состояние светодиодов модуля, версия аппаратной реализации Powerlink-портов) (п. 3.7.7.3)
current_time	Текущее время активного модуля CPU (п. 3.7.8)
ping_diag	Ping-диагностика доступности указанных серверов (п. 3.7.9)
system_diag	Приём системных диагностических сообщений модуля CPU (п. 3.7.10)

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 диагностику аппаратных модулей расширения;
- устройства, реализующие вспомогательные функции в рамках ресурса.

Все дополнительные модули ввода-вывода необходимо добавлять в проект после модулей изделия, так как дополнительные модули ввода-вывода могут в своей работе ссылаться на модули CPU ([п.3.7](#)).

Дополнительные модули ввода-вывода могут быть как комплексными, так и простыми.

В [Табл. 1.3](#) приведён перечень доступных дополнительных модулей ввода-вывода и их функциональное назначение.

3.3 Общие принципы работы с модулями центрального процессора

Наличие модулей CPU (сру501, сру502 или сру503, [Табл. 1.1](#)) в конфигурации является обязательным. Для проектов, использующих режим резервирования (Failover), необходимым является наличие в конфигурации проекта двух модулей CPU.

 **ВНИМАНИЕ**

Запрещено наличие более одного модуля CPU в конфигурации без поддержки резервирования и более двух модулей CPU в конфигурации с поддержкой резервирования. Нарушение этого правила приведёт к невозможности работы всех модулей CPU проекта.

Модули CPU обеспечивают в проекте индикацию текущего состояния программы пользователя на лицевой панели модуля CPU изделия и предоставляют диагностическую информацию о состоянии физического модуля CPU (нагрузка на процессор, состояние портов Ethernet и т.п., подробно см. [п. 3.7](#)).

Также модули CPU являются контейнерами последовательных портов и портов Ethernet, на них неявно ссылаются простые устройства `SerialPort_` и `EthernetPort_`.

3.4 Общие принципы работы с модулями ввода-вывода

Модули ввода-вывода изделия (ai16, ai16a, ai8, ai8a, ai8ahart, ai16ahart, ao8, ao8a, ao8ahart, di32, di32history, di32a, di32ahistory, do32, do32a, [Табл. 1.1](#)) обеспечивают взаимодействие программы пользователя с аналоговыми и дискретными входами и выходами физических модулей ввода-вывода. Модуль psu не обеспечивает взаимодействия с входами и выходами контроллера, но позволяет получить информацию о текущем состоянии модуля блока питания.

Скорость обновления данных в программе пользователя определяется двумя факторами: заданным временем цикла программы Cycle Time ([п. 2.5.2](#)) и периодом опроса данных модуля в его параметрах scan_period ([п. 3.1.3](#)). Для обновления данных модуля ввода-вывода в программе пользователя в каждом цикле программы период опроса модуля должен быть меньше времени цикла в 2-3 раза. При этом не рекомендуется делать период опроса всех модулей ввода-вывода слишком малым (меньше 20 мс) во избежание резкого роста нагрузки на процессор модуля CPU.

В модулях дискретного ввода (di32, di32history, di32a, di32ahistory) измеряемые данные обновляются в модуле сразу в момент изменения.

Для модулей аналогового ввода (ai16, ai16a, ai8, ai8a, ai8ahart, ai16ahart) период обновления измеряемых данных в модуле задаётся параметром scan_period.

3.5 Общие принципы работы с коммуникационными модулями

Коммуникационные модули изделия (`rs485new`, `cn545`, [Табл. 1.1](#)) обеспечивают обмен с периферийными устройствами по протоколу Modbus RTU и Powerlink.

При выборе периода опроса данных коммуникационных модулей следует руководствоваться теми же соображениями, что и для модулей ввода-вывода ([п. 3.4](#)).

Коммуникационный модуль изделия `rs485new` позволяет выполнять на каждый порт RS485 до 64 независимых команд, и имеет 960 регистров Modbus адресного пространства. Адресное пространство `rs485new` является общим для всех типов команд, и доступно через канал ввода-вывода `rs485data_` ([п. 3.13](#)).



ВНИМАНИЕ

В связи с ограниченной пропускной способностью шины CAN, допускается использование не более 8 коммуникационных модулей `rs485new` в одном проекте, с суммарным числом не более 2000 используемых регистров Modbus адресного пространства. Нарушение этой рекомендации приведёт к нестабильному обмену на шине CAN.



ВНИМАНИЕ

Следует всегда применять модуль `rs485new`. Модуль `rs485` является устаревшим, не рекомендуется к использованию. С 13.05.2022 модуль `rs485` полностью удалён из системного ПО изделия.

При работе переменной, связанной с каналом ввода-вывода `rs485data_`, следует соблюдать следующие рекомендации:

- в программе следует работать с отдельной копией данных канала ввода-вывода, не привязанной к каналу; копирование данных в рабочую переменную следует выполнять в начале цикла обработки, копировать изменённые данные рабочей переменной в связанную переменную следует в конце цикла обработки данных коммуникационного модуля.
- не следует объединять в одной переменной массива данных коммуникационного модуля выходное значение (например, код команды) и возвращаемое значение (например, результат выполнения команды). Значение этой переменной будет слишком зависеть от соотношения периодов опроса коммуникационного модуля, времени цикла программы и времени обработки этой переменной в коммуникационном модуле. В случае неверного выбора этих времён (например, время цикла программы слишком мало) значение переменной может не изменяться вообще по причине постоянного обновления его в программе ещё не обновившимся значением со стороны коммуникационного модуля.

Модули `cn545` позволяют установить время цикла обмена по Powerlink (параметр `cycleTime_ms`) и предоставляют диагностическую информацию о состоянии физического модуля МК-545 (версию прошивки, состояние портов Ethernet и количество ошибок диагностического канала, подробно см. [п. 3.14](#)).

**ВНИМАНИЕ**

Для коммуникационного модуля `cn545` приоритетным является время цикла обмена по `Powerlink`, указанное в конфигурационном файле ([п. 3.7.5](#)). Если в конфигурационном файле время цикла обмена по `Powerlink` не задано, используется время, заданное параметром `cycleTime_ms` коммуникационного модуля `cn545`.

3.6 Работа с модулями питания МК-550-024

Согласно [Табл. 1.1](#), модулю питания МК-550-024 в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия `psu`.

Кроме диагностического канала, модуль изделия `psu` имеет в своём составе канал данных модуля питания МК-550-024, реализованный в простом входном устройстве `psu_`. Данный канал имеет одну переменную-структуру типа `PSUdiag` ([Рис. 3.18](#)), предназначенную для диагностики состояния модуля питания.

Name	Data Type	Comment
PSUdiag		
PSUdiag	PSUdiag	
voltage	REAL	Выходное напряжение, В
id	WORD	ID модуля
errorCode	WORD	Код ошибки работы модуля
canBusSpeed	WORD	Скорость CAN-шины, кбит/с
can1ErrorCounter	WORD	Число ошибок на шине CAN1
can2ErrorCounter	WORD	Число ошибок на шине CAN2

Рис. 3.18 – Описание структуры данных PSUdiag

В поле `errorCode` модуль возвращает битную маску ошибок в своей работе. Расшифровка кодов ошибок работы модуля приведена в [Табл. 1.4](#).

Табл. 1.4– Расшифровка кодов ошибок errorCode структуры PSUdiag

Номер бита errorCode	Расшифровка
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

3.7 Работа с модулями центрального процессора МК-501-022, МК-502-142 и МК-503-120

Согласно [Табл. 1.1](#), модулям центрального процессора МК-501-022, МК-502-142 и МК-503-120 в среде разработки ACP Workbench ISaGRAF 6.50 соответствуют модуль изделия `cpu501`, `cpu502` и `cpu503` соответственно.

Кроме диагностического канала, модули изделия `cpu501`, `cpu502` и `cpu503` имеют в своём составе канал данных модуля CPU, реализованный в простом устройстве `cpu_`. Данный канал имеет одну переменную-структуру типа `CPUDiag` ([Рис. 3.19](#)).

Name	Data Type	Comment
CPUDiag	CPUDiag	
CPUDiag	CPUDiag	
cpuLoad	REAL	Загрузка процессора, %
memoryFree	WORD	Объём свободной памяти, МБ
ethernetPorts	CPUEthernetPorts	Массив диагностики портов Ethernet CPU
ethernetPorts[1]	EthernetPortDiag	
ethernetPorts[1].present	BOOL	TRUE если порт присутствует в модуле
ethernetPorts[1].uplink	BOOL	TRUE если порт подключен
*		
ethernetPorts[2]	EthernetPortDiag	
ethernetPorts[3]	EthernetPortDiag	
ethernetPorts[19]	EthernetPortDiag	
ethernetPorts[20]	EthernetPortDiag	
*		
uplinkPortIndex	INT	Индекс порта с ролью Uplink в массиве ethernetPorts, 0 - нет порта с ролью Uplink
datalinkPortIndex	INT	Индекс порта с ролью Datalink в массиве ethernetPorts, 0 - нет порта с ролью Datalink

Рис. 3.19 – Описание структуры данных CPUDiag

Разные модули CPU имеют в своём составе разное количество портов RS485 и Ethernet. Для унификации привязки к портам из программы пользователя было введено понятие идентификатора порта (`port_id`). В [Табл. 1.5-Табл. 1.7](#) и на [Рис. 3.20-Рис. 3.22](#) (ссылки вниз) отображено соответствие значений идентификатора порта физическим портам модулей CPU.

В случае МК-503-120 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). Это важно для корректного отображения вкладок с именами интерфейсов в окне сетевых настроек плагина ([п. 2.4](#)). При этом нумерация слотов расширения для устройств `mk544` и `mk546` – слева направо ([Рис. 3.22](#)).

Табл. 1.5– Расшифровка значений поля port_id параметров устройств SerialPort_ и EthernetPort_ модуля сру501 (МК-501-022)

Аппаратный порт	Значение port_id
Порт 1 интерфейса RS-485	1
Порт 2 интерфейса RS-485	2
Порт ETH1 интерфейса Ethernet	3
Порт ETH2 интерфейса Ethernet	4

Табл. 1.6– Расшифровка значений поля port_id параметров устройств SerialPort_ и EthernetPort_ модуля сру502 (МК-502-142)

Аппаратный порт	Значение port_id
Порт 1 интерфейса RS-485	1
Порт 2 интерфейса RS-485	2
Порт Fiber Optic интерфейса Ethernet	3
Порт ETH1 интерфейса Ethernet	4
Порт ETH2 интерфейса Ethernet	5
Порт ETH3 интерфейса Ethernet	6
Порт ETH4 интерфейса Ethernet	7

Табл. 1.7– Расшифровка значений поля port_id параметров устройств EthernetPort_ модуля сри503 (МК-503-120)

Слот шины	Аппаратный порт	Значение port_id
Центральный слот (CPU)	Порт SFP интерфейса Ethernet	3
	Порт ETН1 интерфейса Ethernet	4
	Порт ETН2 интерфейса Ethernet	5
Слот расширения L2	Порт ETН1 модуля МК-544-040	7
	Порт Powerlink модуля МК-546-010	
	Порт ETН2 модуля МК-544-040	8
	Порт ETН3 модуля МК-544-040	9
	Порт ETН4 модуля МК-544-040	10
Слот расширения L1	Порт ETН1 модуля МК-544-040	11
	Порт Powerlink модуля МК-546-010	12
	Порт ETН2 модуля МК-544-040	
	Порт ETН3 модуля МК-544-040	
	Порт ETН4 модуля МК-544-040	
Слот расширения R1	Порт ETН1 модуля МК-544-040	15
	Порт Powerlink модуля МК-546-010	16
	Порт ETН2 модуля МК-544-040	
	Порт ETН3 модуля МК-544-040	
	Порт ETН4 модуля МК-544-040	
Слот расширения R2	Порт ETН1 модуля МК-544-040	19
	Порт Powerlink модуля МК-546-010	20
	Порт ETН2 модуля МК-544-040	
	Порт ETН3 модуля МК-544-040	
	Порт ETН4 модуля МК-544-040	

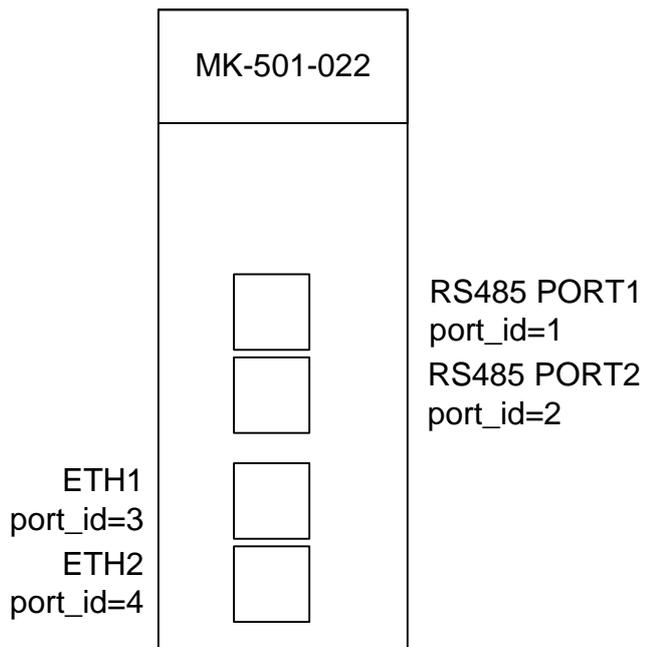


Рис. 3.20 – Схема соответствия параметра port_id физическим портам модуля МК-501-022

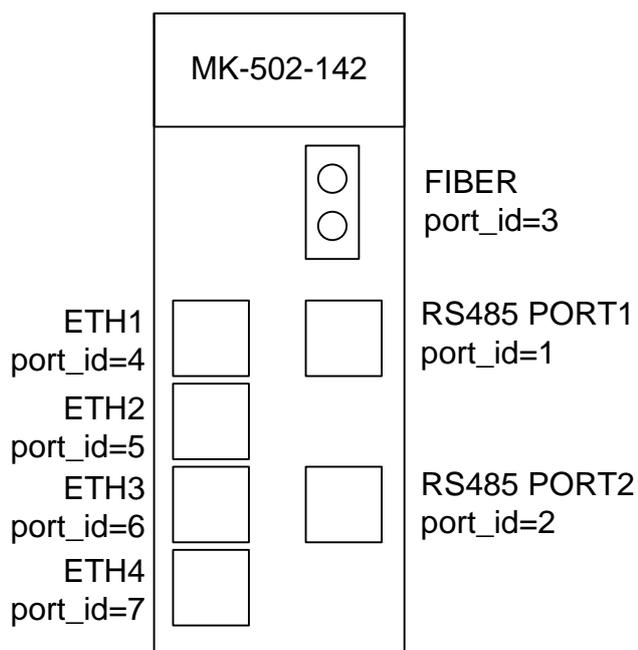


Рис. 3.21 – Схема соответствия параметра port_id физическим портам модуля МК-502-142

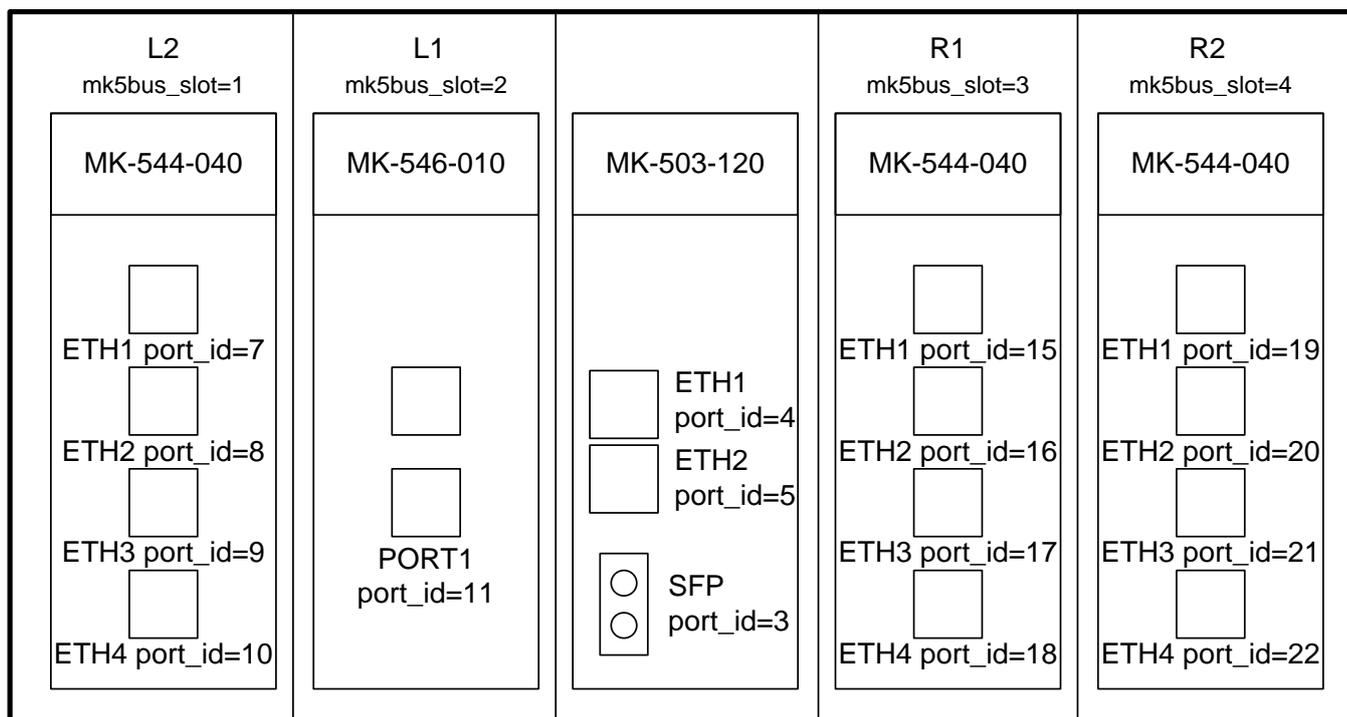
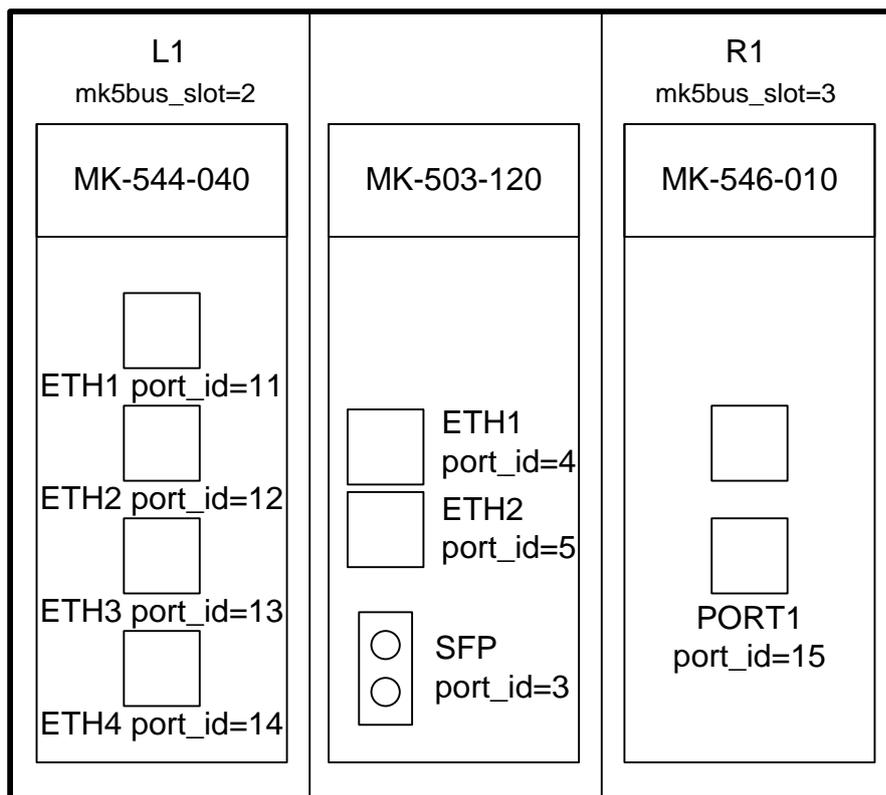


Рис. 3.22 – Схема соответствия параметра port_id и mk5bus_slot физическим портам модуля MK-503-120 в стойках с задними шинами MK-5-BUSE3 и MK-5-BUSE5

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.

3.7.1 Реализация протоколов Modbus RTU и Modbus TCP (ведущий) в модулях CPU

Для работы со встроенным в модуль CPU протоколом Modbus RTU/TCP в режиме ведущий предназначены специальные составные устройства ([Табл. 1.8](#)).

Табл. 1.8– Типы Modbus-устройств с режимом «ведущий»

Тип Modbus-устройства	Имя составного устройства	Имена простых устройств	Число регистров WORD	Число регистров BOOL	Число команд
Малое	modbusRTU_mst_4k modbusTCP_mst_4k	EthernetPort_ SerialPort_ mb_ctrl8_ mb_ctrl_diag8_ mb_regs4_ mb_bits1_ mb_control_	4096	1024	128
Среднее	modbusRTU_mst_16k modbusTCP_mst_16k	EthernetPort_ SerialPort_ mb_ctrl16_ mb_ctrl_diag16_ mb_regs16_ mb_bits4_ mb_control_	16384	4096	256
Большое	modbusRTU_mst_64k modbusTCP_mst_64k	EthernetPort_ SerialPort_ mb_ctrl32_ mb_ctrl_diag32_ mb_regs64_ mb_bits16_ mb_control_	65536	16384	512

Составные Modbus-устройства с поддержкой режима «ведущий» включают в себя простые устройства, с помощью которых реализуется привязка к аппаратному порту, адресное пространство, управление и диагностика устройства. Составные Modbus-устройства созданы трёх типов: малое, среднее и большое, отличаясь доступным размером пространства регистров и числом команд.

Для ускорения синхронизации данных устройств Modbus RTU/TCP при работе с включённым режимом Failover, рекомендуется выбирать составные Modbus-устройства с количеством регистров, максимально соответствующим задаче (то есть для задачи с потребностью в 1000 регистров WORD выбирать малое Modbus-устройство, а не среднее или большое).

В состав Modbus-устройства с поддержкой режима «ведущий» входят следующие простые устройства:

3.7.1.1 Порт последовательный (SerialPort_) или Ethernet-порт (EthernetPort_)

Входное простое устройство `SerialPort_` или `EthernetPort_` предназначено для привязки Modbus-устройства к аппаратному порту модуля CPU. Устройство имеет 1 канал типа `CPUPortStatus` (Рис. 3.23), предназначенный для диагностики работы порта (Табл. 1.9). Через параметры этого простого устройства (Рис. 3.24 и Рис. 3.25) выполняется настройка физического порта модуля CPU.

Name	Data Type	String Size	Comment
CPUPortStatu			
CPUPortStatus	CPUPortStatus		
mode	INT		Режим работы порта (0 - свободен, 1 - Modbus Slave, 2 - Modbus Master, 3 - IEC 60870-5-104 Server, 4 - OPC-UA Server)
status	INT		Текущее состояние порта
parameters	STRING	80	Строка параметров порта

Рис. 3.23 – Описание структуры `CPUPortStatus`

Табл. 1.9– Расшифровка значений поля `status` структуры `CPUPortStatus`

Значение <code>status</code>	Расшифровка
1	Порт инициализован и готов к работе
0	Порт инициализован, но не в работе
-1	Ошибка инициализации библиотеки Modbus
-2	Ошибка подключения к порту (в режиме ведомого)
-3	Ошибка выделения памяти
-4	Ошибка работы с портом
-5	Системная ошибка
-6	Ошибка выделения памяти в режиме ведущего
-7	Не инициализован массив команд в режиме ведущего
-8	Порт уже занят

Для привязки к аппаратному порту модуля CPU следует в параметрах устройства `SerialPort_` или `EthernetPort_` настроить поле `port_id` согласно Табл. 1.5-Табл. 1.7.

Так как поле `port_id` ссылается на аппаратный порт в составе модуля CPU, следует в ходе ранжирования модулей (п. 3.1.2) помещать Modbus-устройство ниже модулей CPU, в противном случае Modbus-устройство работать не будет.

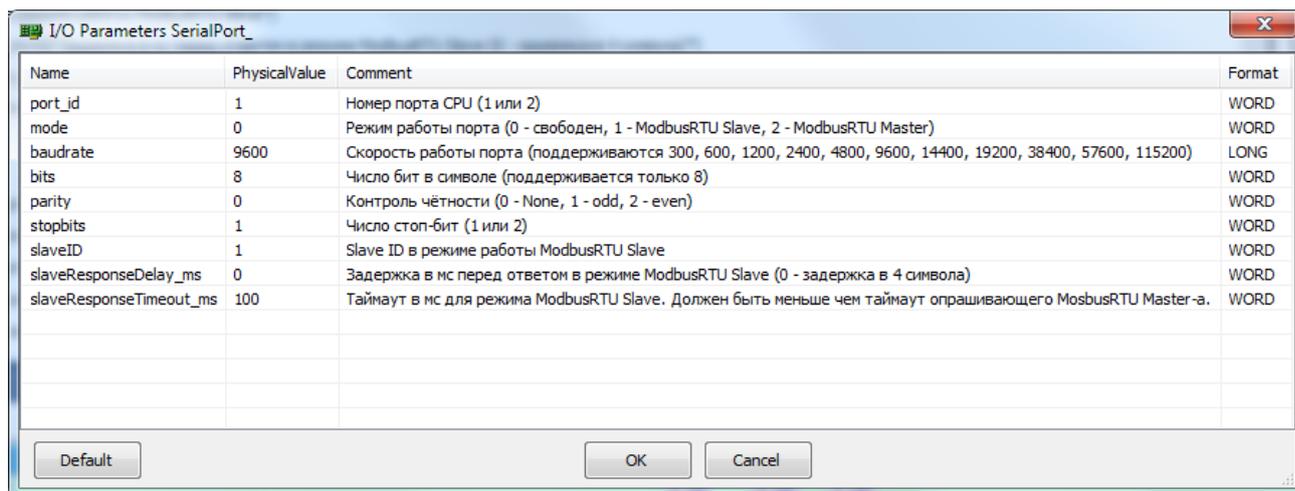


Рис. 3.24 – Описание параметров устройства SerialPort_

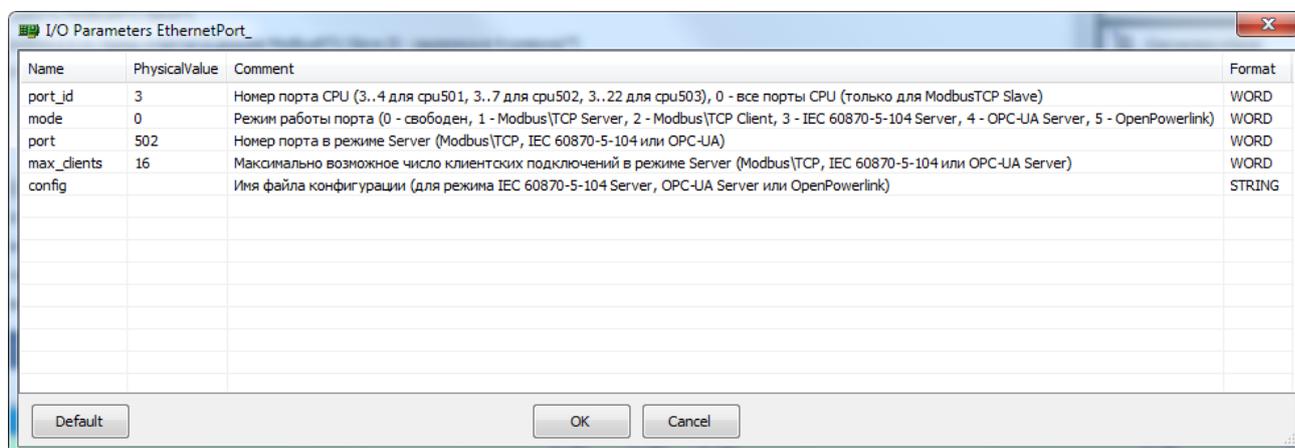


Рис. 3.25 – Описание параметров устройства EthernetPort_

Для работы порта в режиме Modbus в режиме ведущего следует в параметрах устройства SerialPort_ или EthernetPort_ настроить поле mode равным 2 (Табл. 1.10).

При работе с устройством SerialPort_ следует обратить внимание на поля baudrate, bits, parity и stopbits параметров устройства и настроить их согласно настройками ведомых устройств ModbusRTU.

Поля slaveID, slaveResponseDelay_ms и slaveResponseTimeout_ms в параметрах устройства SerialPort_ для устройства ModbusRTU в режиме ведущего значения не имеют.

Поля port, max_clients, slaveID и config в параметрах устройства EthernetPort_ для устройства Modbus TCP в режиме ведущего значения не имеют.

Табл. 1.10– Расшифровка значений поля mode параметров устройств SerialPort_ и EthernetPort_

Значение mode	Тип порта	Режим работы порта
0	SerialPort_ EthernetPort_	Порт свободен

Значение mode	Тип порта	Режим работы порта
1	SerialPort_	ModbusRTU в режиме ведомый
	EthernetPort_	Modbus TCP в режиме ведомый
2	SerialPort_	ModbusRTU в режиме ведущий
	EthernetPort_	Modbus TCP в режиме ведущий
3	SerialPort_	Не поддерживается, порт свободен
	EthernetPort_	IEC 60870-5-104 в режиме сервера
4	SerialPort_	Не поддерживается, порт свободен
	EthernetPort_	OPC UA в режиме сервера

3.7.1.2 Устройство управления (mb_ctrl*)

Выходное простое устройство управления `mb_ctrl*` (Табл. 1.8), предназначено для передачи команд в драйвер Modbus. Устройство имеет 8, 16 или 32 канала типа `ModbusMasterRequests` (структура-массив из 16 переменных типа `ModbusMasterRequest`, Рис. 3.26). Через параметры этого простого устройства (Рис. 3.29) настраивается время цикла опроса драйвером Modbus ведомых устройств.

Name	Data Type	String Size
ModbusMasterRequests	ModbusMasterRequests	
- ModbusMasterRequests	ModbusMasterRequests	
- Requests	ModbusMasterReqArr	
+ Requests[1]	ModbusMasterRequest	
+ Requests[2]	ModbusMasterRequest	
+ Requests[15]	ModbusMasterRequest	
+ Requests[16]	ModbusMasterRequest	

Рис. 3.26 – Описание структуры `ModbusMasterRequests`

Name	Data Type	String Size	Comment
ModbusMasterRequ	ModbusMasterRequest		
ModbusMasterRequest	ModbusMasterRequest		
enable	BOOL		TRUE - запрос выполняется, FALSE - пропускается
single_request	BOOL		TRUE - запрос выполняется единожды, FALSE - запрос выполняется непрерывно
on_modify_request	BOOL		TRUE - запрос (на запись) выполняется автоматически при изменении данных (т
do_single_req	BOOL		TRUE - выполняет одиночный запрос, сбрасывается в FALSE по выполнении
+ command	ModbusMasterCommand		Собственно выполняемая команда Modbus Master

Рис. 3.27 – Описание структуры `ModbusMasterRequest`

Структура `ModbusMasterRequest` (Рис. 3.27) состоит из полей управления `enable`, `single_request`, `on_modify_request` и `do_single_req`, и собственно команды Modbus, поля `command` типа `ModbusMasterCommand` (Рис. 3.28).

Поле `enable` служит для включения и отключения запросов в ходе выполнения программы пользователя. Для разрешения работы запроса должно быть установлено в 1. Рекомендуется явно сбрасывать его в 0 для всех неиспользуемых запросов.

Поле `single_request` определяет режим отправки запроса: 0 – непрерывная отправка каждый цикл, 1 – отправка по установке поля `do_single_req` в 1. Сброс значения поля `do_single_req` в 0 должен выполняться в программе пользователя.

Поле `on_modify_request` не поддерживается в драйвере Modbus для модуля CPU, но поддерживается модулем МК-541-002 (п. 3.13).

Name	Data Type	String Size	Comment
ModbusMasterCommand	ModbusMasterCommand		
slave_tcp_address	STRING	20	IPv4 адрес запрашиваемого устройства (для Modbus\TCP)
slave_tcp_port	WORD		tcp-порт запрашиваемого устройства (для Modbus\TCP)
slave_id	BYTE		Slave ID запрашиваемого устройства
func_code	BYTE		Код modbus-функции. Поддерживаются функции 1, 2, 3, 4, 5, 6, 15 и 16.
slave_data_addr	WORD		Адрес данных в запрашиваемом устройстве
slave_data_length	WORD		Размер данных в запрашиваемом устройстве
offset	WORD		Смещение от начала массива данных, связанных с Modbus Master.
timeout_ms	WORD		Таймаут одного запроса, в мс
delay_before_ms	WORD		Задержка перед выполнением запроса (0 - задержка в 4 символа), в мс
repeats	WORD		Число повторов запроса в случае неудачи
skip_repeats_when_bad	WORD		Число запросов, которое пропускается перед опросом не отвечающего устройства
repeat_over_scan	BOOL		TRUE - повтор неудачного запроса выполнять через скан, FALSE - повтор выполнять сразу

Рис. 3.28 – Описание структуры ModbusMasterCommand

Структура `ModbusMasterCommand` рассчитана на работу как с Modbus RTU и с Modbus TCP, ненужные в конкретном случае поля игнорируются драйвером Modbus.

Параметр `cycle_time_ms` в параметрах устройства управления служит для настройки полного времени цикла опроса ведомых устройств драйвером Modbus. В случае, если все команды успевают выполняться за указанное время, драйвер будет ждать истечения указанного времени до повторной отправки первого запроса. В случае превышения указанного времени либо в случае установки указанного параметра в 0 первый запрос отправляется повторно сразу после завершения последнего.

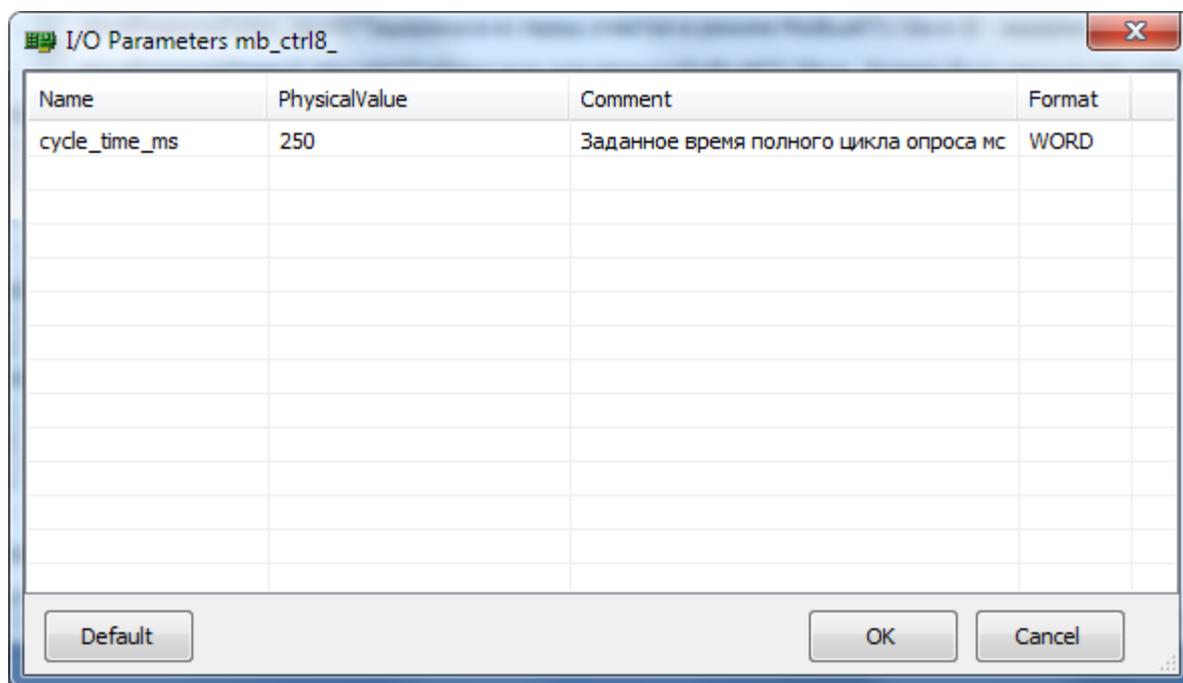


Рис. 3.29 – Описание параметров устройства mb_ctrl* модуля CPU изделия

3.7.1.3 Устройство диагностики команд (mb_ctrl_diag*)

Входное простое устройство диагностики команд mb_ctrl_diag* (Табл. 1.8), предназначено для получения статуса выполнения команд из драйвера Modbus. Устройство имеет 8, 16 или 32 канала типа modbusMasterRequestsAndStats (структура-массив из 16 переменных типа ModbusMasterReqStatus, Рис. 3.30).

Name	Data Type	String Size
ModbusMasterRequestsAndStats		
modbusMasterRequestsAndStats	modbusMasterRequestsAndStats	
RequestsAndStats	ModbusMasterReqStatArr	
RequestsAndStats[1]	ModbusMasterReqStatus	
RequestsAndStats[2]	ModbusMasterReqStatus	
RequestsAndStats[15]	ModbusMasterReqStatus	
RequestsAndStats[16]	ModbusMasterReqStatus	

Рис. 3.30 – Описание структуры modbusMasterRequestsAndStats

Name	Data Type	String Size	Comment
ModbusMasterReqStatus			
ModbusMasterReqStatus	ModbusMasterReqStatus		
status_request	INT		Обновляющийся код состояния запроса
status_exectime_ms	INT		Обновляющееся последнее время выполнения запроса, в мс
status_repeats	UINT		Обновляющееся число выполненных повторных запросов (при неудаче)
status_skips	UINT		Обновляющееся число пропусков в опросе для не отвечающего устройства
requests_total	DWORD		Общее число выполненных запросов
requests_good	DWORD		Число запросов, выполненных без ошибок

Рис. 3.31 – Описание структуры ModbusMasterReqStatus

Табл. 1.11– Расшифровка значений поля status_request структуры MosbusMasterReqStatus

Значение status_request	Расшифровка
0 либо 100	Запрос выполнен успешно (для устройства rs485new коды 0 и 100 чередуются)
1	Неверная функция
2	Неверный адрес данных
3	Неверное значение данных
4	Общий сбой устройства сервера
5	Ведомое устройство приняло запрос и обрабатывает его, но это требует много времени
6	Устройство сервера занято
7	Ведомое устройство не может выполнить программную функцию, заданную в запросе
8	Данный код не поддерживается
9	Не определено стандартом

Значение status_request	Расшифровка
10	Данный код не поддерживается
11	Нет ответа от целевого устройства либо оно отсутствует в сети (для Modbus TCP)
12	Контрольная сумма не совпала
13	Данные ответа не соответствуют запросу
14	Неизвестный код ошибки
15	Код ошибки не соответствует запросу
16	При корректной CRC данных в ответе больше, чем 123 для регистров и 2000 для битов
17	Ответ от ведомого устройства с другим Slave ID
20	Нет ответа по истечении заданного времени (таймаут)
21	Неправильно заполнены поля структуры запроса
22	Ошибка инициализации библиотеки Modbus
23	Не удалось соединиться с Modbus TCP-сервером
24	Передано/принято данных меньше, чем следовало
25	Адрес/длина принимаемых/передаваемых данных выходят за пределы связанных с устройством массивов регистров/битов
30	Запрос обновляется (передаётся по CAN из модуля CPU в модуль RS485). Только для устройства rs485new
31	Ошибка обновления запроса модуля RS485 (при этой ошибке следует повторить попытку обновления). Только для устройства rs485new
254	Запрос обрабатывается
255	Запрос ещё ни разу не обрабатывался

Структура `ModbusMasterReqStatus` ([Рис. 3.31](#)) содержит в себе диагностические и статистические данные по Modbus-запросу:

- поле `status_request` ([Табл. 1.11](#)) содержит информацию о ходе выполнения запроса;
- поле `status_exeptime_ms` содержит информацию о времени выполнения последнего запроса, обновляется после завершения выполнения запроса;
- поле `status_repeats` содержит число повторов запроса при ошибке в ходе выполнения, сбрасывается в 0 при успешном выполнении запроса;
- поле `status_skips` содержит число пропусков запроса при достижении полем `status_repeats` значения, определённого параметром `repeats` в настройках запроса ([Рис. 3.31](#)), сбрасывается в 0 при успешном выполнении запроса;
- поле `requests_total` содержит общее число запросов, отправленных за всё время работы программы пользователя, сбрасывается в 0 только при рестарте программы пользователя;

- поле `requests_good` содержит число запросов, отправленных и успешно обработанных за всё время работы программы пользователя, сбрасывается в 0 только при рестарте программы пользователя.

3.7.1.4 Устройства регистровых и битных переменных (mb_regs* и mb_bits*)

Выходные простые устройства регистровых и битных переменных (mb_regs* и mb_bits*, Табл. 1.8) предназначены для хранения отправляемых и получаемых командами Modbus данных. В начале цикла выполнения программы каналы этих устройств содержат в себе обновившиеся в ходе обработки команд Modbus данные. Поэтому для связанных с этими устройствами переменными следует устанавливать атрибут Read/Write (п. 3.1.4), а в начале каждого цикла следует копировать значения переменных в отдельные, не связанные с каналами рабочие переменные.

Устройство mb_regs* имеет 4, 16 или 64 переменных-структуры типа ModbusREGs (массив из 1024 переменных типа WORD, Рис. 3.32). Устройство mb_bits* имеет 1, 4 или 16 переменных-структуры типа ModbusDiscrets (массив из 1024 переменных типа BOOL, Рис. 3.33).

Через параметр offset устройства (Рис. 3.34) передается настройка смещения адресного пространства, которая учитывается при формировании команд.

Name		Data Type	String Size	Comment
ModbusREGs				
ModbusREGs		ModbusREGs		
regs		Words1k		Массив регистров Modbus
	regs[1]	WORD		
	regs[2]	WORD		
	regs[3]	WORD		
	regs[4]	WORD		
	regs[5]	WORD		
	regs[6]	WORD		
	regs[7]	WORD		
	regs[8]	WORD		
	regs[9]	WORD		
	regs[10]	WORD		
	regs[1020]	WORD		
	regs[1021]	WORD		
	regs[1022]	WORD		
	regs[1023]	WORD		
	regs[1024]	WORD		
	*			

Рис. 3.32 – Описание структуры ModbusREGs

Name	Data Type	String Size	Comment
ModbusDiscrets			
ModbusDiscrets	ModbusDiscrets		
discrets	Bools1k		Массив дискретов modbus
discrets[1]	BOOL		
discrets[2]	BOOL		
discrets[3]	BOOL		
discrets[4]	BOOL		
discrets[5]	BOOL		
discrets[6]	BOOL		
discrets[7]	BOOL		
discrets[8]	BOOL		
discrets[9]	BOOL		
discrets[10]	BOOL		
discrets[1020]	BOOL		
discrets[1021]	BOOL		
discrets[1022]	BOOL		
discrets[1023]	BOOL		
discrets[1024]	BOOL		

Рис. 3.33 – Описание структуры ModbusDiscrets



ВНИМАНИЕ

Настоятельно рекомендуется формировать групповые запросы на чтение и запись переменных типа Discrete Inputs и Coil (функции 1, 2 и 15) так, чтобы данные одного запроса на стороне модуля CPU укладывались в значение одного поля ModbusDiscrets. Например, при необходимости прочитать 1500 Coils в первом запросе на чтение указывать смещение 0 и запрашивать 1024 Coils, во втором запросе на чтение указывать смещение 1024 и запрашивать 476 Coils.

3.7.1.5 Устройство управления работой драйвера Modbus (mb_control_)

Выходное простое устройство mb_control_ предназначено для управления работой драйвера Modbus. Устройство имеет 1 канал типа ВУТЕ, в который следует записывать команду управления. Перечень допустимых значений команд приведён в [Табл. 1.12](#).

Табл. 1.12– Расшифровка значений выхода устройства mb_control_

Значение выхода mb_control_	Имя Defined Word	Расшифровка команды
0	CPUPORT_STOP	Остановить работу драйвера Modbus
1	CPUPORT_START	Запустить драйвер Modbus в работу

3.7.2 Реализация протоколов Modbus RTU и Modbus TCP (ведомый) в модулях CPU

Для работы со встроенным в модуль CPU протоколом Modbus RTU/TCP в режиме «ведомый» предназначены специальные составные устройства (Табл. 1.13).

Составные Modbus-устройства с поддержкой режима «ведомый» включают в себя простые устройства, с помощью которых реализуется привязка к аппаратному порту, адресное пространство, управление и диагностика устройства. Составные Modbus-устройства созданы трёх типов: малое, среднее и большое, отличаясь доступным размером пространства регистров.

Табл. 1.13– Типы Modbus устройств с режимом «ведомый»

Тип Modbus-устр.	Имя составного устройства	Имена простых устройств	Число Holding Registers	Число Input Registers	Число Coils	Число Discrete Inputs
Малое	modbusRTU_sl_4k modbusTCP_sl_4k	EthernetPort_ SerialPort_ mb_HR4_ mb_DI1_ mb_IR1_ mb_Coils1_ mb_control_	4096	1024	1024	1024
Среднее	modbusRTU_sl_16k modbusTCP_sl_16k	EthernetPort_ SerialPort_ mb_HR16_ mb_DI4_ mb_IR4_ mb_Coils4_ mb_control_	16384	4096	4096	4096
Большое	modbusTCP_slh_64k	EthernetPort_ mb_HR64_ mb_control_	65536	0	0	0
Большое	modbusRTU_sl_64k modbusTCP_sl_64k	EthernetPort_ SerialPort_ mb_HR64_ mb_DI16_ mb_IR16_ mb_Coils16_ mb_control_	65536	16384	16384	16384

Для ускорения синхронизации данных устройств Modbus RTU/TCP при работе с включённым режимом Failover, рекомендуется выбирать составные Modbus-устройства с количеством регистров, максимально соответствующим задаче (то есть для задачи с потребностью в 1000 Holding Register выбирать малое Modbus-устройство, а не среднее или большое).

В состав Modbus-устройства с поддержкой режима ведомый входят следующие простые устройства:

3.7.2.1 Порт последовательный (SerialPort_) или Ethernet-порт (EthernetPort_)

Входное простое устройство порт последовательный (SerialPort_) или Ethernet (EthernetPort_) предназначено для привязки Modbus-устройства к аппаратному порту модуля CPU. Его назначение и параметры полностью аналогичны описанным в [п. 3.7.1](#) для Modbus-устройств с поддержкой режима «ведущий».

Поле `port` в параметрах устройства EthernetPort_ следует установить согласно настройкам ведущего устройства Modbus TCP (по умолчанию 502).

Для привязки к аппаратному порту модуля CPU следует в параметрах устройства SerialPort_ или EthernetPort_ настроить поле `port_id` согласно [Табл. 1.5-Табл. 1.7](#).

Для привязки устройства ModbusTCP сразу ко всем аппаратным портам модуля CPU следует записать 0 в поле `port_id`. При этом будут заняты TCP-порты (поле `port` в параметрах устройства EthernetPort_) на всех аппаратных портах модуля CPU. В таком случае при добавлении в проект второго и более устройств ModbusTCP в режиме «ведомый» следует установить для них другой TCP-порт. При выборе нестандартных значений портов рекомендуется выбирать значения в диапазоне 1024...65535.

Так как поле `port_id` ссылается на аппаратный порт в составе модуля CPU, следует в ходе ранжирования модулей ([п. 3.1.2](#)) помещать Modbus-устройство ниже модулей CPU, в противном случае Modbus-устройство работать не будет.

Для работы порта в режиме Modbus в режиме ведомого следует в параметрах устройства SerialPort_ или EthernetPort_ настроить поле `mode` равным 1 ([Табл. 1.10](#)).

Значение поля `slaveID` следует настроить согласно требованиям к реализуемому Modbus-устройству. Значения 0 и 248..255 являются запрещёнными. При этом на резервном CPU `slaveID` устройства ModbusRTU будет автоматически увеличен на 32, с учётом допустимости значений `slaveID` (см. [Табл. 1.14](#)). При переключении между активным и резервным модулями CPU значения `slaveID` меняются соответственно.

Табл. 1.14– Таблица смещений адресов slaveID устройства ModbusRTU для резервного CPU

CPU	slaveID			
Primary	1..215	216..247	248..255	0
Secondary	33.. 247	1..32	0	0

При работе с устройством SerialPort_ следует обратить внимание на поля `baudrate`, `bits`, `parity` и `stopbits` параметров устройства и настроить их сообразно настройками ведущего устройства ModbusRTU.

Поле `slaveResponseDelay_ms` в параметрах устройства SerialPort_ для устройства ModbusRTU в режиме работы «ведомый» задаёт длительность паузы в миллисекундах перед ответом на запрос со стороны ведущего (0 – автоматически вычисляемая пауза длительностью в 4 символа).

Поле `slaveResponseTimeout_ms` в параметрах устройства SerialPort_ для устройства ModbusRTU в режиме работы «ведомый» задаёт величину таймаута на запрос со стороны ведущего. Его величина должна быть меньше значения таймаута на запрос в устройстве ведущего, в противном случае есть вероятность несогласованной работы драйвера Modbus.

Поле `max_clients` в параметрах устройства `EthernetPort_` следует установить равным предполагаемому числу подключаемых устройств Modbus TCP в режиме ведущий (не более 255, значение 0 трактуется как 1).

Поле `config` в параметрах устройства `EthernetPort_` для устройства Modbus в режиме ведомого значения не имеет.

3.7.2.3 Устройство управления работой драйвера Modbus (mb_control_)

Выходное простое устройство `mb_control_` предназначено для управления работой драйвера Modbus. Данный канал имеет одну переменную типа `BYTE`, в которую следует записывать команду управления работой драйвера Modbus. Перечень допустимых значений команд приведён в [Табл. 1.12](#).



ВНИМАНИЕ

В настоящее время в устройствах Modbus в режиме «ведомый» устройство `mb_control_` не используется, драйвер Modbus всегда активен.

3.7.3 Реализация протокола IEC 60870-5-104 (сервер) в модулях CPU

Для работы со встроенным в модуль CPU протоколом IEC 60870-5-104 в режиме сервера предназначены следующие простые устройства ввода-вывода:

- простое устройство `EthernetPort_`, с помощью которого реализуется привязка сервера IEC 60870-5-104 к аппаратному порту;
- специальные простые устройства ([Табл. 1.15](#) и [Табл. 1.16](#)), с помощью которых реализуется адресное пространство IEC 60870-5-104;
- простое устройство `iec104_server_diag_`, с помощью которого реализуется диагностика работы сервера IEC 60870-5-104.

Табл. 1.15– iec104*-устройства типа output (выходные данные)

Имя устройства	Типы IEC 60870-5-104		
	Имя	Номер	Назначение
<code>iec104_SP_out_</code>	<code>M_SP_NA_1</code>	1	Одноэлементная информация
<code>iec104_DP_out_</code>	<code>M_DP_NA_1</code>	3	Двухэлементная информация
<code>iec104_ST_out_</code>	<code>M_ST_NA_1</code>	5	7 бит значащей информации
<code>iec104_BO_out_</code>	<code>M_BO_NA_1</code>	7	Строка из 32 бит
<code>iec104_MEa_out_</code>	<code>M_ME_NA_1</code>	9	Значение измеряемой величины, нормализованное значение
<code>iec104_MEb_out_</code>	<code>M_ME_NB_1</code>	11	Значение измеряемой величины, масштабированное значение
<code>iec104_MEc_out_</code>	<code>M_ME_NC_1</code>	13	Значение измеряемой величины, короткий формат с плавающей запятой
<code>iec104_IT_out_</code>	<code>M_IT_NA_1</code>	15	Интегральная сумма
<code>iec104_MEd_out_</code>	<code>M_ME_ND_1</code>	21	Значение измеряемой величины, нормализованное значение без описания качества
<code>iec104_SP_TS_out_</code>	<code>M_SP_TB_1</code>	30	Одноэлементная информация с меткой времени
<code>iec104_DP_TS_out_</code>	<code>M_DP_TB_1</code>	31	Двухэлементная информация с меткой времени
<code>iec104_ST_TS_out_</code>	<code>M_ST_TB_1</code>	32	7 бит значащей информации с меткой времени
<code>iec104_BO_TS_out_</code>	<code>M_BO_TB_1</code>	33	32-битная строка с меткой времени
<code>iec104_MEa_TS_out_</code>	<code>M_ME_TD_1</code>	34	Значение измеряемой величины, нормализованное значение с меткой времени
<code>iec104_MEb_TS_out_</code>	<code>M_ME_TE_1</code>	35	Значение измеряемой величины, масштабированное значение с меткой времени

Имя устройства	Типы IEC 60870-5-104		
	Имя	Номер	Назначение
iec104_MEc_TS_out_	M_ME_TF_1	36	Значение измеряемой величины, короткий формат с плавающей запятой с меткой времени
iec104_IT_TS_out_	M_IT_TB_1	37	Интегральная сумма с меткой времени
iec104_EP_TS_out_	M_EP_TD_1	38	Действия устройств защиты с меткой времени
iec104_T230_out_	M_IT_ND_1	230	Интегральная сумма, длинный формат
iec104_T231_out_	M_IT_TD_1	231	Интегральная сумма, длинный формат с меткой времени

Табл. 1.16– iec104*-устройств типа input (команды)

Имя устройства	Типы IEC 60870-5-104		
	Имя	Номер	Назначение
iec104_SP_in_	C_SC_NA_1	45	Одноэлементная команда
iec104_DP_in_	C_DC_NA_1	46	Двухэлементная команда
iec104_RC_in_	C_RC_NA_1	47	Команда пошагового регулирования
iec104_MEa_in_	C_SE_NA_1	48	Команда уставки, нормализованное значение
iec104_MEb_in_	C_SE_NB_1	49	Команда уставки, масштабированное значение
iec104_MEc_in_	C_SE_NC_1	50	Команда уставки, короткий формат с плавающей запятой
iec104_BO_in_	C_BO_NA_1	51	Строка из 32 бит, команда
iec104_SP_TS_in_	C_SC_TA_1	58	Одноэлементная команда с меткой времени
iec104_DP_TS_in_	C_DC_TA_1	59	Двухэлементная команда с меткой времени
iec104_RC_TS_in_	C_RC_TA_1	60	Команда пошагового регулирования с меткой времени
iec104_MEa_TS_in_	C_SE_TA_1	61	Команда уставки, нормализованное значение с меткой времени
iec104_MEb_TS_in_	C_SE_TB_1	62	Команда уставки, масштабированное значение с меткой времени
iec104_MEc_TS_in_	C_SE_TC_1	63	Команда уставки, короткий формат с плавающей запятой с меткой времени
iec104_BO_TS_in_	C_BO_TA_1	64	Строка из 32 бит, команда с меткой времени

Для ускорения синхронизации данных устройств IEC 60870-5-104 при работе с включённым режимом Failover, рекомендуется при добавлении устройств iec104*

указывать реально используемое число каналов вместо предлагаемого по умолчанию значения 512 ([Рис. 3.36](#)).

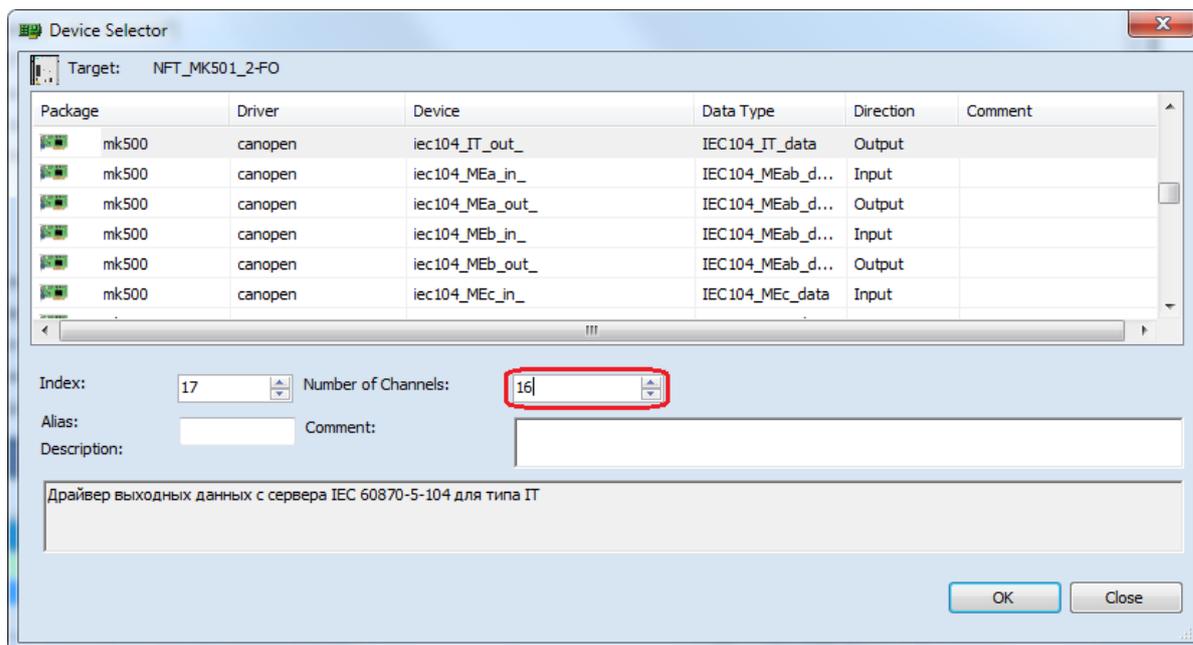


Рис. 3.36 – Выбор необходимого числа каналов при добавлении устройства iec104*

Для поддержки сервера IEC 60870-5-104 в проект следует добавить простые устройства, перечисленные в [пп.3.7.3.1](#) – [пп.3.7.3.3](#), желательно в порядке перечисления.

3.7.3.1 Ethernet-порт (EthernetPort_)

Входное устройство порта Ethernet (EthernetPort_), предназначенное для привязки сервера IEC 60870-5-104 к аппаратному порту модуля CPU. Его назначение и параметры полностью аналогичны описанным в [п. 3.7.1](#).

Поле `port` в параметрах устройства EthernetPort_ следует установить согласно настройкам клиента IEC 60870-5-104. Стандартным значением является 2404 (по умолчанию установлено 502).

Для привязки к аппаратному порту модуля CPU следует в параметрах устройства EthernetPort_ настроить поле `port_id` согласно [Табл. 1.5-Табл. 1.7](#).

Для привязки сразу ко всем аппаратным портам модуля CPU следует записать 0 в поле `port_id`. При этом будут заняты TCP-порты (поле `port` в параметрах устройства EthernetPort_) на всех аппаратных портах модуля CPU. В таком случае при добавлении в проект второго сервера IEC 60870-5-104 следует установить для него другой TCP-порт.

При выборе нестандартных значений портов рекомендуется выбирать значения в диапазоне 1024...65535.

Так как поле `port_id` ссылается на аппаратный порт в составе модуля CPU, следует в ходе ранжирования модулей (см. [п. 3.1.2](#)) помещать устройство EthernetPort_ ниже модулей CPU, в противном случае сервер IEC 60870-5-104 работать не будет.

Для работы порта в режиме сервера IEC 60870-5-104 следует (согласно [Табл. 1.10](#)) в параметрах устройства EthernetPort_ настроить поле `mode` равным 3.

Поле `max_clients` в параметрах устройства EthernetPort_ не используется. Число подключаемых клиентов IEC 60870-5-104 настраивается в файле конфигурации (см. [п. 3.7.3.4](#)).

В поле `config` в параметрах устройства EthernetPort_ следует ввести полное имя файла конфигурации (см. [п. 3.7.3.4](#)).

3.7.3.2 Устройства данных *iec104_**

Выходные устройства *iec104_** (см. [Табл. 1.15](#) и [Табл. 1.16](#)) предназначены для реализации пространства *iec104*-данных.

В начале цикла выполнения программы каналы входных устройств содержат в себе обновившиеся в ходе работы сервера IEC 60870-5-104 данные или команды. Поэтому все устройства данных *iec104_** являются выходными независимо от фактического направления передачи данных. В начале каждого цикла следует копировать значения связанных с устройствами *iec104_** переменных в отдельные, не связанные с каналами рабочие переменные, и далее работать уже с ними.

Все устройства *iec104_** позволяют работать как с IEC104-типами с меткой времени, так и без неё. Метку времени *iec104*-данных можно использовать для задания системного времени CPU (см. [пп. 4.14](#))

Структуры переменных *iec104_**-устройств приведены на [Рис. 3.37](#)...[Рис. 3.66](#).

Параметры всех устройств идентичны (см. [Рис. 3.67](#)).

Параметр `port_id` каждого устройства должен совпадать с параметром `port_id` устройства `EthernetPort_`, к экземпляру сервера IEC 60870-5-104 которого относится данный диапазон данных.

В параметре `ioa` передаётся смещение адресного пространства (отдельное для каждого устройства), которая учитывается при обмене данными с сервером IEC 60870-5-104.



ВНИМАНИЕ

*Адресные пространства устройств *iec104_** не должны пересекаться, значения данных в каналах с совпадающими адресами в ходе работы не определены.*

Name	Data Type	String Size	Comment
IEC104_SP_Data			
IEC104_SP_data	IEC104_SP_data		
value	BOOL		Значение
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения

Рис. 3.37 – Описание структуры IEC104_SP_Data

Name	Data Type	String Size	Comment
IEC104_DP_Data			
IEC104_DP_data	IEC104_DP_data		
value	SINT		Значение
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения

Рис. 3.38 – Описание структуры IEC104_DP_Data

IEC104_ST_data	IEC104_ST_data		
value	SINT		Значение
OV	BOOL		Флаг переполнения
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения

Рис. 3.39 – Описание структуры IEC104_ST_Data

IEC104_BO_data	IEC104_BO_data		
value	DWORD		Значение
OV	BOOL		Флаг переполнения
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения

Рис. 3.40 – Описание структуры IEC104_BO_Data

Name	Data Type	String Size	Comment
IEC104_MEab_Data			
IEC104_MEab_data	IEC104_MEab_data		
value	INT		Значение
OV	BOOL		Флаг переполнения
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения

Рис. 3.41 – Описание структуры IEC104_MEab_Data

Name	Data Type	String Size	Comment
IEC104_MEc_Data			
IEC104_MEc_data	IEC104_MEc_data		
value	REAL		Значение
OV	BOOL		Флаг переполнения
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения

Рис. 3.42 – Описание структуры IEC104_MEc_Data

Name	Data Type	String Size	Comment
IEC104_IT_Data			
IEC104_IT_data	IEC104_IT_data		
value	DINT		Значение
SQ	BYTE		Порядковый номер
CY	BOOL		Флаг переполнения
CA	BOOL		Флаг инициализации
IV	BOOL		Флаг действительности значения

Рис. 3.43 – Описание структуры IEC104_IT_Data

IEC104_MEd_data	IEC104_MEd_data		
value	INT		Значение

Рис. 3.44 – Описание структуры IEC104_MEd_Data

IEC104_SP_TS_data	IEC104_SP_TS_data		
value	BOOL		Значение
BL	BOOL		Флаг блокировки
SB	BOOL		Флаг замещения
NT	BOOL		Флаг актуальности значения
IV	BOOL		Флаг действительности значения
timestamp	IEC104Time		Метка времени
timestamp.msec	UINT		Миллисекунда
timestamp.minute	USINT		Минута
timestamp.hour	USINT		Час
timestamp.summer_time	BOOL		Летнее/зимнее время
timestamp.monthday	USINT		День месяца
timestamp.month	USINT		Месяц
timestamp.year	USINT		Год

Рис. 3.45 – Описание структуры IEC104_SP_TS_Data

IEC104_DP_TS_data		IEC104_DP_TS_data		
<input type="checkbox"/>	value	SINT	▼	Значение
<input type="checkbox"/>	BL	BOOL	▼	Флаг блокировки
<input type="checkbox"/>	SB	BOOL	▼	Флаг замещения
<input type="checkbox"/>	NT	BOOL	▼	Флаг актуальности значения
<input type="checkbox"/>	IV	BOOL	▼	Флаг действительности значения
<input checked="" type="checkbox"/>	timestamp	IEC104Time	▼	Метка времени
<input type="checkbox"/>	timestamp.msec	UINT	▼	Миллисекунда
<input type="checkbox"/>	timestamp.minute	USINT	▼	Минута
<input type="checkbox"/>	timestamp.hour	USINT	▼	Час
<input type="checkbox"/>	timestamp.summer_time	BOOL	▼	Летнее/зимнее время
<input type="checkbox"/>	timestamp.monthday	USINT	▼	День месяца
<input type="checkbox"/>	timestamp.month	USINT	▼	Месяц
<input type="checkbox"/>	timestamp.year	USINT	▼	Год

Рис. 3.46 – Описание структуры IEC104_DP_TS_Data

IEC104_ST_TS_data		IEC104_ST_TS_data		
<input type="checkbox"/>	value	SINT	▼	Значение
<input type="checkbox"/>	OV	BOOL	▼	Флаг переполнения
<input type="checkbox"/>	BL	BOOL	▼	Флаг блокировки
<input type="checkbox"/>	SB	BOOL	▼	Флаг замещения
<input type="checkbox"/>	NT	BOOL	▼	Флаг актуальности значения
<input type="checkbox"/>	IV	BOOL	▼	Флаг действительности значения
<input checked="" type="checkbox"/>	timestamp	IEC104Time	▼	Метка времени
<input type="checkbox"/>	timestamp.msec	UINT	▼	Миллисекунда
<input type="checkbox"/>	timestamp.minute	USINT	▼	Минута
<input type="checkbox"/>	timestamp.hour	USINT	▼	Час
<input type="checkbox"/>	timestamp.summer_time	BOOL	▼	Летнее/зимнее время
<input type="checkbox"/>	timestamp.monthday	USINT	▼	День месяца
<input type="checkbox"/>	timestamp.month	USINT	▼	Месяц
<input type="checkbox"/>	timestamp.year	USINT	▼	Год

Рис. 3.47 – Описание структуры IEC104_ST_TS_Data

IEC104_BO_TS_data		IEC104_BO_TS_data		
	value	DWORD		Значение
	OV	BOOL		Флаг переполнения
	BL	BOOL		Флаг блокировки
	SB	BOOL		Флаг замещения
	NT	BOOL		Флаг актуальности значения
	IV	BOOL		Флаг действительности значения
▶	timestamp	IEC104Time		Метка времени
	timestamp.msec	UINT		Миллисекунда
	timestamp.minute	USINT		Минута
	timestamp.hour	USINT		Час
	timestamp.summer_time	BOOL		Летнее/зимнее время
	timestamp.monthday	USINT		День месяца
	timestamp.month	USINT		Месяц
	timestamp.year	USINT		Год

Рис. 3.48 – Описание структуры IEC104_BO_TS_Data

IEC104_MEab_TS_data		IEC104_MEab_TS_data		
	value	INT		Значение
	OV	BOOL		Флаг переполнения
	BL	BOOL		Флаг блокировки
	SB	BOOL		Флаг замещения
	NT	BOOL		Флаг актуальности значения
	IV	BOOL		Флаг действительности значения
▶	timestamp	IEC104Time		Метка времени
	timestamp.msec	UINT		Миллисекунда
	timestamp.minute	USINT		Минута
	timestamp.hour	USINT		Час
	timestamp.summer_time	BOOL		Летнее/зимнее время
	timestamp.monthday	USINT		День месяца
	timestamp.month	USINT		Месяц
	timestamp.year	USINT		Год

Рис. 3.49 – Описание структуры IEC104_MEab_TS_Data

IEC104_MEc_TS_data		IEC104_MEc_TS_data		
<input type="checkbox"/>	value	REAL	▼	Значение
<input type="checkbox"/>	OV	BOOL	▼	Флаг переполнения
<input type="checkbox"/>	BL	BOOL	▼	Флаг блокировки
<input type="checkbox"/>	SB	BOOL	▼	Флаг замещения
<input type="checkbox"/>	NT	BOOL	▼	Флаг актуальности значения
<input type="checkbox"/>	IV	BOOL	▼	Флаг действительности значения
<input checked="" type="checkbox"/>	timestamp	IEC104Time	▼	Метка времени
<input type="checkbox"/>	timestamp.msec	UINT	▼	Миллисекунда
<input type="checkbox"/>	timestamp.minute	USINT	▼	Минута
<input type="checkbox"/>	timestamp.hour	USINT	▼	Час
<input type="checkbox"/>	timestamp.summer_time	BOOL	▼	Летнее/зимнее время
<input type="checkbox"/>	timestamp.monthday	USINT	▼	День месяца
<input type="checkbox"/>	timestamp.month	USINT	▼	Месяц
<input type="checkbox"/>	timestamp.year	USINT	▼	Год

Рис. 3.50 – Описание структуры IEC104_MEc_TS_Data

IEC104_IT_TS_data		IEC104_IT_TS_data		
<input type="checkbox"/>	value	DINT	▼	Значение
<input type="checkbox"/>	SQ	BYTE	▼	Порядковый номер
<input type="checkbox"/>	CY	BOOL	▼	Флаг переполнения
<input type="checkbox"/>	CA	BOOL	▼	Флаг инициализации
<input type="checkbox"/>	IV	BOOL	▼	Флаг действительности значения
<input checked="" type="checkbox"/>	timestamp	IEC104Time	▼	Метка времени
<input type="checkbox"/>	timestamp.msec	UINT	▼	Миллисекунда
<input type="checkbox"/>	timestamp.minute	USINT	▼	Минута
<input type="checkbox"/>	timestamp.hour	USINT	▼	Час
<input type="checkbox"/>	timestamp.summer_time	BOOL	▼	Летнее/зимнее время
<input type="checkbox"/>	timestamp.monthday	USINT	▼	День месяца
<input type="checkbox"/>	timestamp.month	USINT	▼	Месяц
<input type="checkbox"/>	timestamp.year	USINT	▼	Год

Рис. 3.51 – Описание структуры IEC104_IT_TS_Data

IEC104_EP_TS_data		IEC104_EP_TS_data		
<input type="checkbox"/>	<input type="checkbox"/>	ES	BYTE	
<input type="checkbox"/>	<input type="checkbox"/>	EL	BOOL	
<input type="checkbox"/>	<input type="checkbox"/>	BL	BOOL	Флаг блокировки
<input type="checkbox"/>	<input type="checkbox"/>	SB	BOOL	Флаг замещения
<input type="checkbox"/>	<input type="checkbox"/>	NT	BOOL	Флаг актуальности значения
<input type="checkbox"/>	<input type="checkbox"/>	IV	BOOL	Флаг действительности значения
<input type="checkbox"/>	<input type="checkbox"/>	cp16time	WORD	Пройденное время (мс)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	timestamp	IEC104Time	Метка времени
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.msec	UINT	Миллисекунда
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.minute	USINT	Минута
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.hour	USINT	Час
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.summer_time	BOOL	Летнее/зимнее время
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.monthday	USINT	День месяца
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.month	USINT	Месяц
<input type="checkbox"/>	<input type="checkbox"/>	timestamp.year	USINT	Год

Рис. 3.52 – Описание структуры IEC104_EP_TS_Data

IEC104_SP_command		IEC104_SP_command		
<input type="checkbox"/>	<input type="checkbox"/>	value	BOOL	Значение
<input type="checkbox"/>	<input type="checkbox"/>	QU	BYTE	
<input type="checkbox"/>	<input type="checkbox"/>	SE	BOOL	

Рис. 3.53 – Описание структуры IEC104_SP_Command

IEC104_DP_command		IEC104_DP_command		
<input type="checkbox"/>	<input type="checkbox"/>	value	SINT	Значение
<input type="checkbox"/>	<input type="checkbox"/>	QU	BYTE	
<input type="checkbox"/>	<input type="checkbox"/>	SE	BOOL	

Рис. 3.54 – Описание структуры IEC104_DP_Command

IEC104_RC_command		IEC104_RC_command		
<input type="checkbox"/>	<input type="checkbox"/>	value	SINT	Значение
<input type="checkbox"/>	<input type="checkbox"/>	QU	BYTE	
<input type="checkbox"/>	<input type="checkbox"/>	SE	BOOL	

Рис. 3.55 – Описание структуры IEC104_RC_Command

IEC104_MEab_command		IEC104_MEab_command			
	value	INT	▼		Значение
	SE	BOOL	▼		
	QL	BYTE	▼		

Рис. 3.56 – Описание структуры IEC104_MEab_Command

IEC104_MEc_command		IEC104_MEc_command			
	value	REAL	▼		Значение
	SE	BOOL	▼		
	QL	BYTE	▼		

Рис. 3.57 – Описание структуры IEC104_MEc_Command

IEC104_BO_command		IEC104_BO_command			
	value	DWORD	▼		Значение

Рис. 3.58 – Описание структуры IEC104_BO_Command

IEC104_SP_TS_command		IEC104_SP_TS_command			
	value	BOOL	▼		Значение
	QU	BYTE	▼		
	SE	BOOL	▼		
▶	timestamp	IEC104Time	▼		Метка времени
	timestamp.msec	UINT	▼		Миллисекунда
	timestamp.minute	USINT	▼		Минута
	timestamp.hour	USINT	▼		Час
	timestamp.summer_time	BOOL	▼		Летнее/зимнее время
	timestamp.monthday	USINT	▼		День месяца
	timestamp.month	USINT	▼		Месяц
	timestamp.year	USINT	▼		Год

Рис. 3.59 – Описание структуры IEC104_SP_TS_Command

IEC104_DP_TS_command		IEC104_DP_TS_command			
	value	SINT	▼		Значение
	QU	BYTE	▼		
	SE	BOOL	▼		
+	timestamp	IEC104Time	▼		Метка времени

Рис. 3.60 – Описание структуры IEC104_DP_TS_Command

IEC104_RC_TS_command		IEC104_RC_TS_command			
<input checked="" type="checkbox"/>	value	SINT	▼		Значение
<input checked="" type="checkbox"/>	QU	BYTE	▼		
<input checked="" type="checkbox"/>	SE	BOOL	▼		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> timestamp	IEC104Time	▼	<input checked="" type="checkbox"/>	Метка времени
	<input checked="" type="checkbox"/> timestamp.msec	UINT	▼		Миллисекунда
	<input checked="" type="checkbox"/> timestamp.minute	USINT	▼		Минута
	<input checked="" type="checkbox"/> timestamp.hour	USINT	▼		Час
	<input checked="" type="checkbox"/> timestamp.summer_time	BOOL	▼		Летнее/зимнее время
	<input checked="" type="checkbox"/> timestamp.monthday	USINT	▼		День месяца
	<input checked="" type="checkbox"/> timestamp.month	USINT	▼		Месяц
	<input checked="" type="checkbox"/> timestamp.year	USINT	▼		Год

Рис. 3.61 – Описание структуры IEC104_RC_TS_Command

IEC104_MEab_TS_command		IEC104_MEab_TS_command			
<input checked="" type="checkbox"/>	value	INT	▼		Значение
<input checked="" type="checkbox"/>	SE	BOOL	▼		
<input checked="" type="checkbox"/>	QL	BYTE	▼		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> timestamp	IEC104Time	▼	<input checked="" type="checkbox"/>	Метка времени
	<input checked="" type="checkbox"/> timestamp.msec	UINT	▼		Миллисекунда
	<input checked="" type="checkbox"/> timestamp.minute	USINT	▼		Минута
	<input checked="" type="checkbox"/> timestamp.hour	USINT	▼		Час
	<input checked="" type="checkbox"/> timestamp.summer_time	BOOL	▼		Летнее/зимнее время
	<input checked="" type="checkbox"/> timestamp.monthday	USINT	▼		День месяца
	<input checked="" type="checkbox"/> timestamp.month	USINT	▼		Месяц
	<input checked="" type="checkbox"/> timestamp.year	USINT	▼		Год

Рис. 3.62 – Описание структуры IEC104_MEab_TS_Command

IEC104_MEc_TS_command		IEC104_MEc_TS_command			
<input checked="" type="checkbox"/>	value	REAL	▼		Значение
<input checked="" type="checkbox"/>	SE	BOOL	▼		
<input checked="" type="checkbox"/>	QL	BYTE	▼		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> timestamp	IEC104Time	▼	<input checked="" type="checkbox"/>	Метка времени
	<input checked="" type="checkbox"/> timestamp.msec	UINT	▼		Миллисекунда
	<input checked="" type="checkbox"/> timestamp.minute	USINT	▼		Минута
	<input checked="" type="checkbox"/> timestamp.hour	USINT	▼		Час
	<input checked="" type="checkbox"/> timestamp.summer_time	BOOL	▼		Летнее/зимнее время
	<input checked="" type="checkbox"/> timestamp.monthday	USINT	▼		День месяца
	<input checked="" type="checkbox"/> timestamp.month	USINT	▼		Месяц
	<input checked="" type="checkbox"/> timestamp.year	USINT	▼		Год

Рис. 3.63 – Описание структуры IEC104_MEc_TS_Command

IEC104_BO_TS_command		IEC104_BO_TS_command			
<input type="checkbox"/>	value	DWORD	▼		Значение
<input checked="" type="checkbox"/>	timestamp	IEC104Time	▼		Метка времени
<input type="checkbox"/>	timestamp.msec	UINT	▼		Миллисекунда
<input type="checkbox"/>	timestamp.minute	USINT	▼		Минута
<input type="checkbox"/>	timestamp.hour	USINT	▼		Час
<input type="checkbox"/>	timestamp.summer_time	BOOL	▼		Летнее/зимнее время
<input type="checkbox"/>	timestamp.monthday	USINT	▼		День месяца
<input type="checkbox"/>	timestamp.month	USINT	▼		Месяц
<input type="checkbox"/>	timestamp.year	USINT	▼		Год

Рис. 3.64 – Описание структуры IEC104_BO_TS_Command

IEC104_T230_data		IEC104_T230_data			
<input type="checkbox"/>	value	LREAL	▼		Значение
<input type="checkbox"/>	QD	BYTE	▼		Описание качества

Рис. 3.65 – Описание структуры IEC104_T230_Data

IEC104_T231_data		IEC104_T231_data			
<input type="checkbox"/>	value	LREAL	▼		Значение
<input type="checkbox"/>	QD	BYTE	▼		Описание качества
<input checked="" type="checkbox"/>	timestamp	IEC104Time	▼		Метка времени
<input type="checkbox"/>	timestamp.msec	UINT	▼		Миллисекунда
<input type="checkbox"/>	timestamp.minute	USINT	▼		Минута
<input type="checkbox"/>	timestamp.hour	USINT	▼		Час
<input type="checkbox"/>	timestamp.summer_time	BOOL	▼		Летнее/зимнее время
<input type="checkbox"/>	timestamp.monthday	USINT	▼		День месяца
<input type="checkbox"/>	timestamp.month	USINT	▼		Месяц
<input type="checkbox"/>	timestamp.year	USINT	▼		Год

Рис. 3.66 – Описание структуры IEC104_T231_Data

3.7.3.3 Устройство диагностики драйвера IEC 60870-5-104 (ieci104_server_diag_)

Входное простое устройство `ieci104_server_diag_` предназначено для диагностики драйвера IEC 60870-5-104. Устройство имеет 1 канал типа `IEC104ServerDiag`.

Структура `IEC104ServerDiag` (Рис. 3.68) содержит в себе диагностические данные сервера IEC 60870-5-104:

- в поле `clients` передаётся число подключенных клиентов IEC 60870-5-104;
- поле `server_running` показывает, запущен сервер IEC 60870-5-104 или нет;
- поле `server_error_code` содержит код ошибки работы сервера IEC 60870-5-104 (Табл. 1.17);
- поле `sync_time_count` показывает, сколько запросов синхронизации времени приходило;
- поле `sync_time` показывает метку времени, которая пришла в последней команде синхронизации времени.

IEC104ServerDiag		IEC104ServerDiag		
	<code>clients</code>	INT		Число подключившихся клиентов
	<code>server_running</code>	BOOL		TRUE, если сервер IEC104 запущен
	<code>server_error_code</code>	INT		0 - нет ошибок, иначе код ошибки
	<code>sync_time_count</code>	DINT		Сколько запросов синхронизации времени было
▶	<code>sync_time</code>	IEC104Time		Время, пришедшее по синхронизации
	<code>sync_time.msec</code>	UINT		Миллисекунда
	<code>sync_time.minute</code>	USINT		Минута
	<code>sync_time.hour</code>	USINT		Час
	<code>sync_time.summer_time</code>	BOOL		Летнее/зимнее время
	<code>sync_time.monthday</code>	USINT		День месяца
	<code>sync_time.month</code>	USINT		Месяц
	<code>sync_time.year</code>	USINT		Год

Рис. 3.68 – Описание структуры `IEC104ServerDiag`

Табл. 1.17– Расшифровка кодов ошибки `server_error_code` устройства `ieci104_server_diag_`

Значение <code>server_error_code</code>	Расшифровка
-1	Не совпадают значения <code>port_id</code> в устройствах <code>ieci104_server_diag_</code> и <code>EthernetPort_</code>
0	Сервер работает без ошибок
1	Ошибка инициализации сервера
2	Неизвестный тип IEC 60870-5-104
3	Недопустимая комбинация тип IEC 60870-5-104 – IOA
4	Выполняется подключение клиента
5	Сервер остановлен
6	Не найден файл конфигурации

7	Неверный файл конфигурации
8	Определено более одного сервера

Параметр `port_id` устройства ([Рис. 3.69](#)) должен совпадать с параметром `port_id` устройства `EthernetPort_`, с экземпляра сервера IEC 60870-5-104 которого собирается диагностическая информация.

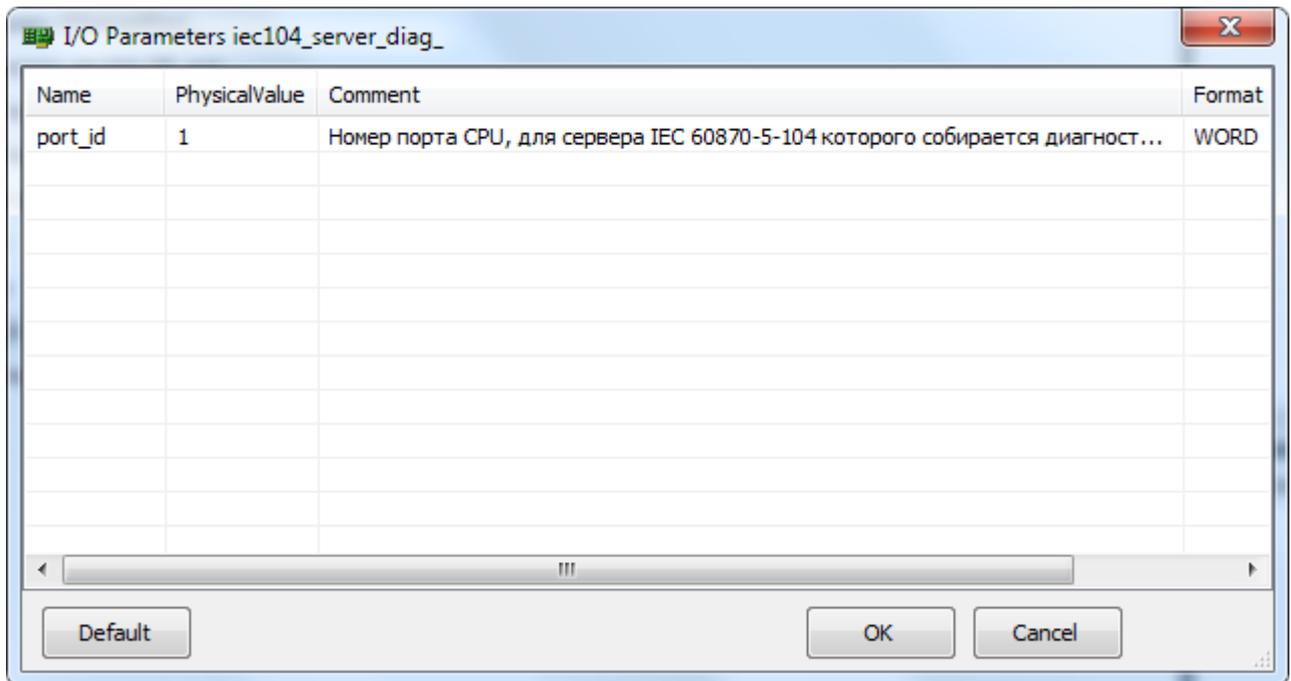


Рис. 3.69 – Описание параметров устройства `iec104_server_diag_`

3.7.3.4 Конфигурация драйвера IEC 60870-5-104

Для работы драйвера IEC 60870-5-104 требуется сформировать и передать во все модули CPU проекта по протоколу FTP ([п. 2.3.1](#)) конфигурационный файл. Конфигурационный файл отвечает за параметры работы и подключения сервера IEC 60870-5-104, а также связывает пространство `iec104-переменных` сервера с добавленными в [пп. 3.7.3.2](#) устройствами.

Отсутствие конфигурационного файла, ошибка в указании его имени в параметрах устройства `EthernetPort_`, либо ошибки в формате конфигурационного файла приводят к невозможности старта сервера IEC 60870-5-104. Все эти ошибки диагностируются в ходе работы ([см. 3.7.3.3](#)).



ВНИМАНИЕ

Начиная с релиза от 2023-08-08 формат файла конфигурации драйвера IEC 60870-5-104 изменился. В целях обратной совместимости поддерживаются оба формата, но в новых разработках рекомендуется использовать новый формат файла.

После успешного запуска драйвер 60870-5-104 автоматически преобразует конфигурационный файл старого образца в файл нового образца.

Все выявленные ошибки в конфигурационном файле автоматически помещаются в FTP в файл `list_error_iec104.ini`.

Общие требования к формату файла конфигурации:

- наличие заголовков вида `[заголовок]`, за которыми следуют пары `ключ=значение,...`
- имена ключей и заголовков **критичны к регистру**;
- порядок следования ключей после заголовка не критичен;
- между знаком равенства и ключом/значением допускается наличие пробела;
- допускается наличие однострочных комментариев. Строка, начинающаяся с символа «#», игнорируется при разборе конфигурационного файла

Старый формат файла конфигурации

Конфигурационный файл старого образца (до релиза от 2023-08-08) должен иметь следующий формат:

1. Секция конфигурации сервера открывается заголовком `[IEC104]`. Параметры `ASDUADR`, `W`, `K`, `T0-T3` должны совпадать с параметрами мастера.

Список ключей:

- 1.1 `TSporadic` - временной период посылки спорадических переменных мастеру, в миллисекундах (устарело, используется для обратной совместимости);
- 1.2 `ASDUADR` - общий адрес ASDU для устройства;
- 1.3 `w` - Последнее подтверждение после приема `W APDU`;
- 1.4 `K` - максимальная разность между переменной состояния передачи и номером последнего подтвержденного `APDU`;

- 1.5 T0 - Тайм-аут при установлении соединения, в секундах (устарело, используется для обратной совместимости);
- 1.6 T1 - Тайм-аут при отправке или тестировании APDU, в секундах;
- 1.7 T2 - Тайм-аут для подтверждения в случае отсутствия сообщения с данными (меньше T1), в секундах;
- 1.8 T3 - Тайм-аут для отправки блоков тестирования в случае долгого простоя, в секундах (0 – не посылать блоки тестирования);
- 1.9 bufsize - Размер буферов для каждого соединения в МБ (от 1 до 8, по умолчанию 1 МБ);
- 1.10 ip - сетевой адрес, с которого принимаются подключения к серверу, может быть несколько строк. Пример:
ip=10.155.26.220
ip=10.155.26.138
По умолчанию, архивирование данных в буфер в оффлайн-режиме отключено. Для управления архивацией следует в конце IP адреса дописать (true) - включения для архивирования, (false) - для отключения архивирования. По умолчанию применяется (false). Пример:
ip=10.155.26.220(true)
ip=10.155.26.138(false)
Добавление строки ip=0.0.0.0 позволяет принимать указанное в параметре Connections (см. ниже) число подключений с прочих IP-адресов.
- 1.11 Connections - Максимально доступное количество подключений с адресов, не перечисленных в ключах ip=xxx.xxx.xxx.xxx, 0 – неограниченное число подключений. По умолчанию принимает значение 0.
- 1.12 AutoFillingTS – Используется для автозаполнения меток времени (1 – включает автозаполнение поля timestamp структур данных для устройств iec104_SP_TS_out_, iec104_DP_TS_out_, iec104_ST_TS_out_, iec104_BO_TS_out_, iec104_MEa_TS_out_, iec104_MEb_TS_out_, iec104_MEc_TS_out_, iec104_IT_TS_out_, iec104_T231_out_). Время меток берётся из процессорного модуля.
- 1.13 TypeTsIrMode=Default|Sporadic|Mirror – режим поведения типов с меткой времени при общем опросе станции. Default – стандартный режим работы протокола, причина передачи 20 (INTERROGATED_COT), 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. При отсутствии строки TypeTsIrMode в конфигурации используется режим Default.

2. Секция конфигурации переменных сервера открывается заголовком [variables]. Переменные одного типа заключаются во вложенный тег, внутри которого перечисляются отдельные переменные данного типа. Имя тега соответствует типу переменной согласно стандарту IEC 60870-5-104.

Пример:

```
<M_SP_NA_1>  
...  
Переменные  
...  
</M_SP_NA_1>
```

Каждая переменная начинается ключевым словом `point`, после которого через пробел следует указать следующие параметры:

- 2.1 `ioa` – уникальный адрес объекта информации, обязательно должен присутствовать;
- 2.2 `sporadic` – является ли объект информации спорадически передаваемым (1 – является, 0 – не является, по умолчанию 0);
- 2.3 `cycle` – передавать ли данные объекта информации с каждым циклом (1 – передавать, 0 – не передавать, по умолчанию 0).

Пример:

```
point ioa=1,sporadic=0,cycle=1  
point ioa=1000,sporadic=1,cycle=0
```

- 2.4 `selectable` – 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 `selectable`, в мс. Допускается значение от 5 до 65535 мс, при выходе за пределы принимается значение 10 мс.

Пример:

```
point ioa=100, selectable=2000  
point ioa=101, selectable=10
```

Пример содержимого конфигурационного файла старого формата приведён в [Листинг 3.1](#).

```
[IEC104]
# секция настроек
ASDUADR=1000
W=8
K=10
T0=0
T1=10
T2=20
T3=30
Connections = 2
AutoFillingTS = 0
buffsize = 4
ip=0.0.0.0
ip=10.155.26.220
ip=10.155.26.138(true)
TypeTsIrMode=Default

[variables]
# начало секции переменных
<M_ME_TF_1>
point ioa=1,sporadic=0,cycle=1
point ioa=2,sporadic=0,cycle=0
</M_ME_TF_1>
<C_SC_NA_1>
point ioa=3
point ioa=4
</C_SC_NA_1>
```

Листинг 3.1 – Пример файла конфигурации сервера IEC 60870-5-104 старого формата

Новый формат файла конфигурации

Конфигурационный файл нового образца (начиная с релиза от 2023-08-08) имеет следующие отличия:

- игнорируются ключи `ip`, теперь «белые адреса» определяются на уровне сетевых интерфейсов модулей CPU (см. [пп. 2.3.2.1](#));
- игнорируется заголовок `[variables]`;
- переменные определённого типа предваряются заголовком с именем типа из [Табл. 1.15](#) и [Табл. 1.16](#);

Пример:

```
[M_ME_NA_1]
```

- переменные описываются в следующем формате:

```
ioa=<address>,sc,<selectable_time>
```

где:

`address` – `ioa`-адрес переменной, обязательный параметр;

`s` – признак спорадически передаваемой переменной (нет символа – не спорадическая переменная);

`c` – передавать ли значением переменной с каждым циклом (нет символа – не циклическая переменная);

`selectable_time` – 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 `selectable`, в мс. Допускается значение от 5 до 65535 мс, при выходе за пределы принимается значение 10 мс.

- в описании переменных обязательно разделять поля запятыми даже если значения полей отсутствуют;
- в описании переменной порядок `s` и `c` не имеет значения.

Пример корректных описаний переменных:

```
ioa=1,s,  
ioa=2,sc,30000  
ioa=3,cs,30000  
ioa=4,s,500  
ioa=5,c,500  
ioa=6,,500  
ioa=7,,
```

Пример некорректных описаний переменных:

```
# ошибка - параметры не разделены запятыми
ioa=666
ioa=667,
# ошибка - selectable_time стоит не на своём месте
ioa=668,30000,
ioa=669,30000,cs
# ошибка - число параметров больше ожидаемого
ioa=671,s,600,c
```

Пример содержимого конфигурационного файла нового формата приведён в [Листинг 3.2](#).

```
[IEC104]
# секция настроек
ASDUADR=1
W=8
K=12
T0=10
T1=15
T2=10
T3=30
TSporadic=100
Connections = 4
AutoFillingTS = 1
buffsize = 4
TypeTsIrMode=Mirror

[M_ME_NA_1]
ioa=1,s,
ioa=2,sc,30000
ioa=3,cs,30000
ioa=4,s,500
ioa=5,c,500
ioa=6,,500
ioa=7,,

[C_SC_TA_1]
ioa=8,,5000

[M_SP_TB_1]
ioa=9,s,
```

Листинг 3.2 – Пример файла конфигурации сервера IEC 60870-5-104 нового формата

3.7.4 Реализация протокола OPC UA (сервер) в модулях CPU МК-503-120

Для работы со встроенным в модуль CPU протоколом OPC UA в режиме сервера создано специальное составное устройство `opcua_server`.

В состав устройства `opcua_server` входят простые устройства `EthernetPort_`, `opcua_server_ctrl_` и `opcua_server_diag_`.

3.7.4.1 Ethernet-порт (EthernetPort_)

Входное простое устройство порт Ethernet (EthernetPort_) предназначено для привязки OPC UA сервера к аппаратному порту модуля CPU. Его назначение и параметры полностью аналогичны описанным в [п. 3.7.1](#) для Modbus-устройств с поддержкой режима «ведущий».

Для привязки к аппаратному порту модуля CPU следует в параметрах устройства EthernetPort_ настроить поле port_id согласно [Табл. 1.7](#).

Так как поле port_id ссылается на аппаратный порт в составе модуля CPU, следует в ходе ранжирования модулей ([п. 3.1.2](#)) помещать OPC UA-устройство ниже модулей CPU, в противном случае OPC UA-устройство работать не будет.

Для работы порта в режиме сервера OPC UA следует (согласно [Табл. 1.10](#)) в параметрах устройства EthernetPort_ настроить поле mode равным 4.

Поле port в параметрах устройства EthernetPort_ следует установить согласно настройкам клиента OPC UA. Стандартным значением по стандарту является 4840 (по умолчанию, в устройстве установлено 502).

Поле max_clients в параметрах устройства EthernetPort_ следует установить равным предполагаемому числу подключаемых клиентов OPC-UA (не более 255, значение 0 трактуется как 1).

Поле slaveID в параметрах устройства EthernetPort_ для устройства OPC UA значения не имеет.

В поле config в параметрах устройства EthernetPort_ в режиме привязки к серверу OPC UA следует ввести имя файла (с расширением) конфигурации (см. ниже), загруженного в модули CPU проекта.

3.7.4.2 Устройство управления работой драйвера OPC UA (opcua_server_ctrl_)

Выходное простое устройство `opcua_server_ctrl_` предназначено для управления работой драйвера OPC UA. Устройство имеет 1 канал типа `Control`, в поле `Command` который следует записывать команду управления (Рис. 3.71). Перечень допустимых значений приведён в Табл. 1.18.

Через параметры устройства можно настроить логин и пароль для аутентификации (Рис. 3.70).

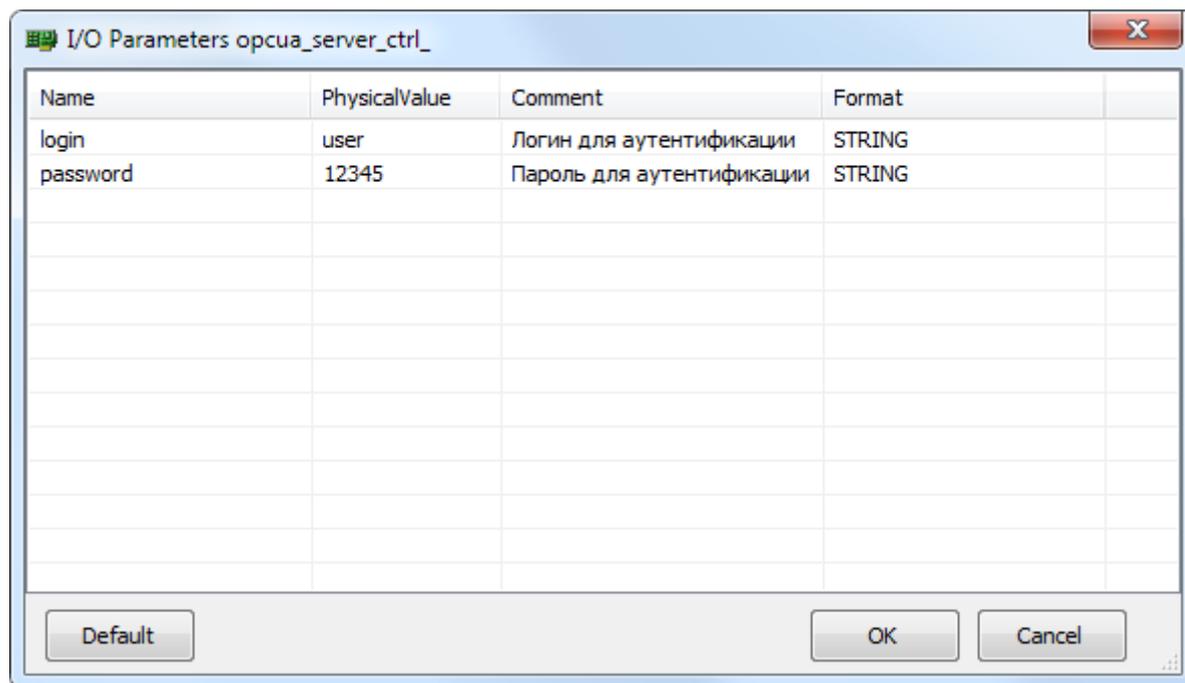


Рис. 3.70 – Параметры устройства `opcua_server_ctrl_`

Name	Data Type	String Size	Comment
OPCUAControl	OPCUAControl		
Command	BYTE		Команда разрешения работы OPC-UA сервера (0 - остановить работу OPC UA, 1 - запустить OPC UA)

Рис. 3.71 – Описание структуры OPCUAControl

Табл. 1.18– Расшифровка значений выхода устройства `opcua_server_ctrl_`

Значение выхода <code>opcua_server_ctrl_</code>	Расшифровка команды
0	Остановить драйвер OPC UA
1	Запустить драйвер OPC UA

3.7.4.3 Устройство диагностики драйвера OPC UA (opcua_server_diag_)

Входное простое устройство `opcua_server_diag_` предназначено для диагностики драйвера OPC UA. Устройство имеет 1 канал типа `OPCUADiag`.

Структура `OPCUADiag` (Рис. 3.72) содержит в себе диагностические данные сервера OPC UA:

- в поле `clients` передаётся число подключенных клиентов к серверу OPC UA;
- поле `status` показывает, запущен сервер OPC UA или нет;
- поле `error_code` содержит код ошибки работы сервера OPC UA. Перечень допустимых значений приведён в Табл. 1.19;
- в поле `workCycleLastTime` передаётся время обработки переменных OPC-UA, в мс;
- в поле `fullCycleLastTime` передаётся полное время цикла обработки переменных OPC-UA, в мс.

Name	Data Type	String Size	Comment
OPCUADiag	OPCUADiag		
OPCUADiag	OPCUADiag		
status	BOOL		TRUE, если сервер OPC UA запущен
error_code	INT		0 - нет ошибок, иначе код ошибки
clients	INT		Число подключившихся клиентов
workCycleLastTime	ULINT		Время цикла обработки переменных, мс
fullCycleLastTime	ULINT		Полное время цикла обработки переменных, мс

Рис. 3.72 – Описание структуры `OPCUADiag`

Табл. 1.19– Расшифровка кодов ошибки `error_code` устройства `opcua_server_diag_`

Значение <code>server_error_code</code>	Расшифровка
-1	Ошибка при создании сервера (внутренняя ошибка)
0	Сервер работает без ошибок
1	Сервер не инициализирован
2	Сервер считывает файл конфигурации
3	Сервер инициализирует переменные
4	Инициализация закончена
5	Сервер завершает работу
6	Сервер не запущен, неверная конфигурация сервера
7	Сервер приостановлен
8	Попытка запустить сервер на занятом порту
9	Ожидание файлов шифрования (при вкл. шифровании)

Также для работы сервера OPC UA требуется сформировать и передать во все модули CPU проекта по протоколу FTP ([п. 2.3.1](#)) конфигурационный файл. Отсутствие конфигурационного файла, ошибка в указании его имени в параметрах устройства EthernetPort_, либо ошибки в формате конфигурационного файла приводят к невозможности старта сервера OPC UA.

Конфигурационный файл должен иметь следующий формат:

1. Секция конфигурации сервера открывается заголовком [OpcUaServer]. Далее следуют пары `ключ=значение`. Порядок следования ключей не критичен, между знаком равенства и ключом/значением допускается наличие пробела. Имена ключей и заголовков критичны к регистру. Перечисленные ключи не являются обязательными.

Список ключей:

- 1.1 `SupportArray` – может иметь значение `true`, все остальные значения трактуется как `false`. Если ключ имеет значение `false`, то элементы переменной-массива создаются как скаляры. Если ключ имеет значение `true`, то переменные-массивы создаются в пространстве OPC UA-сервера как массивы. Не все OPC UA-клиенты поддерживают работу с массивами.

Пример: `SupportArray=true;`

- 1.2 `ApplicationName` – наименование сервера.

Пример: `ApplicationName=opc_ua_server_name`

- 1.3 `ApplicationUri` – уникальное название для адресного пространства сервера.

Пример: `ApplicationUri=urn:localhost:opc_ua_server::nft_space`

- 1.4 `UseAuthentication` – может иметь значение `true`, а все остальные значения трактуется как `false`. Если `true`, то для работы с сервером нужно будет авторизоваться.

Пример: `UseAuthentication=true;`

- 1.5 `UseEncrypt` – может иметь значение `true`, а все остальные значения трактуется как `false`. Если `true`, то при работе с сервером можно использовать шифрование. Также необходимо настроить сертификат с помощью плагина MK500 IODevice, как описано в [разделе 2.3.2.6](#).

Пример: `UseEncrypt=false;`

- 1.6 `UseNumericNodeId` – может иметь значение `true`, а все остальные значения трактуется как `false`. Если `true`, то `NodeId` имеет тип `Numeric`, иначе `NodeId` имеет тип `String`

Пример: `UseNumericNodeId=true;`

- 1.7 `UseNonDimArray` – может иметь значение `true`, а все остальные значения трактуется как `false`. Флаг работает только при `SupportArray=true`. Если `true`, то одномерные массивы имеют другое значение `valueRank`, что позволяет некоторым клиентам (`Alpha.Platform`, `masterOPC`) читать элементы массива.

Пример: `UseNonDimArray=false;`

2. Секция конфигурации переменных открывается заголовком [variables]. Далее следует перечисление имён переменных.

Следует указывать имена только глобальных переменных, локальные переменные будут игнорироваться.

3. Допускается наличие однострочных комментариев. Строка, начинающаяся с символа «#», игнорируется при разборе конфигурационного файла.

Пример: # Параметры сервера

Пример содержимого конфигурационного файла приведён в [Листинг 3.3](#)

```
[OpcUaServer]
# секция настроек
ApplicationUri=urn:localhost:OpcUaServer::TestAppl
ApplicationName=OpcUaServer
# шифрование
UseAuthentication=false
UseEncrypt=true
# работа с массивами
SupportArray=true
UseNonDimArray=true

[variables]
# секция переменных
AI_number
AI_data1
```

Листинг 3.3 – Пример файла конфигурации сервера OPC UA

3.7.4.4 Особенности работы OPC UA

Для улучшения производительности OPC UA рекомендуется выбирать в клиенте тип опроса через чтение (Read в Multi-Protocol MasterOPC или Poll Update Mode в KerServerEx).

При переходе модуля CPU в режим Secondary, значения пользовательских переменных сервера OPC UA обнуляются. Также модуль CPU в режиме Secondary не принимает подключения по OPC UA.

При возврате модуля CPU в режим Primary значения пользовательских переменных сервера OPC UA обновляются, и сервер OPC UA снова начинает принимать подключения по OPC UA и отдавать данные.



ВНИМАНИЕ

Не рекомендуется использовать переменные, отдаваемые по OPC-UA, в промежуточных вычислениях. Желательно использовать эти переменные в программе пользователя один раз.

Ключи шифрования и конфигурационный файл синхронизируются от модуля CPU в режиме Primary в модуль CPU в режиме Secondary автоматически, при изменении настроек либо при подключении Secondary-контроллера.

3.7.5 Реализация протокола Powerlink (ведущий) в модулях CPU МК-502-142 и МК-503-120

Для работы со встроенным в модуль CPU протоколом Powerlink в режиме ведущий предназначено специальное составное устройство PowerlinkMN.

ВНИМАНИЕ

В настоящее время полноценная поддержка протокола Powerlink реализована только на модулях CPU МК-503-120 с использованием коммуникационного модуля МК-546-010.

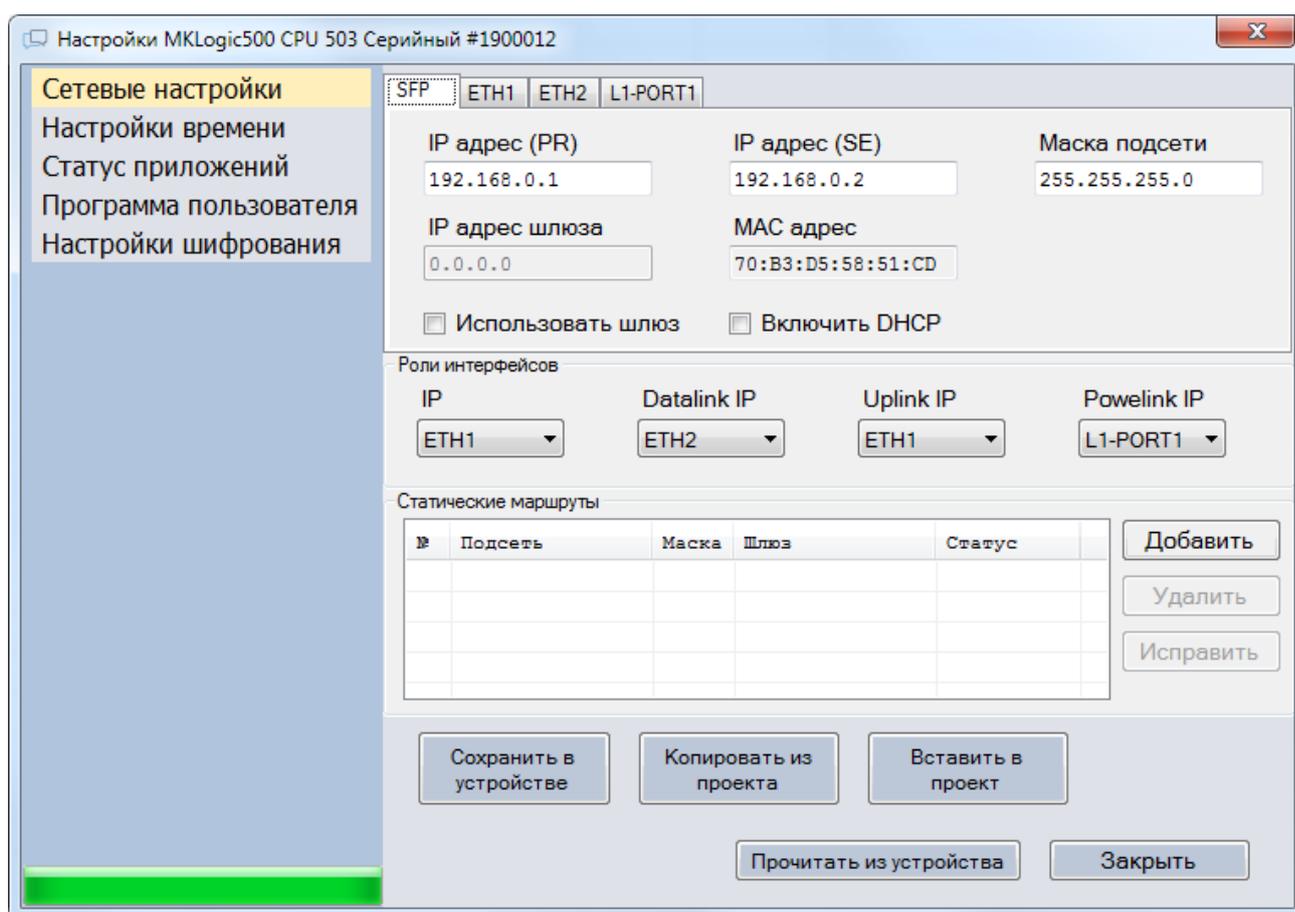


Рис. 3.73 – Выбор роли Powerlink IP

Перед началом работы с устройством PowerlinkMN следует убедиться, что в сетевых настройках модуля CPU выбран сетевой интерфейс, к которому будет привязан драйвер Powerlink. Для этого следует подключиться к модулю CPU с помощью плагина MK500 IO Device (п. 2.3.1) и на вкладке «Сетевые настройки» убедиться, что роль Powerlink IP назначена на сетевой интерфейс коммуникационного модуля МК-546-010, например L1-PORT1. (см. Рис. 3.73).

Сами сетевые настройки модуля МК-546-010 (IP-адреса и пр.) особого значения не имеют, так как модуль МК-546-010 не предназначен для использования в сетях Ethernet.

Составное устройство PowerlinkMN включает в себя простое устройство EthernetPort_, с помощью которого реализуется привязка к аппаратному порту и его диагностика ([Рис. 3.74](#)).

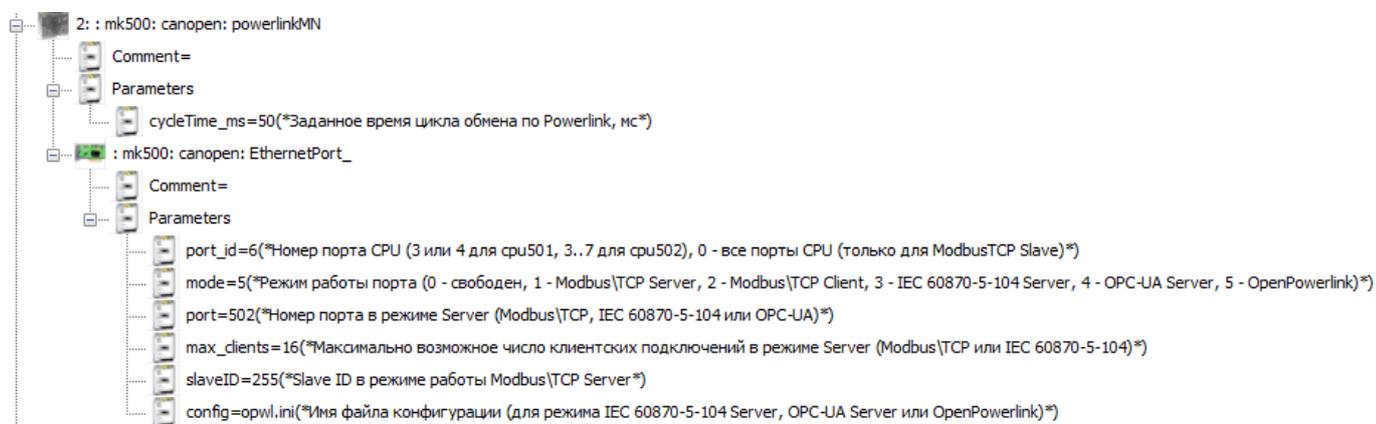


Рис. 3.74 – Составное устройство PowerlinkMN

Входное устройство порт Ethernet (EthernetPort_), предназначенное для привязки Powerlink к аппаратному порту модуля CPU. Его назначение и параметры полностью аналогичны описанным в [п. 3.7.1.1](#) для Modbus-устройств с поддержкой режима ведущий.

Для привязки к аппаратному порту модуля CPU следует в параметрах устройства EthernetPort_ настроить поле port_id согласно [Табл. 1.7](#).

Так как поле port_id ссылается на аппаратный порт в составе модуля CPU, следует в ходе ранжирования модулей ([п. 3.1.2](#)) помещать Powerlink-устройство ниже модулей CPU, в противном случае Powerlink-устройство работать не будет.

Для работы порта в режиме Powerlink следует (согласно [Табл. 1.10](#)) в параметрах устройства EthernetPort_ настроить поле mode равным 5.

Поля port, max_clients и slaveID в параметрах устройства EthernetPort_ для устройства Powerlink значения не имеет.

В поле config в параметрах устройства EthernetPort_ в режиме привязки к серверу Powerlink следует ввести имя файла (с расширением) конфигурации, загруженного в модули CPU проекта.

Для работы Powerlink требуется сформировать и передать во все модули CPU проекта по протоколу FTP ([п. 2.3.1](#)) конфигурационный файл. Отсутствие конфигурационного файла, ошибка в указании его имени в параметрах устройства EthernetPort_, либо ошибки в формате конфигурационного файла приводят к невозможности старта обмена по сети Powerlink.

Формат конфигурационного файла приведен в [Табл. 1.20](#).

Табл. 1.20– COD

Индекс (hex)	Субиндекс (hex)	Назначение	Мин. значение	Макс. значение	Значение по умолчанию
1006	00	Время цикла Powerlink, в мкс	10000	4294967295	50000
1C00	03	Число ошибок CRC в пакетах MN, приводящих к ошибке «CRC Error» (1 ошибка +8 к счётчику, 1 цикл без ошибки -1 к счётчику)	8	1000	15
1C02	03	Число выходов за пределы времени цикла MN, приводящих к ошибке «Cycle time exceeded» (1 ошибка +8 к счётчику, 1 цикл без ошибки -1 к счётчику)	8	1000	15
1C09	01..28	Число потерь CN-ом пакетов Pres, приводящих к ошибке «Loss of PollResponse» (1 ошибка +8 к счётчику, 1 цикл без ошибки -1 к счётчику) Пока не реализовано	8	1000	15
1C0B	03	Число потерь CN-ом пакетов SoC, приводящих к ошибке «Loss of Soc» (1 ошибка +8 к счётчику, 1 цикл без ошибки -1 к счётчику)	8	1000	15
1C0D	03	Число потерей CN-ом пакетов Preq, приводящих к ошибке «Loss of Preq» (1 ошибка +8 к счётчику, 1 цикл без ошибки -1 к счётчику)	8	1000	15

Индекс (hex)	Субиндекс (hex)	Назначение	Мин. значение	Макс. значение	Значение по умолчанию
1C17	01..28	Число потерь MN-ом пакетов StatusResponse для соответствующего CN, приводящих к ошибке «Lost StatusResponse» (1 ошибка +8 к счётчику, 1 цикл без ошибки -1 к счётчику) Пока не реализовано	8	1000	15
1C14	00	Гистерезис для 0x1C0B_03, в нс	1000	4294967295	100000
1F89	01	Время, в течение которого MN остаётся в NMT_MS_NOT_ACTIVE и ждёт кадры POWERLINK до того, как перейдёт в NMT_MS_PRE_OPERATION AL_1, в мкс	250	1000000	100000
1F89	02	Таймаут, в течение которого MN ждёт в состоянии NMT_MS_PRE_OPERATION AL_1 прохождения IdentRequest/IdentResponse всеми обязательными CN, прежде чем перейдёт в ошибку, в мкс 0 – нет таймаута	0	5000000	500000
1F89	03	Время, в течение которого MN ждёт в состоянии NMT_MS_PRE_OPERATION AL_1, в мкс 0 – MN выходит из NMT_MS_PRE_OPERATION AL_1 как только все обязательные CN будут идентифицированы	0	5000000	500000

Индекс (hex)	Субиндекс (hex)	Назначение	Мин. значение	Макс. значение	Значение по умолчанию
1F89	04	Сколько мкс MN ждёт в MS_PRE_OPERATIONAL_2 пока все обязательные CN не перейдут в CS_READY_TO_OPERATE. Если все CN опциональные - сколько мкс MN ждёт после отправки в CN команды NMTEnableReadyToOperate до перехода CN в BOOT_STEP2	0	4294967295	500000
1F89	05	Сколько мкс MN ждёт в MS_READY_TO_OPERATE пока все обязательные CN не перейдут в CS_OPERATIONAL. 0 - MN будет ждать неопределённо долго	0	5000000	500000
1F89	0A	Приоритет RMN, согласно которого выполняется захват шины при отказе AMN (1 - высший приоритет, 10 - низший)	1	10	1 для 241, 2 для 242
1F89	0B	Коэффициент для голосования RMN	0	4294967295	10
1F89	0C	Коэффициент для голосования RMN	0	4294967295	10

- Секция конфигурации сервера открывается заголовком [Powerlink]. Далее следуют пары `ключ=значение`, где ключ - адрес COD (Can Object Dictionary - словарь объектов CAN) в формате `<index>_<subindex>`, значение - текстовое представление значения элемента словаря. `<index>` и `<subindex>` ожидаются в шестнадцатеричном формате (как в документации). Значение можно передавать в десятичном или в шестнадцатеричном (с префиксом 0x) представлении.

В случае ошибки (отсутствует элемент словаря, нет прав на запись, значение вне допустимых пределов) параметр игнорируется.

Все значения секции относятся к MN и применяются в PLCPowerlinkMN. Порядок следования ключей не критичен, между знаком равенства и ключом/значением допускается наличие пробела. Имена ключей и заголовков критичны к регистру.

Поддерживаемые устройством PowerlinkMN COD представлены в [Табл. 1.20](#).

2. Секция кратности открывается заголовком [TPDO_Mu1] либо [RPDO_Mu1]. Секция TPDO_Mu1 содержит список модулей, для которых получаемые из MN данные следует передавать в несколько приёмов. Секция RPDO_Mu1 содержит список модулей, для которых передаваемые в MN данные следует передавать в несколько приёмов. Каждая из секций содержит в себе пары вида "адрес=кратность", где адрес - полный индекс модуля ввода-вывода за CN в формате <opwlIndex_canIndex>, кратность - значение кратности передачи данных модуля (допустимые значения 2, 4, 8, 16) в соответствующем направлении. Модули с кратностью 1 описывать не нужно, это значение кратности по умолчанию.

Значения opwlIndex и canIndex приводятся в десятичном формате.

Рекомендованные значения времени цикла Powerlink в зависимости от числа CN в конфигурации приведены в [Табл. 1.21](#).

Табл. 1.21– Рекомендуемые значения времени цикла Powerlink

Число модулей CN	Рекомендуемое время цикла Powerlink, мкс
1..3	>35000
4..10	>40000
11..20	>45000
21..30	>50000
31..40	>60000

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 удалённой стойки. То есть для стойки с 20 модулями остаётся свободным под данные 1469 байт в TPDO и 1469 байт в RPDO.

В [Табл. 1.22](#) приводится информация о числе байт в TPDO и RPDO всех типов модулей ввода-вывода.

Табл. 1.22– Число байт в TPDO и RPDO для всех типов модулей ввода-вывода

Модель	Официальное наименование	Описание	Байт в TPDO	Байт в RPDO
МК-550-024	Модуль питания МК-550-024	Модуль питания напряжения постоянного тока с интерфейсом CAN	0	8
МК-545-010	Коммуникационный модуль МК-545-010	Коммуникационный модуль Powerlink CN с 2 портами	0	8
МК-521-032	Модуль дискретного ввода МК-521-032	Модуль дискретного ввода напряжения постоянного тока	8	32

Модель	Официальное наименование	Описание	Байт в TPDO	Байт в RPDO
		с 32 дискретными входами		
МК-521-032 А	Модуль дискретного ввода МК-521-032 А	Модуль дискретного ввода напряжения постоянного тока с 32 дискретными входами, исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	8	32
МК-523-032 А	Модуль дискретного ввода МК-523-032 А	Модуль дискретного ввода напряжения постоянного тока с 32 входами NAMUR (согласно IEC 60947-5-6); исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	0	16
МК-531-032	Модуль дискретного вывода МК-531-032	Модуль дискретного вывода напряжения постоянного тока с 32 дискретными выходами	16	0
МК-531-032 А	Модуль дискретного вывода МК-531-032 А	Модуль дискретного вывода напряжения постоянного тока с 32 дискретными выходами, исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	16	0
МК-532-032 А	Модуль дискретного вывода МК-532-032 А	Модуль дискретного вывода напряжения постоянного тока с 32 выходами NAMUR (согласно IEC 60947-5-6); исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	8	16
МК-513-016	Модуль аналогового ввода МК-513-016	Модуль аналогового ввода с 16 аналоговыми входами 0...20 (4...20) мА	0	32
МК-516-008 А	Модуль аналогового ввода МК-516-008 А	Модуль аналогового ввода, 8 аналоговых входов 0...20 (4...20) мА, изолированных друг от друга; исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	0	16
МК-514-008	Модули аналогового вывода МК-514-008	Модуль аналогового вывода с 8 аналоговыми выходами 0...20 (4...20) мА, с 20-контактным разъёмом PUSH-IN	16	8
МК-514-008 А	Модули аналогового вывода МК-514-008 А	Модуль аналогового вывода с 8 аналоговыми выходами 0...20 (4...20)	16	8

Модель	Официальное наименование	Описание	Байт в TPDO	Байт в RPDO
		мА, исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN		
МК-576-008 А	Модуль аналоговых входов МК-576-008 А	Модуль аналогового ввода с 8 изолированными аналоговыми входами 0(4)...20 мА с поддержкой HART; исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	0	176
МК-576-016 А	Модуль аналоговых входов МК-576-016 А	Модуль аналогового ввода с 16 изолированными аналоговыми входами 0(4)...20 мА с поддержкой HART; исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	0	352
МК-574-008 А	Модуль аналоговых выходов МК-574-008 А	Модуль аналогового вывода с 8 аналоговыми выходами 0(4)...20 мА с поддержкой HART; исполнение с быстросъёмным разъёмом 40 контактов с пружинными клеммами типа PUSH-IN	16	168
МК-541-002	Коммуникационный модуль МК-541-002	Коммуникационный модуль с 2 интерфейсами RS-485	2048	2048

Как видно, данные модуля МК-541-002 не укладываются в кадр PDO, поэтому вводится понятие кратности: для каждого модуля можно выбрать кратность, с которой передаются его данные в определённом направлении. Например, для модуля МК-541-002 кадрирование 4 на TPDO и 8 на RPDO означает, что данные TPDO будут передаваться в модуль в течение 4 циклов обмена Powerlink, занимая при этом 512 байт, а данные RPDO будут приниматься из модуля в течение 8 циклов обмена Powerlink, занимая при этом 256 байт.

Рекомендуется указывать равные размеры кратности для одного модуля в обоих направлениях.

Пример содержимого конфигурационного файла приведён в [Листинг 3.4](#).

```
[Powerlink]
1006_00=35000
[RPDO_Mul]
1_4=2
2_5=4
[TPDO_Mul]
2_4=2
2_5=4
```

Листинг 3.4 – Пример файла конфигурации Powerlink

3.7.6 Сбор диагностической информации модуля CPU

В силу особенностей работы модулей CPU в режиме Failover использование в программе пользователя диагностических переменных времени исполнения (доступны в секции Global Variables, имена имеют вид `__SYSVA_xxx`) не допускается. Поэтому все их значения реализованы в качестве входных переменных в дополнительном устройстве `system_info`. Также устройство `system_info` возвращает состояние служб модуля CPU.

Устройство `system_info` имеет один входной канал типа `SysInfo`. Расшифровка его полей приводится в [Табл. 1.23](#).

Табл. 1.23– Расшифровка значений полей структуры SysInfo

Поле структуры SysInfo	Описание
<code>Cycle.CycleCounter</code>	Число выполненных циклов программы пользователя
<code>Cycle.CycleDateTime</code>	Время с момента начала работы программы пользователя
<code>Cycle.ScanCounter</code>	Число выполненных чтений входных переменных
<code>Cycle.CycleTime</code>	Заданное время цикла, мс
<code>Cycle.CurrentCycleTime</code>	Текущее время выполнения программы пользователя, мс
<code>Cycle.MaxCycleTime</code>	Максимальное время выполнения программы пользователя, мс
<code>Cycle.CycleOverflows</code>	Число превышений заданного времени цикла
<code>ResInfo.Name</code>	Полное имя ресурса проекта
<code>ResInfo.Mode</code>	Режим выполнения ресурса
<code>ResInfo.CRC32</code>	CRC32 файлов ресурса в CPU
<code>ResInfo.SHA1</code>	SHA1 файлов ресурса в CPU (как массив USINT)
<code>ResInfo.StrSHA1</code>	SHA1 файлов ресурса в CPU (как строка)
<code>ResInfo.ModifyDate</code>	Дата последнего изменения ресурса
<code>ResInfo.ModifyTime</code>	Время последнего изменения ресурса
<code>ResInfo.OnlineChangeCnt</code>	Количество выполненных программистом Online Change
<code>Failover.IsEnable</code>	TRUE если разрешён режим работы Failover
<code>Failover.isPrimary</code>	TRUE если CPU стартовал как первичный, FALSE если CPU стартовал как вторичный
<code>Failover.IsActive</code>	TRUE если CPU является активным
<code>Failover.ErrCode</code>	Код ошибки Failover, согласно документации: 0 – Нет ошибок 1 – Резервный CPU не может считать тактовый импульс с активного CPU 2 – CPU не могут установить соединение через канал передачи данных 3 – CPU – разных типов и несовместимы между собой 4 – CPU поддерживают различающийся функционал
<code>Failover.DataSyncTime</code>	Время синхронизации данных между CPU, мс

Поле структуры SysInfo	Описание
Failover.DataSyncCount	Число полных синхронизаций с момента запуска программы пользователя
Failover.HeartbeatSyncTime	Время между тактовыми импульсами Failover, мс
Failover.ActiveCPURack	Номер стойки с активным CPU (1..8)
Failover.ActiveCPUSlot	Номер слота с активным CPU (3 или 4)
Services.FTPStatus	Код состояния службы FTP: 0 – служба FTP остановлена 1 – служба FTP запущена -1 – служба FTP запускается -2 – служба FTP останавливается

3.7.7 Получение расширенной диагностики для модуля CPU МК-503-120

Модуль CPU МК-503-120 может расширяться коммуникационными модулями МК-544-040 (4 интерфейса Ethernet) и МК-546-010 (1 интерфейс Powerlink). Также модуль CPU МК-503-120 имеет в своём составе слот для сетевого модуля SFP. Для получения программой пользователя диагностической информации обо всех аппаратных расширениях модуля CPU МК-503-120 служат дополнительные модули ввода-вывода `mk544`, `mk546` и `mk503ext`.

3.7.7.1 Модуль расширенной диагностики CPU МК-503-120 mk503ext

Входное простое устройство `mk503ext` предназначено для получения дополнительной диагностической информации о модуле CPU МК-503-120. Устройство имеет 1 канал типа `MK503ExtT` (Рис. 3.75) Для корректной работы устройства `mk503ext` следует настроить его параметры `rack` и `slot` так, чтобы они «указывали» на положение модуля CPU МК-503-120 (Рис. 3.76).

Name	Data Type	String Size	Comment
Mk503ExtT			
MK503ExtT	MK503ExtT		
SFP	SFPDiag		Диагностическая информация модуля SFP
SFP.present	BOOL		Наличие модуля SFP
SFP.dist	SINT		Максимальная дальность работы модуля SFP, км
*			
version	STRING	16	Версия реализации аппаратной составляющей CAN-портов
*			

Рис. 3.75 – Описание структуры MK503ExtT

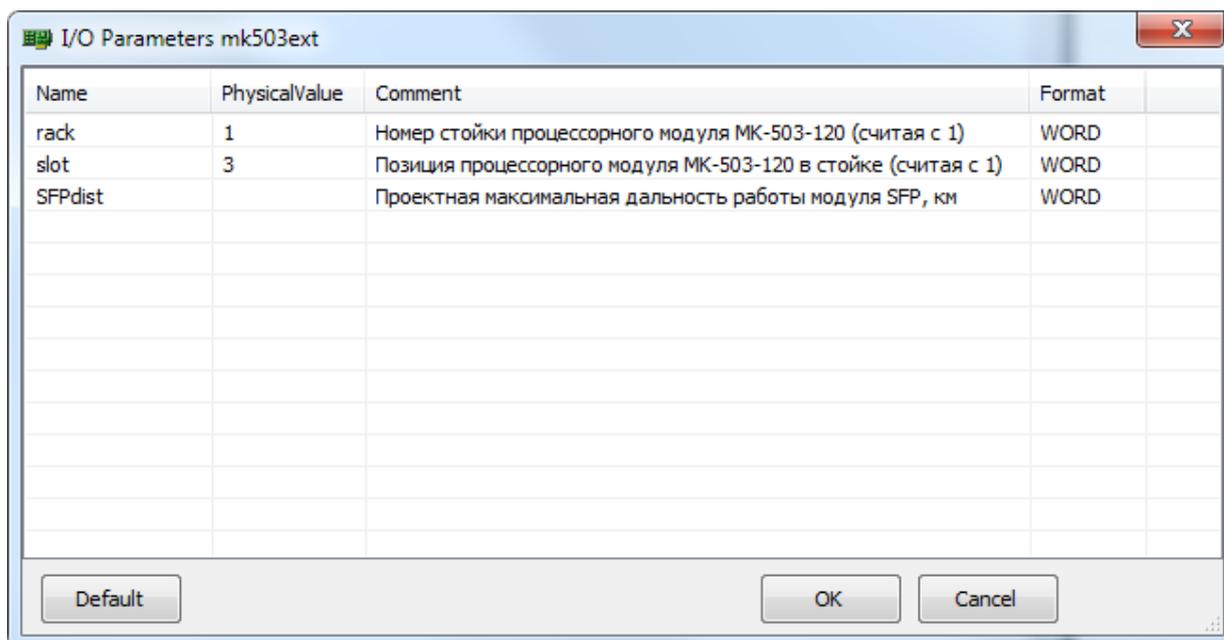


Рис. 3.76 – Параметры устройства mk503Ext

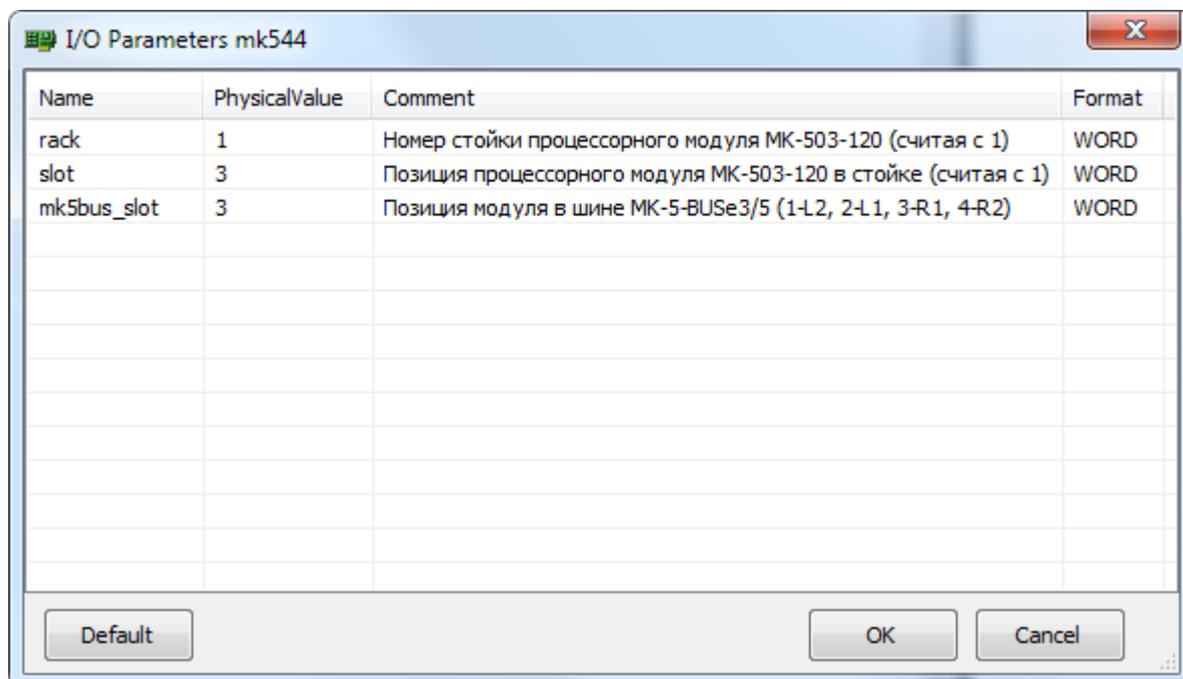
Так как поля `rack` и `slot` ссылаются на модуль CPU, следует в ходе ранжирования модулей (п. 3.1.2) помещать устройство `mk503ext` ниже модулей CPU, в противном случае диагностическая информация собираться не будет.

3.7.7.2 Модуль диагностики коммуникационного модуля МК-544-040 mk544

Входное простое устройство mk544 предназначено для получения дополнительной диагностической информации о коммуникационном модуле МК-544-040. Устройство имеет 1 канал типа MK544Diag ([Рис. 3.77](#)) Для корректной работы устройства mk544 следует настроить его параметры rack, slot и mk5bus_slot так, чтобы они «указывали» на положение модуля CPU МК-503-120, и на положение модуля МК-544-040 в шине МК-5-BUSE3/5 ([Рис. 3.78](#)).

Name	Data Type	String Size	Comment
MK544Diag			
MK544Diag	MK544Diag		
ports	MK544Ether		Диагностика Ethernet-портов
ports[1]	EthernetPortI		
ports[1].present	BOOL		TRUE если порт присутствует в модуле
ports[1].uplink	BOOL		TRUE если порт подключен
*			
ports[2]	EthernetPortI		
ports[3]	EthernetPortI		
ports[4]	EthernetPortI		
*			
version	STRING	16	Версия реализации аппаратной составляющей портов

Рис. 3.77 – Описание структуры MK544Diag



Name	PhysicalValue	Comment	Format
rack	1	Номер стойки процессорного модуля МК-503-120 (считая с 1)	WORD
slot	3	Позиция процессорного модуля МК-503-120 в стойке (считая с 1)	WORD
mk5bus_slot	3	Позиция модуля в шине МК-5-BUSE3/5 (1-L2, 2-L1, 3-R1, 4-R2)	WORD

Рис. 3.78 – Параметры устройства mk544

Так как поля rack и slot ссылаются на модуль CPU, следует в ходе ранжирования модулей ([п. 3.1.2](#)) помещать устройство mk544 ниже модулей CPU, в противном случае диагностическая информация собираться не будет.

3.7.7.3 Модуль диагностики коммуникационного модуля МК-546-010 mk546

Входное простое устройство mk546 предназначено для получения дополнительной диагностической информации о коммуникационном модуле МК-546-010. Устройство имеет 1 канал типа MK546Diag ([Рис. 3.79](#)) Для корректной работы устройства mk546 следует настроить его параметры rack, slot и mk5bus_slot так, чтобы они «указывали» на положение модуля CPU МК-503-120, и на положение модуля МК-546-010 в шине МК-5-BUSE3/5 ([Рис. 3.80](#)).

Name	Data Type	String Size	Comment
MK546Diag	MK546Diag		
MK546Diag	MK546Diag		
ports	PowerlinkPorts		Диагностика Powerlink-портов
ports[1]	PowerlinkPortD		
ports[1].Linkup	BOOL		Наличие соединения на порту (TRUE - есть соединение)
ports[1].FullDuplex	BOOL		Режим работы порта (TRUE - полный дуплекс, FALSE - полудуплекс)
ports[1].Speed	BYTE		Скорость порта (0 - 10 Мбит/с, 1 - 100 Мбит/с, 2 - 1000 Мбит/с)
ports[1].PortBlocking	BOOL		TRUE - порт заблокирован
ports[1].RemoteRXFailure	BOOL		TRUE - отсутствует приём на удалённом порту
ports[1].RXFailure	BOOL		TRUE - отсутствует приём на порту
ports[1].Loopback	BOOL		TRUE - наличие на порту петли
ports[1].MultipleErrors	BOOL		TRUE - множественные ошибки на порту
ports[1].DiagErrorsCounter	BYTE		Счётчик ошибок диагностического канала
*			
ports[2]	PowerlinkPortD		
*			
Leds	PowerlinkLeds		Состояние светодиодов передней панели модуля
Leds.Line	BOOL		TRUE - на модуле горит светодиод Line, сеть Powerlink работает в режиме Line
Leds.Ring	BOOL		TRUE - на модуле горит светодиод Ring, сеть Powerlink работает в режиме кольца
Leds.Eth	BOOL		TRUE - на модуле горит светодиод Eth, сеть работает в режиме Ethernet
Leds.Pwl	BOOL		TRUE - на модуле горит светодиод Pwl, сеть работает в режиме Powerlink
*			
version	STRING	16	Версия реализации аппаратной составляющей портов

Рис. 3.79 – Описание структуры MK546Diag

Поля PortBlocking, RemoteRXFailure, RXFailure, Loopback, MultipleErrors и DiagErrorsCounter в структуре PowerlinkPortDiag предназначены для использования в будущем и на 03.2021 не реализованы; их значения следует игнорировать.

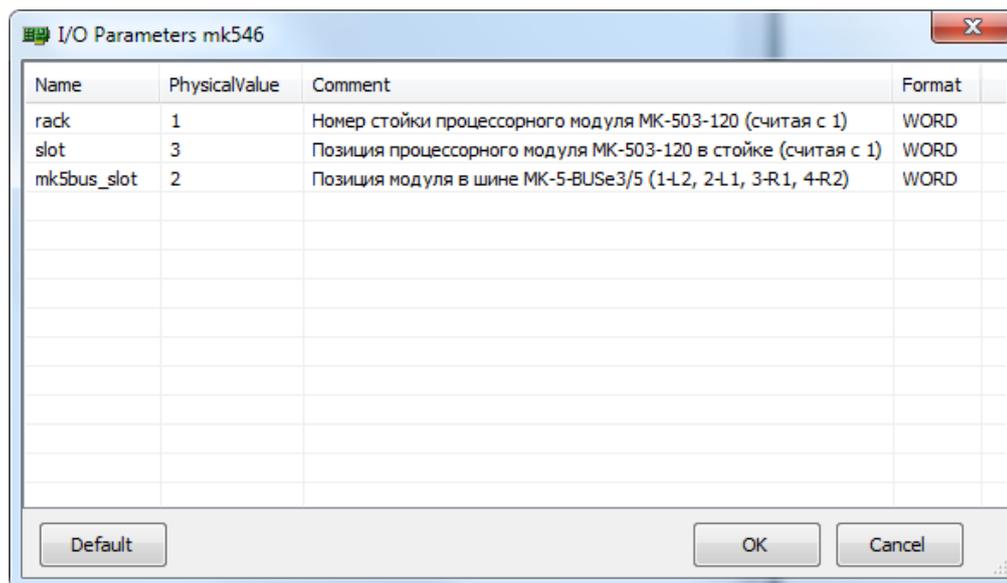


Рис. 3.80 – Параметры устройства mk546

Так как поля `rack` и `slot` ссылаются на модуль CPU, следует в ходе ранжирования модулей ([п. 3.1.2](#)) помещать устройство `mk546` ниже модулей CPU, в противном случае диагностическая информация собираться не будет.

3.7.8 Получение значения текущего времени активного модуля CPU

В силу особенностей работы модулей CPU в режиме Failover использование в программе пользователя функций получения текущего времени запрещено ([п. 5.2](#)). Для получения и использования в программе значений текущего времени следует использовать специальное простое устройство `current_time`, так как его входные значения времени синхронизируются между модулями CPU и совпадают для программ пользователя обоих CPU.

Входное простое устройство `current_time` предназначено для получения текущего времени активного модуля CPU. Устройство имеет 1 канал типа MK544Diag ([Рис. 3.81](#)).

Name	Data Type	String Size	Comment
Now			
NowTime	NowTime		
sec_unix	DINT		Число секунд с 1970/01/01
sec_2000	DINT		Число секунд с 2000/01/01
msec	DINT		Число миллисекунд в текущей секунде
usec	DINT		Число микросекунд в текущей секунде
lastScan_msec	DINT		Время выполнения последнего скана, в миллисекундах
current_date	DATE		Текущая дата в формате DATE
current_time	TIME		Текущее время в формате TIME

Рис. 3.81 – Описание структуры NowTime

Ограничений на размещение устройства `current_time` относительно других устройств нет.

3.7.9 Дополнительная ping-диагностика удалённых серверов

Переключение активного модуля CPU по результатам ping-диагностики реализовано в дополнительном устройстве ping_diag (Рис. 3.82).



Рис. 3.82 – Устройство ping_diag

Параметр configName предназначен для указания имени файла в папке FTP с конфигурацией ping-сервиса.

Устройство имеет 1 входной канал типа PingStatus (Рис. 3.83).

Name	Data Type	String Size	Comment
PingStatus	PingStatus		
statusPR	INT		Число адресов, отвечающих на ping от primary CPU
statusSE	INT		Число адресов, отвечающих на ping от secondary CPU
statusConfig	INT		0 - ping-сервис запущен без ошибок, иначе код ошибки

Рис. 3.83 – Описание структуры PingStatus

Устройство включает ping-сервис, и логику переключения активного CPU по результатам ping-a серверов.

В поле statusPR показывается число доступных адресов Primary CPU, в поле statusSE показывается число доступных адресов Secondary CPU, в поле statusConfig показывается состояние ping-сервиса (Табл. 1.24).

Табл. 1.24– Коды состояний ping-сервиса

Код ошибки	Значение
0	Сервис работает, ошибок нет
1	Сервис не работает, пустое имя файла
2	Сервис не работает, файл не найден
3	Сервис не работает, файл содержит ошибки
20	Сервис не работает, не задана роль uplink в MKLogic 500 CPU 50x configurator

Сервис работает только по интерфейсу, установленному на роль Uplink IP в плагине MK500 IODevice (см. п. 2.3.2.1). Если данная роль не задана, либо есть ошибки при обращении к файлу конфигурации, ping-сервис работать не будет.

Пример файла конфигурации:

```
[Ping]
Timeout=200
Period=500
IP1=192.168.10.1
IP2=192.168.10.2
```

где:

- Поле `Timeout` - задает таймаут ожидания ответа в мс;
- Поле `Period` - задает период опроса адресов в мс;
- Поля `IP#` - задают IP адреса, проверяемые утилитой `ping` на доступность (поддерживаются индексы полей IP с 1 по 8).

Переключение активного модуля CPU выполняется, если на пассивном модуле CPU число отвечающих на `ping` адресов больше, чем на активном модуле CPU. Задержка переключения составляет $(Timeout + Period) * 2$.

При значениях `Timeout` или `Period` менее 10 мс возможны ложные переключения активного модуля CPU, устанавливать такие значения не рекомендуется.

3.7.10 Получение системных диагностических сообщений модуля CPU

Прием диагностических сообщений реализован в дополнительном устройстве `system_diag`.

Устройство `system_diag` имеет 256 входных каналов типа `DiagEventWithTS` (Рис. 3.84).

Name	Data Type	String Size	Comment
DiagEventWithTS	DiagEventWithTS		
DiagEventWithTS	DiagEventWithTS		
timeStamp	NowTime		Метка времени для диагностического сообщения
timeStamp.sec_unix	DINT		Число секунд с 1970/01/01
timeStamp.sec_2000	DINT		Число секунд с 2000/01/01
timeStamp.msec	DINT		Число миллисекунд в текущей секунде
timeStamp.usec	DINT		Число микросекунд в текущей секунде
timeStamp.lastScan_msec	DINT		Время выполнения последнего скана, в миллисекундах
timeStamp.current_date	DATE		Текущая дата в формате DATE
timeStamp.current_time	TIME		Текущее время в формате TIME
*			
diagEvent	DiagEvent		Диагностическое сообщение в указанный момент времени
diagEvent.code	DINT		Код диагностического сообщения
diagEvent.subCode	DINT		Субкод диагностического сообщения
*			
*			

Рис. 3.84 – Описание структуры данных `DiagEventWithTS`

Устройство предназначено для получения диагностических сообщений за время последнего цикла программы пользователя.

Во подструктуре `diagEvent` хранятся значения кода и субкода диагностического сообщения (начиная с самого «старого»). Метки времени изменения входов фиксируется в поле `timeStamp`.

3.8 Работа с модулями аналогового ввода МК-513-016 и МК-513-016А

Согласно [Табл. 1.1](#), модулям аналогового ввода МК-513-016 и МК-513-016А в среде разработки ACP Workbench ISaGRAF 6.50 соответствуют модули изделия ai16 и ai16a соответственно. Функционально модули изделий ai16 и ai16a абсолютно эквивалентны и отличаются только исполнением.

Кроме диагностического канала, модули изделий ai16 и ai16a имеют в своём составе простое устройство ai16_ с 16 входными каналами данных типа WORD.

Значение каждого канала устройства ai16_ 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. 1.25](#).

Табл. 1.25– Расшифровка значений входов устройства ai16_

Значения кода АЦП канала	Величина входного тока канала, мА
0	0,00 (минимальное значение)
2640	4,00
13200	20,00
16383	24,82 (максимальное значение)

Также каждый канал устройства ai16_ имеет свои независимые параметры ([Рис. 3.85](#)).

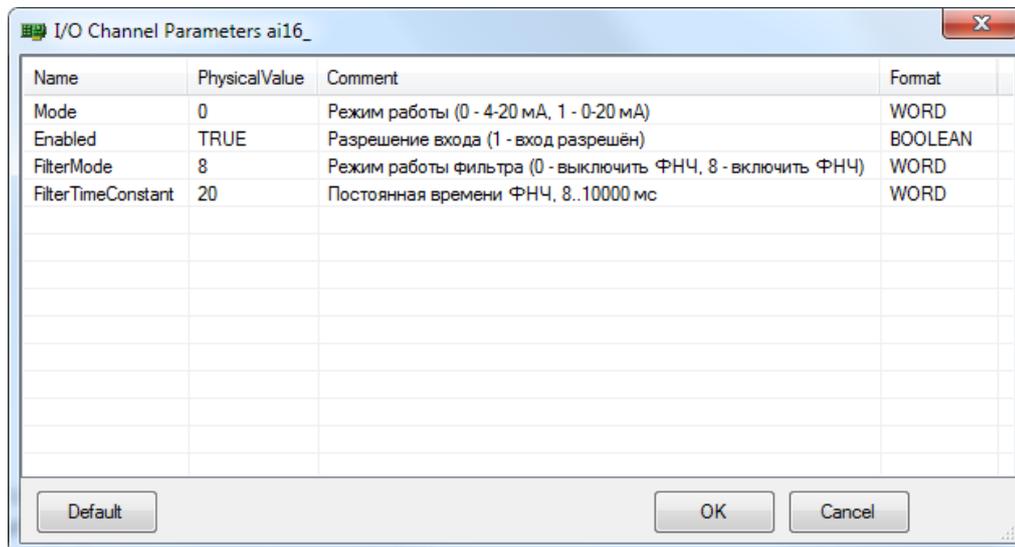


Рис. 3.85 – Описание параметров каналов устройства ai16_

Параметры каналов устройства ai16_:

- 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 АЦП равный 0. По умолчанию все каналы разрешены.

– `FilterMode` – режим работы фильтров канала ([Табл. 1.26](#)). Работа фильтров канала при иных значениях параметра `FilterMode` не определена. По умолчанию фильтрация на всех каналах отключена.

– `FilterLowPassFreq` – постоянная времени фильтра низких частот, от 3 до 10000 мс. Имеет значение только при `FilterMode=8`. По умолчанию постоянная времени на всех каналах равна 20 мс.

Табл. 1.26– Расшифровка значений параметра `FilterMode` канала устройства ai16_

Значение параметра <code>FilterMode</code>	Режим работы фильтров канала устройства ai16_
0	Фильтр низких частот выключен
8	Фильтр низких частот включен

3.9 Работа с модулями аналогового вывода МК-514-008 и МК-514-008А

Согласно [Табл. 1.1](#), модулям аналогового вывода МК-514-008 и МК-514-008А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модули изделия ао8 и ао8а соответственно. Функционально модули изделий ао8 и ао8а абсолютно эквивалентны и отличаются только исполнением.

Кроме диагностического канала, модуль изделия ао8 имеет в своём составе простое устройство ао8_ с 8 выходными каналами данных типа WORD, и простое устройство ао8status_ с 8 входными каналами данных типа BYTE.

Значение каждого канала устройства ао8_ 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. 1.27](#).

Табл. 1.27– Расшифровка значений выходов устройства ао8_

Значения кода ЦАП канала	Величина выходного тока канала, мА
0	0,00 (минимальное значение)
10922	4,00
54613	20,00

Также каждый канал устройства ао8_ имеет свои независимые параметры ([Рис. 3.86](#)).

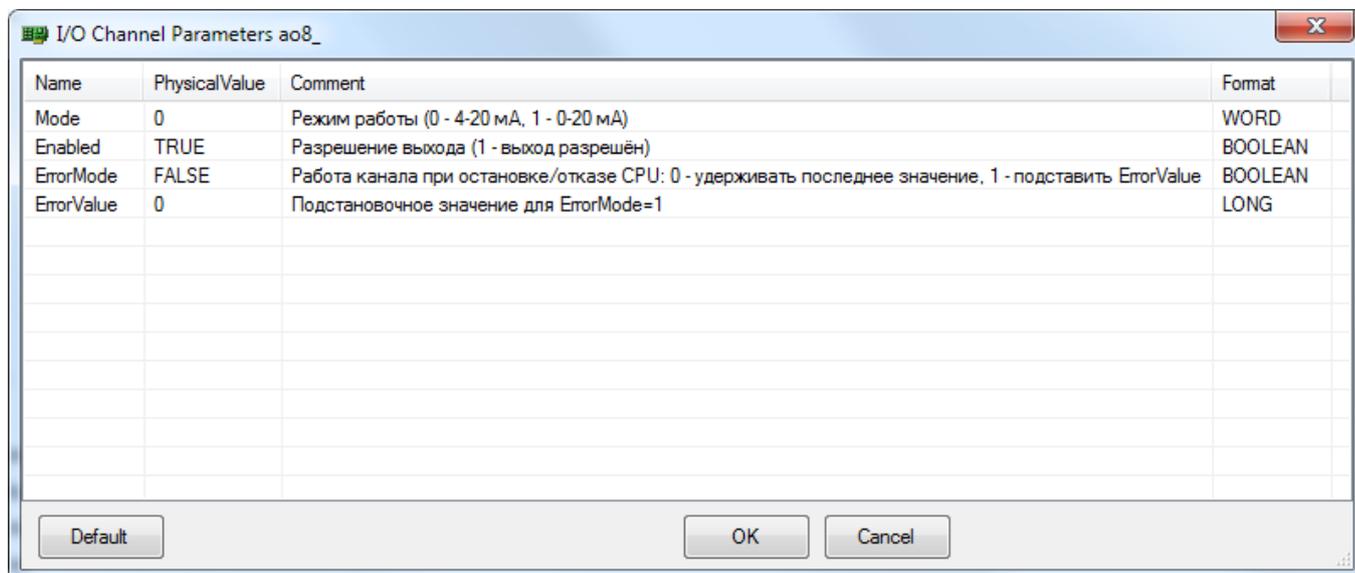


Рис. 3.86 – Описание параметров каналов устройства ао8_

Параметры каналов устройства ао8_ :

– 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.

Значение каналов устройства `ao8status_` соответствует состоянию электрических цепей соответствующих каналов модуля аналогового вывода. Код состояния канала имеет тип `BYTE` и состоит из двух частей.

Младшие 4 бита кода состояния (см. [Табл. 1.28](#)) описывают общее состояние канала, старшие 4 бита кода состояния (см. [Табл. 1.29](#)) содержат в себе код ошибки.

Табл. 1.28– Расшифровка входов устройства `ao8status_`

Значения битов статуса	Состояние канала аналогового выхода <code>ao8_</code>
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

Второй полубайт служит для кода ошибки, если взведен 1 бит статуса:

Табл. 1.29– Расшифровка кодов ошибки каналов устройства `ao8_`

Код ошибки канала	Состояние канала аналогового входа <code>ao8_</code>
0	Нет ошибок
1	Ошибка конфигурации (параметры канала <code>Mode</code> и <code>Enabled</code> 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 <code>watchdog</code>
8	Другой внутренний отказ модуля

3.10 Работа с модулями аналогового ввода МК-516-008 и МК-516-008А

Согласно [Табл. 1.1](#), модулям аналогового ввода МК-516-008 и МК-516-008А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модули изделия ai8 и ai8a соответственно. Функционально модули изделий ai8 и ai8a абсолютно эквивалентны и отличаются только исполнением.

Кроме диагностического канала, модуль изделия ai8 имеет в своём составе простое устройство ai8_ с 8 входными каналами данных типа WORD.

Значение каждого канала устройства ai8_ 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. 1.30](#).

Табл. 1.30– Расшифровка значений входов устройства ai8_

Значения кода АЦП канала	Величина входного тока канала, мА
0	0,00 (минимальное значение)
10486	4,00
52428	20,00
65535	25,00 (максимальное значение)

Также каждый канал устройства ai8_ имеет свои независимые параметры ([Рис. 3.87](#)).

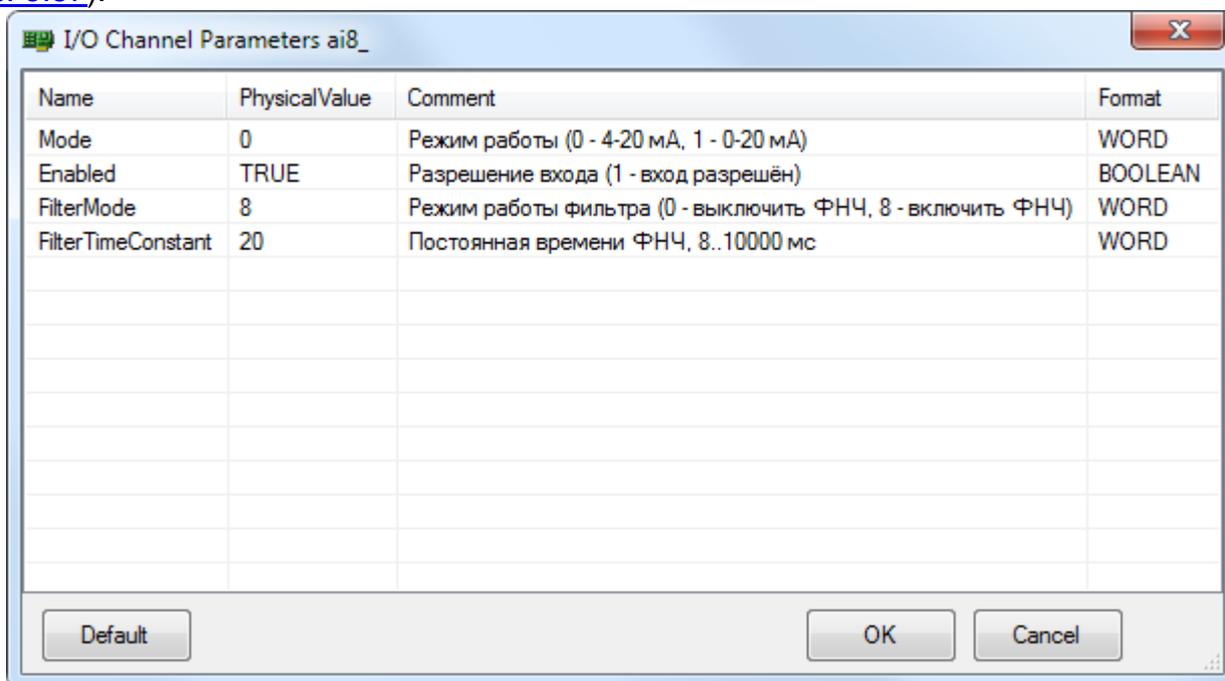


Рис. 3.87 – Описание параметров каналов устройства ai8_

Параметры каналов устройства ai8_:

– 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-516-08, 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 АЦП равный 0. По умолчанию все каналы разрешены.

– `FilterMode` – режим работы фильтров канала ([Табл. 1.31](#)). Работа фильтров канала при иных значениях параметра `FilterMode` не определена. По умолчанию фильтрация на всех каналах отключена.

– `FilterLowPassFreq` – постоянная времени фильтра низких частот, от 3 до 10000 мс. Имеет значение только при `FilterMode=8`. По умолчанию постоянная времени на всех каналах равна 5 мс.

Табл. 1.31– Расшифровка значений параметра `FilterMode` канала устройства `ai8_`

Значение параметра <code>FilterMode</code>	Режим работы фильтров канала устройства <code>ai8_</code>
0	Фильтр низких частот выключен
8	Фильтр низких частот включен

3.11 Работа с модулями дискретного ввода МК-521-032 и МК-521-032 А

Согласно [Табл. 1.1](#), модулю дискретного ввода МК-521-032 в среде разработки ACP Workbench ISaGRAF 6.50 соответствуют модули изделия `di32` и `di32history`; модулю дискретного ввода МК-521-032 А в среде разработки соответствуют модули изделия `di32a` и `di32ahistory`.

Модули дискретного ввода МК-521-032 и МК-521-032 А отличаются только конструктивно и совершенно не отличаются в работе, поэтому далее будет описана только работа с модулями изделия `di32` и `di32history`.

Кроме диагностического канала, модуль изделия `di32` имеет в своём составе простое устройство `di32_` с 32 входными каналами данных типа `DI32Inputs` ([Рис. 3.88](#)), и простое устройство `di32rsttrig_` с 32 выходными каналами данных типа `BOOL`.

Name	Data Type	String Size	Comment
DI32Inputs			
DI32Inputs	DI32Inputs		
input	BOOL		Дискретный вход
trigger	BOOL		Триггер, срабатывающий по срезу входа

Рис. 3.88 – Описание структуры данных DI32Inputs

Входные каналы устройства `di32_` в структуре `DI32Inputs` возвращают текущее значение дискретного входа в поле `input` и флаг срабатывания триггера в поле `trigger`, переходящего в `TRUE` по срезу дискретного входа канала.

Также каждый канал устройства `di32_` имеет свои независимые параметры ([Рис. 3.89](#)).

Параметры каналов устройства `di32_`:

- `Enabled` – разрешение работы канала. `FALSE` – запрещён, `TRUE` – разрешён. Для запрещённого канала поля `input` и `trigger` структуры `DI32Inputs` всегда равны `FALSE`. По умолчанию все каналы разрешены.
- `Invert` – режим инверсии поля `input` структуры `DI32Inputs` канала. `FALSE` – инверсия выключена, `TRUE` – инверсия включена. По умолчанию инверсия всех каналов выключена.
- `BounceTime` – минимально допустимая длина входного сигнала на канале, от 10000 до 100000 мкс, с шагом 10000 мкс. Входной сигнал длительностью меньше значения `BounceTime` игнорируется. Данный параметр используется для защиты входов от дребезга. По умолчанию значение `BounceTime` всех каналов равно 10000 мкс.

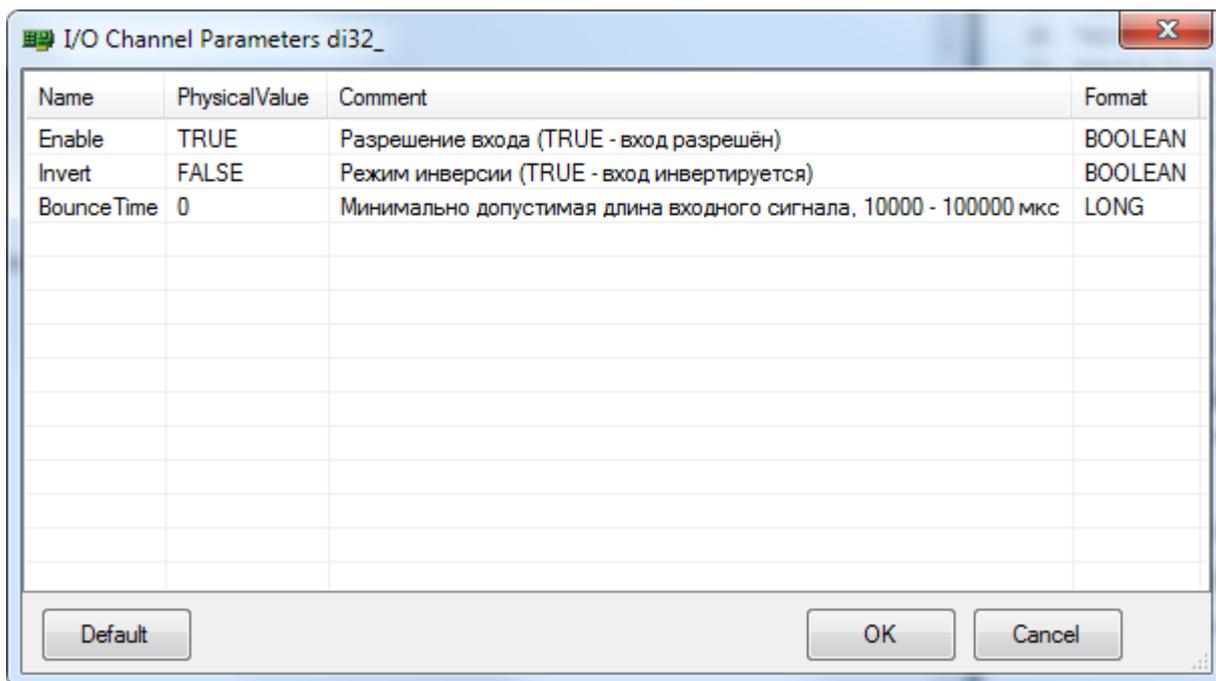


Рис. 3.89 – Описание параметров каналов устройства di32_

Выходные каналы простого устройства di32rsttrig_ предназначены для сброса полей trigger соответствующих входных каналов устройства di32_. Сброс поля trigger производится записью TRUE в выходной канал.

В отличие от di32, di32history имеет в своем составе простое устройство di32history_ с 32 входными каналами данных типа DI32WithTimeStamp (Рис. 3.90). Простое устройство di32rsttrig_ в di32history отсутствует. Также в di32history не определены значения флагов срабатывания триггера в поле trigger устройства di32_, их значения следует игнорировать.

Name	Data Type	String Size	Comment
DI32WithTimeStamp	DI32WithTimeStamp		
DI32WithTimeStamp	DI32WithTimeStamp		
TimeStamp	NowTime		Метка времени для значений входов
TimeStamp.sec_unix	DINT		Число секунд с 1970/01/01
TimeStamp.sec_2000	DINT		Число секунд с 2000/01/01
TimeStamp.msec	DINT		Число миллисекунд в текущей секунде
TimeStamp.usec	DINT		Число микросекунд в текущей секунде
TimeStamp.lastScan_msec	DINT		Время выполнения последнего скана, в миллисекундах
TimeStamp.current_date	DATE		Текущая дата в формате DATE
TimeStamp.current_time	TIME		Текущее время в формате TIME
*			
Values	DI32HistoryArray		Значения всех входов в указанный момент времени
Values[1]	BOOL		
Values[2]	BOOL		
Values[32]	BOOL		

Рис. 3.90 – Описание структуры DI32WithTimeStamp

Простое устройство `di32history_` предназначено для получения изменявшихся значений дискретных входов за время последнего цикла программы пользователя.

В поле `Values` структуры `DI32WithTimeStamp` хранятся значения всех 32 дискретных входов (начиная с самого «старого»). Метки времени изменения входов фиксируются в поле `TimeStamp`. В следующей после завершающей записи в устройстве `di32history_` обнуляется поле `TimeStamp`.

**ВНИМАНИЕ**

Сохранение изменённых значений дискретных входов происходит не чаще, чем раз в 10 мс.

3.12 Работа с модулями дискретного вывода МК-531-032 и МК-531-032 А

Согласно [Табл. 1.1](#), модулю дискретного вывода МК-531-032 в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия `do32`; модулю дискретного вывода МК-531-032 А в среде разработки соответствует модуль изделия `do32a`.

Модули дискретного ввода МК-531-032 и МК-531-032 А отличаются только конструктивно и совершенно не отличаются в работе, поэтому далее будет описана только работа с модулем изделия `do32`.

Кроме диагностического канала, модуль изделия `do32` имеет в своём составе простое устройство `do32_` с 32 выходными каналами данных типа `DO32Outputs` ([Рис. 3.91](#)). Часть каналов устройства `do32_` может работать в режиме широтно-импульсной модуляции (далее ШИМ).

Выходные каналы устройства `do32_` в структуре `DO32Outputs` передают в модуль дискретного вывода изделия значение дискретного выхода в поле `output` либо значение скважности выходного сигнала в поле `pwm` (при работе канала в режиме ШИМ).

Name	Data Type	String Size	Comment
DO32Outputs			
DO32Outputs	DO32Outputs		
output	BOOL		Дискретный выход
pwm	USINT		Выход управления ШИМ

Рис. 3.91 – Описание структуры данных `DO32Outputs`

В режиме ШИМ может работать до 8 каналов устройства `do32_`, при этом несущую частоту можно задавать только для групп каналов ([Рис. 3.92](#) и [Рис. 3.93](#)). 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.

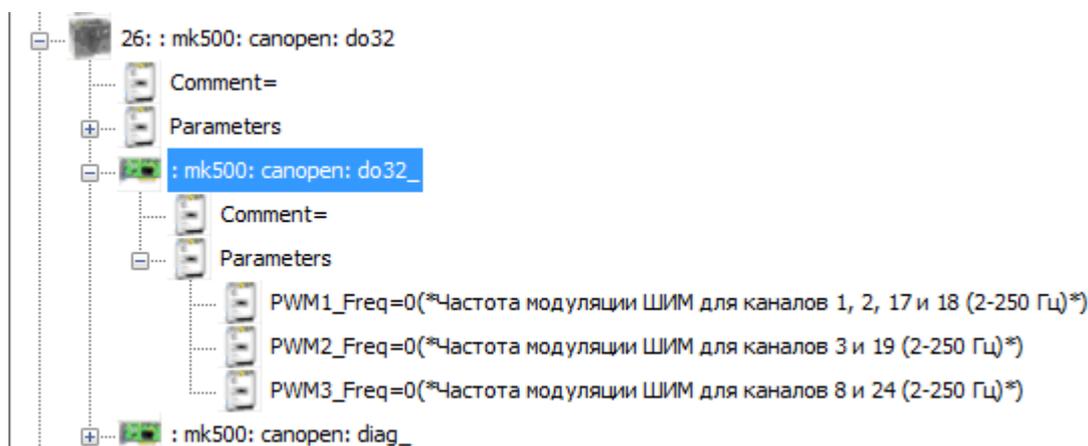


Рис. 3.92 – Параметры устройства `do32_`

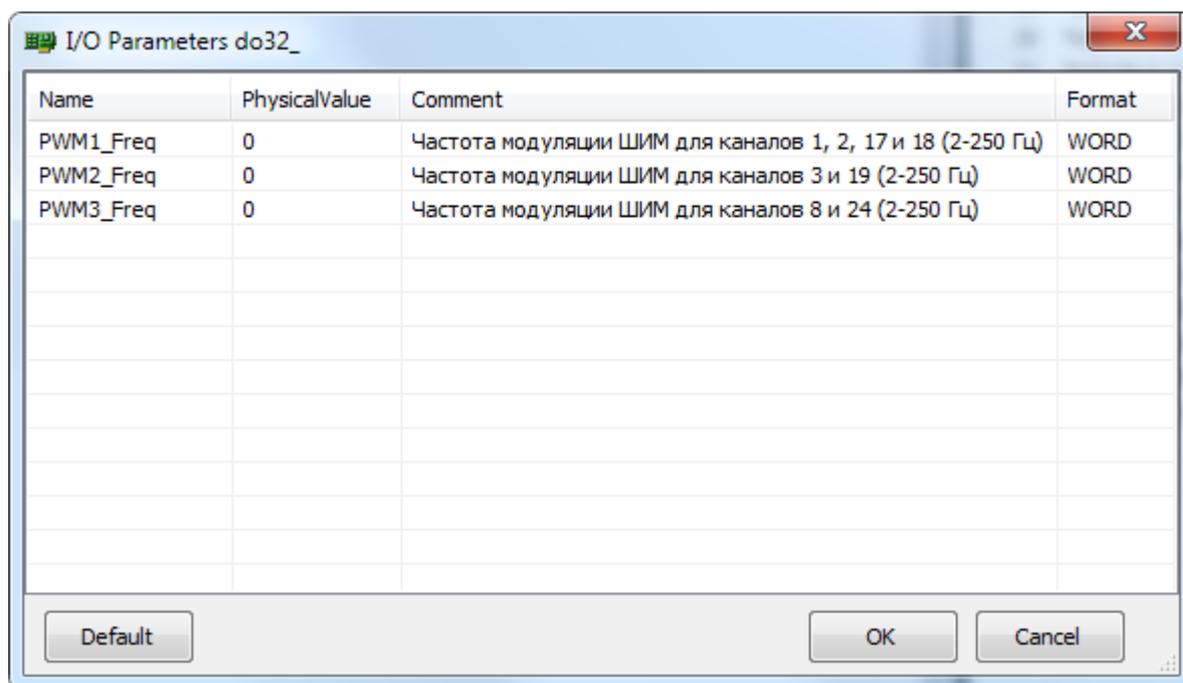


Рис. 3.93 – Окно настройки параметров ШИМ устройства do32_

Также каждый канал устройства do32_ имеет свои независимые параметры (Рис. 3.94).

- Mode – режим работы дискретного выхода. 0 – дискретный выход, 1 – 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 последнему присвоенному ему значению. По умолчанию все каналы разрешены.
- Invert – режим инверсии поля output структуры DO32Outputs канала. 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.

3.13 Работа с коммуникационными модулями МК-541-002

Согласно [Табл. 1.1](#), коммуникационному модулю МК-541-002 в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль `rs485new`.

ВНИМАНИЕ

Следует всегда применять модуль `rs485new`. Модуль `rs485` является устаревшим, не рекомендуется к использованию. С 13.05.2022 модуль `rs485` полностью удалён из системного ПО изделия.

Кроме диагностического канала, модуль `rs485new` имеет в своём составе два простых устройства `rs485port1_` и `rs485port2_` с 64 входными каналами данных типа `BYTE`, и простое устройство `rs485data_` с 1 выходным каналом данных типа `ModbusREGs`.

Модуль `rs485new` также имеет в своем составе два выходных устройства управления `rs485request1_` и `rs485request2_` предназначенные для передачи команд в драйвер Modbus. Устройства `rs485request1_` и `rs485request2_` имеют по 64 переменных типа `ModbusMasterRequest` ([п. 3.7.1.2](#)).

В режиме работы Modbus «ведущий» порт модуля `rs485new` циклически выполняет команды из устройств управления `rs485request1_` и `rs485request2_`, в режиме работы Modbus «ведомый» работает ведомым устройством с фиксированным диапазоном данных с 0 по 959 Holding Registers.

Простые устройства `rs485port1_` и `rs485port2_` служат для конфигурирования портов 1 и 2 коммуникационного модуля изделия, а также для получения диагностической информации о выполняемых каждым из этих портов командах. Оба простых устройства возвращают во входных значениях статус соответствующей команды для режима работы Modbus «ведущий» (расшифровка значений дана в [Табл. 1.11](#)), и не имеют смысла для режима работы Modbus «ведомый».

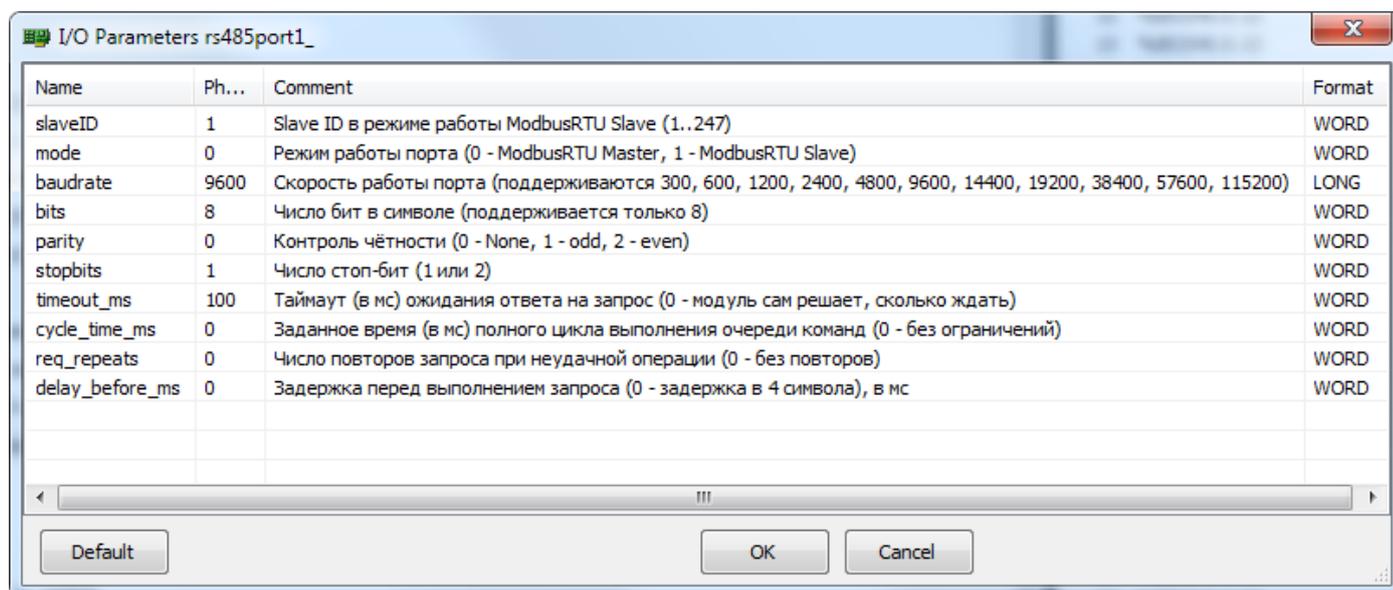


Рис. 3.95 – Окно настройки параметров устройства `rs485port1_` и `rs485port2_`

Конфигурирование портов 1 и 2 коммуникационного модуля изделия выполняется с помощью параметров устройств `rs485port1_` и `rs485port2_` соответственно ([Рис. 3.95](#)). Параметры каналов устройств `rs485port1_` и `rs485port2_`:

- `slaveID` – идентификатор порта на шине Modbus для режима работы «ведомый», по умолчанию установлен 1.
- `mode` – режим работы порта: 0 – ModbusRTU в режиме «ведущий», 1 – ModbusRTU в режиме «ведомый». По умолчанию включен режим 0.
- `baudrate` – скорость работы порта, по умолчанию 9600.
- `bits` – число бит в символе. Поддерживается только режим 8 бит.
- `parity` – контроль чётности: 0 – нет, 1 – even, 2 – odd. По умолчанию 0 – без контроля чётности.
- `stopbits` – число стоп-бит: 1 – 1 стоп-бит, 2 – 2 стоп-бита. По умолчанию 1.
- `timeout_ms` – таймаут ожидания ответа на запрос (в мс) для режима работы Modbus «ведущий», 0 – автоматическое определение времени ожидания ответа. По умолчанию 100 мс.
- `cycle_time_ms` – 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.
- `req_repeats` – число повторов неудачно выполненной команды для режима работы Modbus «ведущий», 0 – нет повторов. По умолчанию 0, без повторов.
- `delay_before_ms` – задержка выполнения следующего запроса в мс для режима «ведущий» сверх стандартных 4 символов. Не может превышать 256 мс, по умолчанию равен 0.

Если какой-либо из параметров лежит вне допустимых пределов (например, выбрано `stopbits=5`), при работе в режиме «ведущий» все Modbus-запросы будут возвращать код 21 ([Табл. 1.11](#)), при работе в режиме «ведомый» этот параметр будет проигнорирован, и будет использоваться последнее корректное значение этого параметра в модуле МК-541-002.

Простое устройство `rs485data_` предназначено для хранения отправляемых и получаемых командами Modbus данных (только типа `WORD`, то есть регистровых данных Modbus) для режима работы Modbus «ведущий», и для формирования адресного пространства устройства на шине Modbus для режима работы Modbus «ведомый».

Устройство `rs485data_` имеет 1 переменную-структуру типа `ModbusREGs` (массив из 1024 переменных типа `WORD`, [Рис. 3.32](#)). В начале цикла выполнения программы канал этого устройства содержит в себе обновившиеся в ходе обработки команд Modbus данные. Поэтому для связанной с этим устройством переменной следует устанавливать атрибут `Read/Write` ([п. 3.1.4](#)), а в начале каждого цикла следует копировать значение переменной в отдельную, не связанную с каналами рабочую переменную.

**ВНИМАНИЕ**

В обоих режимах работы порта поддерживается только 960 регистров из 1024 доступных в переменной типа `ModbusREGs`.

**ВНИМАНИЕ**

Регистровое пространство устройства `rs485data_` используется обоими портами независимо друг от друга. Задача разделения данных между командами обоих портов возлагается на разработчика программы пользователя.

**ВНИМАНИЕ**

При работе с `holding`-регистрами в одном запросе группового чтения или записи (3 и 16 функции) из-за ограничения протокола можно задействовать не больше 123 регистров.

3.13.1 Режимы команд коммуникационного модуля МК-541-002

Режим работы команды задаётся её полями `single_request` и `on_modify_request`.

Для модуля `rs485new` кроме режимов непрерывного опроса и одиночного запроса ([п. 3.7.1](#), описание поля `single_request`) поддерживается режим записи по изменениям ([Табл. 1.32](#)).

Табл. 1.32– Таблица режимов команд МК-541-002

Значения <code>single_request</code> и <code>on_modify_request</code>	Режим команды
<code>single_request=false</code> <code>on_modify_request=false</code>	Режим непрерывного опроса: запросы выполняются по очереди
<code>single_request=true</code> <code>on_modify_request=true</code>	
<code>single_request=true</code> <code>on_modify_request=false</code>	Режим одиночного запроса: запрос выполняется при установке флага <code>do_single_req</code> в TRUE
<code>single_request=false</code> <code>on_modify_request=true</code>	Режим записи по изменениям: проверяется диапазон адресов, на которые ссылается команда (<code>command.slave_data_addr</code> и <code>command.slave_data_length</code>), и в случае их изменения выполняется команда. Данный режим имеет смысл только для функций записи (5, 6, 15 и 16 команды Modbus). Также возможно выполнять запрос по команде (установкой флага <code>do_single_req</code> в TRUE).

3.13.2 Инициализация и передача команд в модуль `rs485new`

Для передачи команд в модуль МК-541-002 следует заполнить соответствующие элементы массивов переменных, привязанных к устройствам `rs485request1_` и `rs485request2_` из состава `rs485new`, см. [Листинг 3.5](#).

```
FOR i := 1 TO 64 DO
  requests_port1[i].enable := true;
  requests_port1[i].single_request := false;
  requests_port1[i].do_single_req := false;
  requests_port1[i].on_modify_request := false;
  requests_port1[i].command.slave_id := ANY_TO_BYTE(i);
  requests_port1[i].command.func_code := 16;
  requests_port1[i].command.slave_data_addr := 0;
  requests_port1[i].command.slave_data_length := 1;
  requests_port1[i].command.offset := ANY_TO_WORD(i*2);
  requests_port1[i].command.repeat_over_scan := true;
  requests_port1[i].command.skip_repeats_when_bad := 5;

  requests_port2[i].enable := true;
  requests_port2[i].single_request := false;
  requests_port2[i].do_single_req := false;
  requests_port2[i].on_modify_request := false;
  requests_port2[i].command.slave_id := ANY_TO_BYTE(i);
  requests_port2[i].command.func_code := 16;
  requests_port2[i].command.slave_data_addr := 0;
  requests_port2[i].command.slave_data_length := 1;
  requests_port2[i].command.offset := ANY_TO_WORD(128 + i*2);
  requests_port2[i].command.repeat_over_scan := true;
  requests_port2[i].command.skip_repeats_when_bad := 5;
END_FOR;
```

Листинг 3.5 – Пример заполнения переменных устройств `rs485request1_` и `rs485request2_` для режима работы порта «ведущий»



ВНИМАНИЕ

Параметры команд `timeout_ms`, `delay_before_ms` и `repeats` (см. [Рис. 3.28](#)) для модуля МК-541-002 реализованы в соответствующих параметрах `rs485port1_` и `rs485port2_`, и не поддерживаются для каждой отдельной команды.

Отличительной особенностью устройства `rs485new` является возможность обновления тела команд 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 поле `status_request` ([Табл. 1.11](#)) устройства `rs485port1_` или `rs485port2_` в любом канале, 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.

3.13.3 Особенности работы Modbus-функций 1, 2, 3 и 4 в модуле `rs485new`

При работе с данными устройства `rs485new` следует учитывать, что если регистр данных (в устройстве `rs485data_`) 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.

3.13.4 Особенности работы Modbus-функций 1, 2, 5 и 15 в модуле rs485new

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

1. Смещение (поле `offset` структуры `ModbusMasterRequest`) 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 не участвующие в операции биты регистров МК-541-002 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.

3.14 Работа с коммуникационными модулями МК-545-010

Согласно [Табл. 1.1](#), коммуникационному модулю МК-545-010 в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль `cn545`.

3.14.1 Реализация протокола Powerlink (ведомый) в модулях `cn545`

Для работы с протоколом Powerlink в режиме ведомый предназначено специальное составное устройство `cn545` ([Рис. 3.96](#)). Только по наличию устройства `cn545` в устройствах проекта определяются корректные `pwl_id`.

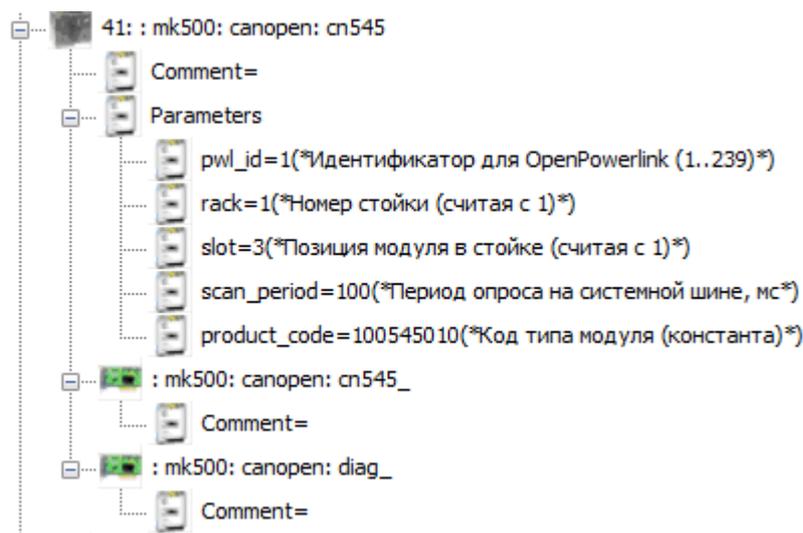


Рис. 3.96 – Описание структуры `cn545`

Устройство `cn545` имеет в своём составе простое входное устройство `cn545_` предназначенное для диагностики работы портов модуля МК-545-010. Устройство имеет 1 канал типа `MK545Diag` ([Рис. 3.97](#)).

Поля `PortBlocking`, `RemoteRXFailure`, `RXFailure`, `Loopback`, `MultipleErrors` и `DiagErrorsCounter` в структуре `PowerlinkPortDiag` предназначены для использования в будущем и на 03.2021 не реализованы; их значения следует игнорировать.

Name	Data Type	String Size	Comment
MK545Diag	MK545Diag		
ports	PowerlinkPorts		Диагностика Powerlink-портов
ports[1]	PowerlinkPortD		
ports[1].Linkup	BOOL		Наличие соединения на порту (TRUE - есть соединение)
ports[1].FullDuplex	BOOL		Режим работы порта (TRUE - полный дуплекс, FALSE - полудуплекс)
ports[1].Speed	BYTE		Скорость порта (0 - 10 МБит/с, 1 - 100 МБит/с, 2 - 1000 МБит/с)
ports[1].PortBlocking	BOOL		TRUE - порт заблокирован
ports[1].RemoteRXFailure	BOOL		TRUE - отсутствует приём на удалённом порту
ports[1].RXFailure	BOOL		TRUE - отсутствует приём на порту
ports[1].Loopback	BOOL		TRUE - наличие на порту петли
ports[1].MultipleErrors	BOOL		TRUE - множественные ошибки на порту
ports[1].DiagErrorsCounter	BYTE		Счётчик ошибок диагностического канала
ports[2]	PowerlinkPortD		
Leds	PowerlinkLeds		Состояние светодиодов передней панели модуля
Leds.Line	BOOL		TRUE - на модуле горит светодиод Line, сеть Powerlink работает в режиме Line
Leds.Ring	BOOL		TRUE - на модуле горит светодиод Ring, сеть Powerlink работает в режиме кольца
Leds.Eth	BOOL		TRUE - на модуле горит светодиод Eth, сеть работает в режиме Ethernet
Leds.Pwl	BOOL		TRUE - на модуле горит светодиод Pwl, сеть работает в режиме Powerlink
version	STRING	16	Версия реализации аппаратной составляющей портов

Рис. 3.97 – Описание структуры MK545Diag

Протокол Powerlink позволяет добавлять в проект модули ввода-вывода и коммуникационные модули удалённой стойки изолированной CAN-шины. Модуль МК-545-010 в удалённой стойке заменяет собой модуль CPU.



ВНИМАНИЕ

Параметр *rack* устройств удалённых стоек не связан с параметром *rack* стоек процессорных модулей, и назначается независимо, как правило, с 1.

Для работы с модулями ввода-вывода и коммуникационными модулями по протоколу Powerlink необходимо в параметрах устройства *cn545* указать идентификатор OpenPowerlink от 1 до 239 (параметр *pwl_id*). Этот идентификатор является одинаковым для всех модулей стойки и его так же нужно указать для всех модулей ввода-вывода стойки. Идентификатор OpenPowerlink для каждой стойки должен быть уникальным.

На 01.03.2021 в одном проекте должно быть не более 40 устройств *cn545* (номера идентификаторов OpenPowerlink должны быть с 1 по 40).

3.15 Работа с модулями аналогового вывода МК-574-008А

Согласно [Табл. 1.1](#), модулю аналогового вывода МК-574-008А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия `ao8ahart`.

По структуре аналогового вывода `ao8ahart` полностью эквивалентен модулю `ao8a` ([п. 3.9](#)), и может использоваться вместо него. Расшифровка кодов ЦАП модуля аналогового вывода приводится в [Табл. 1.27](#)

Для поддержки чтения переменных по протоколу HART модуль `ao8ahart` имеет в своём составе простое устройство `hart8_` с 8 входными каналами типа `HARTChannel`. По каждому каналу допускается чтение по HART до четырёх заранее настроенных переменных.

Для поддержки работы модулей МК-574-008А в резервированной конфигурации модуль `ao8ahart` имеет в своём составе простое устройство `hart_control_` с входным каналом типа `HARTControl`.

Параметры устройства `hart8_` приведены на рисунке ([Рис. 3.98](#)).



Рис. 3.98 – Параметры устройства `hart8_`

Параметр `addrOffset` устройства `hart8_` используется при подключении нескольких модулей МК-574-008А и МК-576-0xxА на общую шину и позволяет задать адреса датчиков на шине HART для каждого канала (см. [Табл. 1.33](#)). Подробнее настройка параметра `addrOffset` описана в [Приложение А. Инструкция по интеграции в PDM-систему](#).

Табл. 1.33– Таблица соответствия адресов датчиков на шине HART для `hart8_`

Значение <code>addrOffset</code>	Диапазон адресов датчиков на шине HART (с 1 по 8 каналы)
0	0..7
1	8..15
2	16..23

Значение addrOffset	Диапазон адресов датчиков на шине HART (с 1 по 8 каналы)
3	24..31
4	32..39
5	40..47
6	48..55
7	56..63
8..255	Не используются

Также каждый канал устройства `hart8_` имеет свои независимые параметры ([Рис. 3.99](#)).

Параметры каналов устройства `hart8_`:

- `hartVarUsed` – число переменных HART, которые будут считываться по данному каналу, от 0 до 4. Указывается именно 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_ms` – таймаут ответа на запросы по протоколу HART в мс, допустимые значения 1000...10000.
- `DelayBetweenRequest_ms` – задержка между запросами разных мастеров в мс, допустимые значения 0...500.

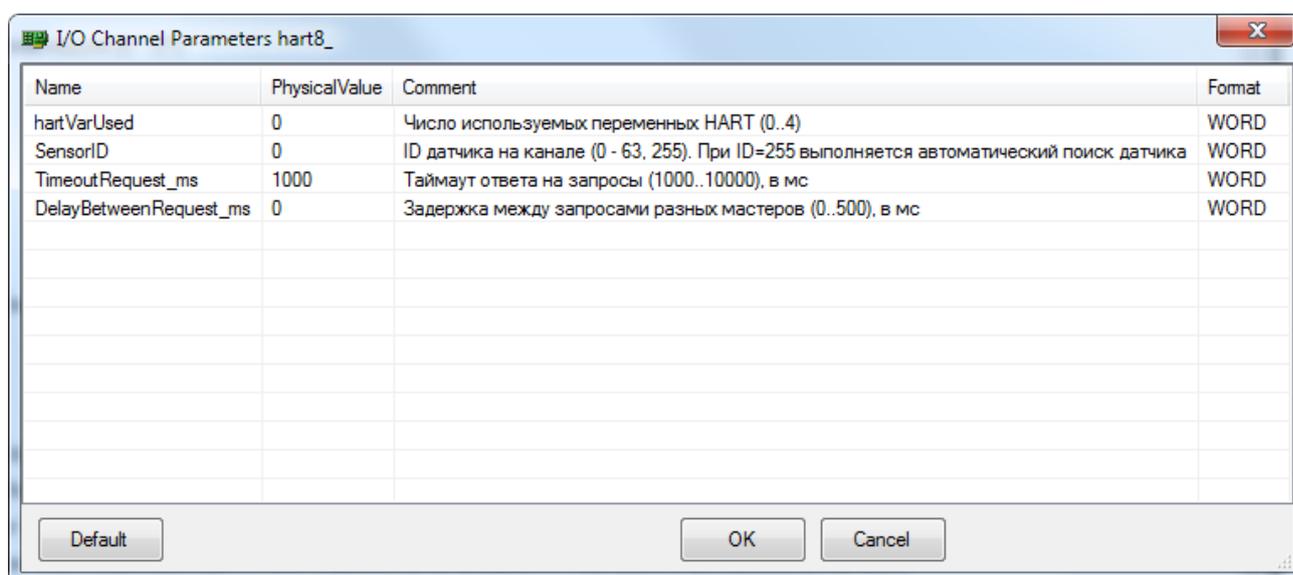


Рис. 3.99 – Описание параметров каналов устройства `hart8_`

Кроме значений переменных по каждому каналу устройства `hart8_` в структуре `HARTChannel` передается диагностическая информация ([Рис. 3.100](#), [Рис. 3.101](#), [Рис. 3.102](#)).

Для отключения HART на определённом модуле МК-574-008А (при работе в резерве), а также для контроля работы HART, следует использовать устройство `hart_control_` (см. [Рис. 3.103](#)).

Инструкции по интеграции модуля МК-574-008А в PDM-системы можно найти в [Приложение А. Инструкция по интеграции в PDM-систему](#).

Name	Data Type	String Size	Comment
HARTChannel			
- HARTChannel	HARTChannel		
hartSelected	USINT		Число выбранных переменных HART (от 0 до 4)
ID	BYTE		ID опрашиваемого устройства
responseError	BOOL		TRUE если нет ответа от устройства
configError	BOOL		TRUE если неверная конфигурация канала (неверное число переменных)
HARTFault	BOOL		TRUE если отказ HART канала
+ commandStatus	HARTCmdStatus		Флаги обработки команды
+ communicationError	HARTCommError:		Флаги коммуникационных ошибок
▶ - HART	HARTVars		Массив переменных HART
HART[1]	REAL		
HART[2]	REAL		
HART[3]	REAL		
HART[4]	REAL		

Рис. 3.100 – Описание структуры HARTChannel

Name	Data Type	String Size	Comment
HartCmdStatus			
- HARTCmdStatus	HARTCmdStatus		
noCommand	BOOL		TRUE если нет команды
wrongChoose	BOOL		TRUE если неверный адрес для опроса
paramTooBig	BOOL		TRUE если последний принятый параметр слишком велик
paramTooSmall	BOOL		TRUE если последний принятый параметр слишком мал
fewData	BOOL		TRUE если получено недостаточно байт данных
writeprotect	BOOL		TRUE если для переменной выставлен режим защиты от записи
accessDenied	BOOL		TRUE если для переменной ограничен доступ
dataNotUpdates	BOOL		TRUE если не обновляются данные переменной
deviceBusy	BOOL		TRUE если устройство занято
commandNotImplemented	BOOL		TRUE если команда не реализована
deviceMalfunction	BOOL		TRUE при неисправности устройства
configChanged	BOOL		TRUE если изменена конфигурация устройства
coldStart	BOOL		TRUE если выполняется "холодный старт" устройства
moreStatusAvailable	BOOL		TRUE если доступна дополнительная информация о статусе устройства
AOFixed	BOOL		TRUE если значение аналогового выхода фиксировано
AOSaturated	BOOL		TRUE если аналоговый выход в насыщении
primaryOutOfLimits	BOOL		TRUE если значение главной переменной вне установленных пределов
nonPrimaryOutOfLimits	BOOL		TRUE если значение главной не переменной вне установленных пределов

Рис. 3.101 – Описание структуры HARTCmdStatus

Name	Data Type	String Size	Comment
HartCommErrors			
- HARTCommErrors	HARTCommErrors		
parity	BOOL		TRUE если обнаружена ошибка чётности
speedOverflow	BOOL		TRUE если обнаружена перегрузка по скорости
sync	BOOL		TRUE если обнаружена ошибка синхронизации или формата посылки
CRC	BOOL		TRUE если обнаружена ошибка контрольной суммы
bufferOverflow	BOOL		TRUE если обнаружено переполнение буфера приёмника

Рис. 3.102 – Описание структуры HARTCommErrors

Name	Data Type	String Size	Comment
HartControl			
- HARTControl	HARTControl		
disableHART	BOOL		Команда отключения HART (TRUE - отключить HART, FALSE - включить)
HARTStatus	BOOL		Флаг состояния HART в модуле (TRUE - включен, FALSE - отключен)

Рис. 3.103 – Описание структуры HARTControl

3.16 Работа с модулями аналогового ввода МК-576-008А

Согласно [Табл. 1.1](#), модулям аналогового ввода МК-576-008А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия `ai8ahart`.

По структуре аналогового вывода `ai8ahart` полностью эквивалентен модулю `ai8a` ([п. 3.10](#)), и может использоваться вместо него. Расшифровка кодов АЦП модуля аналогового ввода приводится в [Табл. 1.30](#).

Для поддержки чтения переменных по протоколу HART модуль `ai8ahart` имеет в своём составе простое устройство `hart8_` с 8 входными каналами типа `HARTChannel`. По каждому каналу допускается чтение по HART до четырёх заранее настроенных переменных. Назначение и параметры устройства `hart8_` полностью аналогичны описаным в [п.3.15](#).

Инструкции по интеграции модуля МК-576-008А в PDM-системы можно найти в [Приложение А. Инструкция по интеграции в PDM-систему](#).

3.17 Работа с модулями аналогового ввода МК-576-016А

Согласно [Табл. 1.1](#), модулям аналогового ввода МК-576-016А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия `ai16ahart`.

Кроме диагностического канала, модуль изделия `ai16ahart` имеет в своём составе простое устройство `ai16_` с 16 входными каналами данных типа `WORD` и простое устройство `hart16_` с 16 входными каналами типа `HARTChannel`.

Значение каждого канала устройства `ai16_` 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. 1.30](#) (аналогично модулям МК-516-008 и МК-576-008А).

Для поддержки чтения переменных по протоколу HART модуль `ai16ahart` имеет в своём составе простое устройство `hart16_` с 16 входными каналами типа `HARTChannel`. По каждому каналу допускается чтение по HART до четырёх заранее настроенных переменных.

Для поддержки работы модулей МК-576-016А в резервированной конфигурации модуль `ai16ahart` имеет в своём составе простое устройство `hart_control_` с входным каналом типа `HARTControl`.

Параметры устройства `hart16_` приведены на [Рис. 3.104](#)

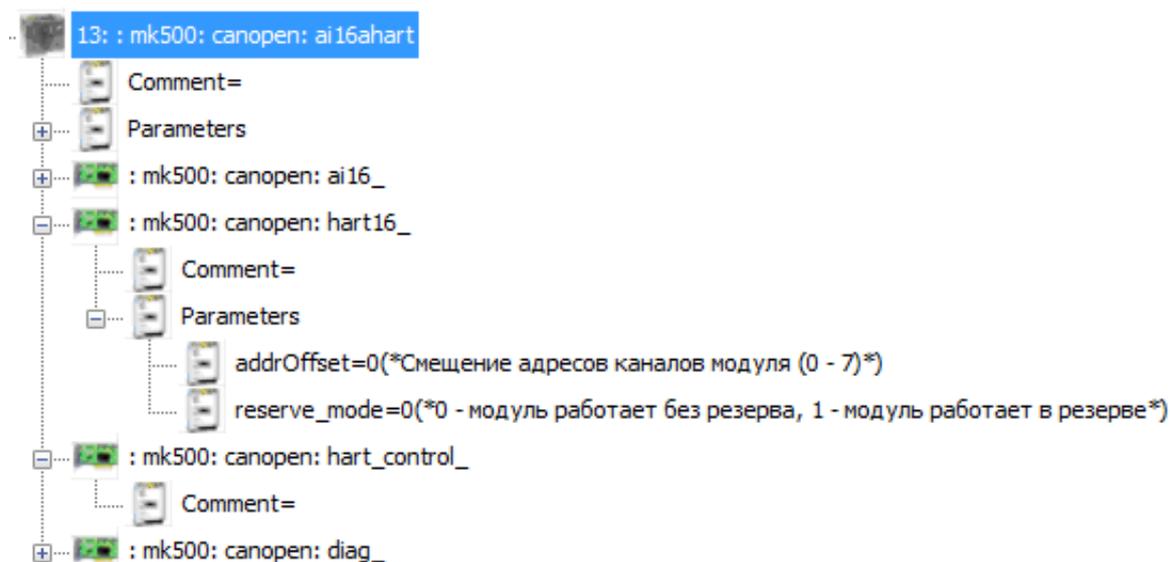


Рис. 3.104 – Параметры устройства `hart16_`

Параметр `addrOffset` устройства `hart16_` используется при подключении нескольких модулей МК-574-008А и МК-576-0xxА на общую шину и позволяет задать адреса датчиков на шине HART для каждого канала (см. [Табл. 1.34](#)). Подробнее настройка параметра `addrOffset` описана в [Приложение А. Инструкция по интеграции в PDM-систему](#).

Табл. 1.34– Таблица соответствия адресов датчиков на шине HART для hart16_

Значение addrOffset	Диапазон адресов датчиков на шине HART (с 1 по 8 каналы)
0	0..15
1	8..23
2	16..31
3	24..39
4	32..47
5	40..55
6	48..63
7..255	Не используется

Параметр `reserve_mode` устройства `hart16_` должен быть равен 1, если модуль МК-576-016А работает с резервированием (см. раздел 15.2 руководства по эксплуатации КДСА.426471.004 РЭ_2.0_00)

Назначение и параметры каналов устройства `hart16_` полностью аналогичны назначению и параметрам каналов устройства `hart8_`, описаным в [п. 3.15](#).

Для отключения HART на определённом модуле МК-576-016А (при работе в резерве), а также для контроля работы HART, следует использовать устройство `hart_control_` (см. [Рис. 3.103](#)).

Инструкции по интеграции модуля МК-576-016А в PDM-системы можно найти в [Приложение А. Инструкция по интеграции в PDM-систему](#).

3.18 Работа с модулями дискретного ввода NAMUR МК-523-032А

Согласно [Табл. 1.1](#), модулям аналогового ввода МК-523-032А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия `di32anamur`.

Кроме диагностического канала, модуль изделия `di32anamur` имеет в своём составе простое устройство `namur32_` с 32 входными каналами данных типа `DI32Namur` (см. [Рис. 3.105](#)).

Name	Data Type	String Size	Comment
DI32Namur			
DI32Namur	DI32Namur		
Input	BOOL		Логическое значение входа
IsValid	BOOL		TRUE если нет ошибок Cutoff, ShortCircuit, Bounce, Fault или CableFault
Cutoff	BOOL		TRUE если на входе обрыв
ShortCircuit	BOOL		TRUE если на входе короткое замыкание
Bounce	BOOL		TRUE если входной сигнал "дребезжит"
Fault	BOOL		TRUE если вход неисправен (аппаратный отказ)
CableFault	BOOL		TRUE если терминальный кабель повреждён

Рис. 3.105 – Описание структуры данных DI32Namur

Входные каналы устройства `namur32_` в структуре `DI32Namur` возвращают текущее значение дискретного входа в поле `Input` и флаги отказов.

Также каждый канал устройства `namur32_` имеет свои независимые параметры (см. [Рис. 3.106](#)).

Параметры каналов устройства `namur32_`:

- `Enabled` – разрешение работы канала. FALSE – запрещён, TRUE – разрешён. Для запрещённого канала все поля структуры `DI32Namur` всегда равны FALSE. По умолчанию все каналы разрешены.
- `Invert` – режим инверсии поля `Input` структуры `DI32Namur` канала. FALSE – инверсия выключена, TRUE – инверсия включена. По умолчанию инверсия всех каналов выключена.
- `BounceTime` – минимально допустимая длина входного сигнала на канале, от 0 до 1000 мс, с шагом 100 мс. Входной сигнал длительностью меньше значения `BounceTime` игнорируется. Данный параметр используется для защиты входов отдребезга. По умолчанию значение `BounceTime` всех каналов равно 0 мс.

3.19 Работа с модулями дискретного вывода NAMUR МК-532-032А

Согласно [Табл. 1.1](#), модулям аналогового ввода МК-532-032А в среде разработки ACP Workbench ISaGRAF 6.50 соответствует модуль изделия do32anamur.

Кроме диагностического канала, модуль изделия do32anamur имеет в своём составе простое устройство do32namur_ с 32 выходными каналами данных типа bool для управления выходами, и простое устройств do32namurdiag_ с 32 входными каналами данных типа DO32Namur (см. [Рис. 3.108](#)).

Name	Data Type	String Size	Comment
DO32Namur			
DO32Namur	DO32Namur		
Output	BOOL		Текущее состояние выхода (TRUE - замкнут, FALSE - разомкнут)
IsValid	BOOL		TRUE если нет ошибок Cutoff, ShortCircuit, PowerOff или Fault
Cutoff	BOOL		TRUE если на выходе обрыв
ShortCircuit	BOOL		TRUE если на выходе короткое замыкание
PowerOff	BOOL		TRUE если отсутствует напряжение питания группы выходов (1..16 или 17..32)
Fault	BOOL		TRUE если имеется внутренний отказ группы выходов (1...16 или 17..32)

Рис. 3.108 – Описание структуры данных DO32Namur

Входные каналы устройства do32namur_ в структуре DO32Namur возвращают текущее значение дискретного выхода в поле Output и флаги отказов.

Параметры устройства do32namur_ приведены на [Рис. 3.109](#).

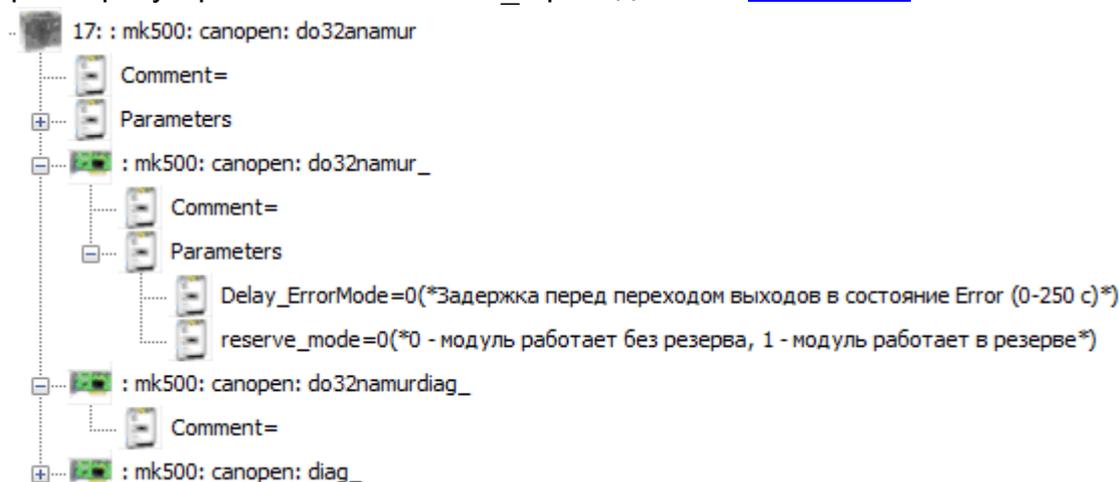


Рис. 3.109 – Параметры устройства do32namur_

Параметр reserve_mode устройства do32namur_ должен быть равен 1, если модуль МК-532-032А работает с резервированием (см. раздел 15.2 руководства по эксплуатации КДСА.426471.004 РЭ_2.0_00).

Глава 4 Использование функций расширения МК500

К функциям расширения МК500 относятся функции и функциональные блоки, добавленные в стандартную библиотеку ISaGRAF с целью расширения возможностей среды исполнения ISaGRAF. Подробнее о стандартной библиотеке ISaGRAF сказано в документах «Функциональные блоки» (lrsb_ISa6_ru), «Функции» (lrsf_ISa6_ru) и «Стандартные операции» (lrso_ISa6_ru).

Для использования функций расширения, поддерживаемых процессорным модулем изделия, следует открыть окно BlockLibrary, выполнив VIEW→Block Library (Рис. 4.1), В окне BlockLibrary следует нажать правую кнопку мыши и в контекстном меню выбрать пункт «Category» (Рис. 4.2). Из списка доступных категорий следует выбрать вкладку «МК500» (Рис. 4.3) и добавить необходимые функции расширения в окно разработки прикладной программы перетаскиванием.

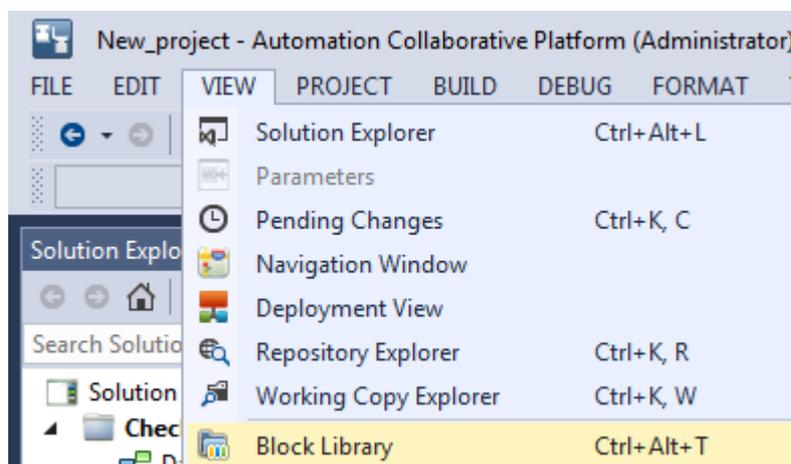


Рис. 4.1 – Открытие окна BlockLibrary

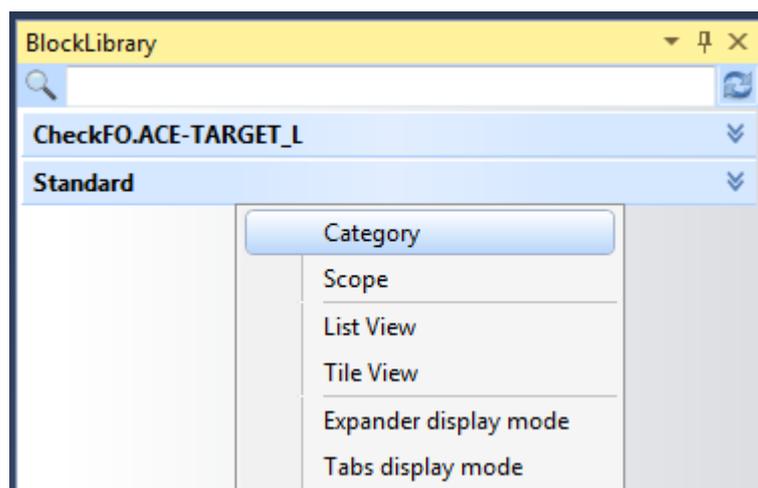


Рис. 4.2 – Переключение окна BlockLibrary в режим просмотра по категориям

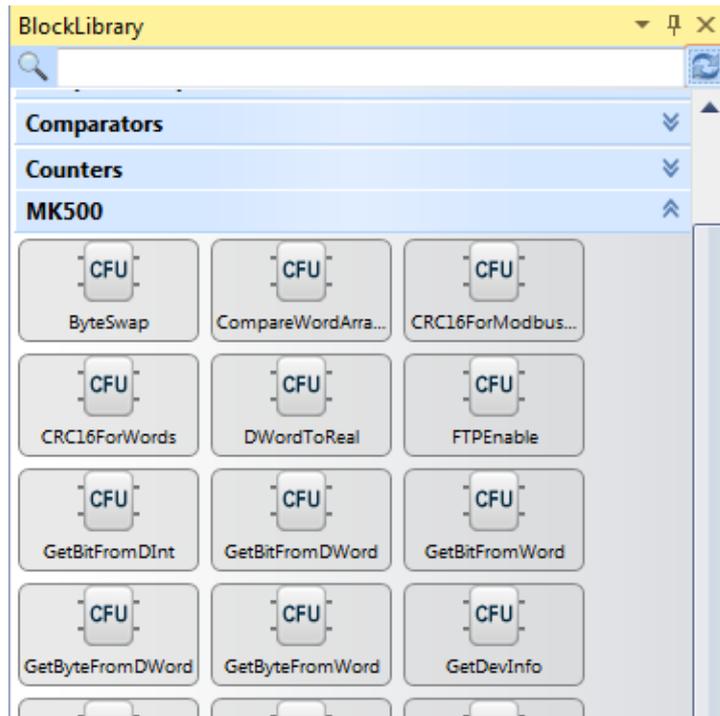


Рис. 4.3 – Окно BlockLibrary с открытой категорией функций расширения МК500

4.1 Функции ByteSwap и WordSwap

Функции `ByteSwap` и `WordSwap` меняют местами в переменной-аргументе байты и слова соответственно, и возвращают их в возвращаемой переменной.

Данные функции могут применяться для согласования данных, принятых либо передаваемых на устройства с отличающимся порядком байт и/или слов.

```
res := ByteSwap(value);
```

Табл. 1.35– Параметры и возвращаемые значения функции ByteSwap

Имя параметра	Тип параметра	Описание
res	WORD	Возвращаемое значение с переставленными байтами
value	WORD	Исходная переменная, в которой требуется поменять байты местами

```
res := WordSwap(value);
```

Табл. 1.36– Параметры и возвращаемые значения функции WordSwap

Имя параметра	Тип параметра	Описание
res	DWORD	Возвращаемое значение с переставленными словами
value	DWORD	Исходная переменная, в которой требуется поменять слова местами

Параметры и возвращаемые значения функций `ByteSwap` и `WordSwap` приведены в [Табл. 1.35](#) и [Табл. 1.36](#) соответственно. Пример использования функций приведён в [Листинг 4.1](#).

```
wData := 16#1234;
dData := 16#12345678;

new_wData := ByteSwap(wData);
(*new_wData = 16#3412*)

new_dData := WordSwap(dData);
(*new_dData = 16#56781234*)
```

Листинг 4.1 – Использование функций ByteSwap и WordSwap

4.2 Функции WordsToReal и RealToWords

Функции `WordsToReal` и `RealToWords` выполняют побитное копирование указанного числа байт из массива типа `WORD` в массив типа `REAL` и из массива типа `REAL` в массив типа `WORD` соответственно.

Данные функции могут применяться для преобразования данных, принятых либо передаваемых по протоколу Modbus (который умеет оперировать только данными типа `BOOL` и `WORD`).

```
res := WordsToReal(wArray, wOffset, rArray, rOffset, size);
```

Табл. 1.37– Параметры и возвращаемые значения функции WordsToReal

Имя параметра	Тип параметра	Описание
<code>res</code>	<code>INT</code>	Результат выполнения преобразования: 0 – успешно, -1 – неверные параметры
<code>wArray</code>	<code>WORD[a..b]</code>	Массив данных типа <code>WORD</code> , подлежащих преобразованию в данные типа <code>REAL</code> (младшими словами вперёд). Допускается произвольная допустимая размерность массива
<code>wOffset</code>	<code>INT</code>	Смещение в массиве <code>wArray</code> (в единицах индекса массива), начиная с которого будет выполняться преобразование
<code>rArray</code>	<code>REAL[c..d]</code>	Массив данных типа <code>REAL</code> , в который будут помещаться преобразованные значения. Допускается произвольная допустимая размерность массива
<code>rOffset</code>	<code>INT</code>	Смещение в массиве <code>rArray</code> (в единицах индекса массива), начиная с которого будут помещаться преобразованные данные
<code>size</code>	<code>INT</code>	Число преобразуемых байт данных массива <code>wArray</code>

Параметры и возвращаемые значения функций `WordsToReal` и `RealToWords` приведены в [Табл. 1.37](#) и [Табл. 1.38](#) соответственно. Пример использования функций приведён в [Листинг 4.2](#).

```
res := RealToWords(rArray, rOffset, wArray, wOffset, size);
```

Табл. 1.38– Параметры и возвращаемые значения функции RealToWords

Имя параметра	Тип параметра	Описание
<code>res</code>	<code>INT</code>	Результат выполнения преобразования: 0 – успешно, -1 – неверные параметры
<code>rArray</code>	<code>REAL[a..b]</code>	Массив данных типа <code>REAL</code> , подлежащих преобразованию в данные типа <code>WORD</code> . Допускается произвольная допустимая размерность массива
<code>rOffset</code>	<code>INT</code>	Смещение в массиве <code>rArray</code> (в единицах индекса массива), начиная с которого будет выполняться преобразование
<code>wArray</code>	<code>WORD[c..d]</code>	Массив данных типа <code>WORD</code> , в который будут помещаться

		преобразованные значения (младшими словами вперёд). Допускается произвольная допустимая размерность массива
wOffset	INT	Смещение в массиве wArray (в единицах индекса массива), начиная с которого будут помещаться преобразованные данные
size	INT	Число преобразуемых байт данных массива rArray

Преобразование не будет выполнено (функции вернут -1) в случае, если после приведения переданных параметров будет получаться нулевой или отрицательный размер копируемых данных и/или отрицательный размер смещения внутри любого массива. Во всех прочих случаях будет выполнено преобразование исходя из минимально общего размера данных (из размеров областей данных внутри массивов и указанного числа байт).

```
(* wArray - WORD[1..4], rArray - REAL[1..2] *)
wArray[1] := 16#0000;
wArray[2] := 16#3F80;
wArray[3] := 16#0000;
wArray[4] := 16#4120;
res := WordsToReal(wArray, 0, rArray, 0, 8);
(* res = 0, rArray[1] = 1.0, rArray[2] = 10.0 *)

rArray[2] := rArray[2] + 0.1;
res := RealToWords(rArray, 1, wArray, 2, 4);
(* res = 0, wArray[3] = 16#99A4, wArray[4] = 16#4121 *)
```

Листинг 4.2 – Использование функций WordsToReal и RealToWords

4.3 Функции WordsToLReal и LRealToWords

Функции `WordsToLReal` и `LRealToWords` выполняют побитное копирование указанного числа байт из массива типа `WORD` в массив типа `LREAL` и из массива типа `LREAL` в массив типа `WORD` соответственно.

Данные функции могут применяться для преобразования данных, принятых либо передаваемых по протоколу Modbus (который умеет оперировать только данными типа `BOOL` и `WORD`).

```
res := WordsToLReal(wArray, wOffset, lrArray, lrOffset, size);
```

Табл. 1.39– Параметры и возвращаемые значения функции WordsToLReal

Имя параметра	Тип параметра	Описание
<code>res</code>	<code>INT</code>	Результат выполнения преобразования: 0 – успешно, -1 – неверные параметры
<code>wArray</code>	<code>WORD[a..b]</code>	Массив данных типа <code>WORD</code> , подлежащих преобразованию в данные типа <code>LREAL</code> (младшими словами вперёд). Допускается произвольная допустимая размерность массива
<code>wOffset</code>	<code>INT</code>	Смещение в массиве <code>wArray</code> (в единицах индекса массива), начиная с которого будет выполняться преобразование
<code>lrArray</code>	<code>LREAL[c..d]</code>	Массив данных типа <code>LREAL</code> , в который будут помещаться преобразованные значения. Допускается произвольная допустимая размерность массива
<code>lrOffset</code>	<code>INT</code>	Смещение в массиве <code>lrArray</code> (в единицах индекса массива), начиная с которого будут помещаться преобразованные данные
<code>size</code>	<code>INT</code>	Число преобразуемых байт данных массива <code>wArray</code>

Параметры и возвращаемые значения функций `WordsToReal` и `RealToWords` приведены в [Табл. 1.39](#) и [Табл. 1.40](#) соответственно. Пример использования функций приведён в [Листинг 4.3](#).

```
res := LRealToWords(lrArray, lrOffset, wArray, wOffset, size);
```

Табл. 1.40– Параметры и возвращаемые значения функции LRealToWords

Имя параметра	Тип параметра	Описание
<code>res</code>	<code>INT</code>	Результат выполнения преобразования: 0 – успешно, -1 – неверные параметры
<code>lrArray</code>	<code>LREAL[a..b]</code>	Массив данных типа <code>LREAL</code> , подлежащих преобразованию в данные типа <code>WORD</code> . Допускается произвольная допустимая размерность массива
<code>lrOffset</code>	<code>INT</code>	Смещение в массиве <code>lrArray</code> (в единицах индекса массива), начиная с которого будет выполняться преобразование
<code>wArray</code>	<code>WORD[c..d]</code>	Массив данных типа <code>WORD</code> , в который будут помещаться

Имя параметра	Тип параметра	Описание
		преобразованные значения (младшими словами вперёд). Допускается произвольная допустимая размерность массива
wOffset	INT	Смещение в массиве wArray (в единицах индекса массива), начиная с которого будут помещаться преобразованные данные
size	INT	Число преобразуемых байт данных массива lrArray

Преобразование не будет выполнено (функции вернут -1) в случае, если после приведения переданных параметров будет получаться нулевой или отрицательный размер копируемых данных и/или отрицательный размер смещения внутри любого массива. Во всех прочих случаях будет выполнено преобразование исходя из минимально общего размера данных (из размеров областей данных внутри массивов и указанного числа байт).

```
(* wArray - WORD[1..8], lrArray - LREAL[1..2] *)
wArray[1] := 16#0000;
wArray[2] := 16#0000;
wArray[3] := 16#0000;
wArray[4] := 16#3FF0;
wArray[5] := 16#0000;
wArray[6] := 16#0000;
wArray[7] := 16#0000;
wArray[8] := 16#4024;

res := WordsToLReal(wArray, 0, lrArray, 0, 16);
(* res = 0, lrArray[1] = 1.0, lrArray[2] = 10.0 *)

lrArray[2] := lrArray[2] + 1.000001;
res := LRealToWords(lrArray, 1, wArray, 4, 8);
(* res = 0,
wArray[5]=16#EF41, wArray[6]=16#218D, wArray[7]=16#0000, wArray[8]=16#4026 *)
```

Листинг 4.3 – Использование функций WordsToReal и RealToWords

4.4 Функции DWordToReal и RealToDWord

Функции `DWordToReal` и `RealToDWord` выполняют побитное копирование из переменной типа `DWORD` в переменную типа `REAL` и, из переменной типа `REAL` в переменную типа `DWORD` соответственно.

Параметры и возвращаемые значения функций `DWordToReal` и `RealToDWord` приведены в [Табл. 1.41](#) и [Табл. 1.42](#) соответственно. Пример использования функций приведён в [Листинг 4.4](#).

```
VarReal := DWordsToReal(VarDWord);
```

Табл. 1.41– Параметры и возвращаемые значения функции DWordToReal

Имя параметра	Тип параметра	Описание
VarReal	REAL	Переменная типа REAL, в которую будет помещаться преобразованное значение
VarDWord	DWORD	Переменная типа DWORD, подлежащая преобразованию в переменную типа REAL

```
VarDWord := RealToDWord(VarReal);
```

Табл. 1.42– Параметры и возвращаемые значения функции RealToDWord

Имя параметра	Тип параметра	Описание
VarDWord	DWORD	Переменная типа DWORD, в которую будет помещаться преобразованное значение
VarReal	REAL	Переменная типа REAL, подлежащая преобразованию в переменную типа DWORD

```
(* VarDWord - DWORD, VarReal - REAL *)
VarDWord:= 1092616192;
VarReal := DWordsToReal(VarDWord)
(* VarReal = 10.0 *)

VarReal := 0.004723788;
VarDWord := RealToDWord(VarDWord);
(* VarDWord = 1000000001 *)
```

Листинг 4.4 – Использование функций DWordToReal и RealToDWord

4.5 Функция Safe2DimWordArrayCopy

Функция `Safe2DimWordArrayCopy` выполняет копирование данных из одного двумерного массива `WORD` в другой двумерный массив `WORD`, не совпадающий по размерностям (иначе можно копировать присваиванием).

Параметры и возвращаемые значения функции `Safe2DimWordArrayCopy` приведены в [Табл. 1.43](#), пример использования функции приведён в [Листинг 4.5](#).

```
res := Safe2DimWordArrayCopy (wArray_in, wArray_out, row_in, column_in,
row_out, column_out, count);
```

Табл. 1.43– Параметры и возвращаемые значения функции `Safe2DimWordArrayCopy`

Имя параметра	Тип параметра	Описание
<code>res</code>	DINT	Результат выполнения преобразования: 0 – успешно, -1 – неверные параметры
<code>wArray_in</code>	WORD	Массив данных типа <code>WORD</code> , из которого будет выполняться копирование. Допускается произвольная размерность массива
<code>wArray_out</code>	WORD	Массив данных типа <code>WORD</code> , в который будет выполняться копирование. Допускается произвольная размерность массива
<code>row_in</code>	DINT	Первый индекс смещение в массиве <code>wArray_in</code> , начиная с которого будет выполняться копирование
<code>column_in</code>	DINT	Второй индекс смещение в массиве <code>wArray_in</code> , начиная с которого будет выполняться копирование
<code>row_out</code>	DINT	Первый индекс смещение в массиве <code>wArray_out</code> , начиная с которого будут помещаться скопированные данные
<code>column_out</code>	DINT	Второй индекс смещение в массиве <code>wArray_out</code> , начиная с которого будут помещаться скопированные данные
<code>count</code>	DINT	Число копируемых переменных типа <code>WORD</code>

```
(* wArray_in - WORD [1..3,1..10], wArray_out - WORD [1..10,1..10] *)
wArray_in[1,1]:=10;
wArray_in[2,5]:=15;
wArray_in[3,10]:=20;
res := Safe2DimWordArrayCopy (wArray_in, wArray_out,1,1,5,1,30);
(* res = 0,
wArray_out[5,1] = 10, wArray_out[6,5] = 15 wArray_out[7,10] = 20 *)
```

Листинг 4.5 – Использование функции `Safe2DimWordArrayCopy`

4.6 Функции SafeBoolArrayCopy и SafeWordArrayCopy

Функции `SafeBoolArrayCopy` и `SafeWordArrayCopy` выполняют безопасное копирование данных из одного массива в другой для данных типа `BOOL` и `WORD` соответственно. Под безопасностью копирования подразумевается проверка размерностей массивов и копирование минимально совпадающего количества данных.

Параметры и возвращаемые значения функций `SafeBoolArrayCopy` и `SafeWordArrayCopy` приведены в [Табл. 1.44](#) и [Табл. 1.45](#) соответственно, пример использования функций приведён в [Листинг 4.6](#).

```
res := SafeBoolArrayCopy(data_in, data_out, offset_in, offset_out, count);
```

Табл. 1.44– Параметры и возвращаемые значения функции SafeBoolArrayCopy

Имя параметра	Тип параметра	Описание
res	DINT	Результат выполнения преобразования: ≥ 0 – число скопированных переменных типа <code>BOOL</code> , -1 – попытка скопировать 0 или меньше переменных типа <code>BOOL</code> , -2 – одно из значений индекса массива находится за пределами границ массива
data_in	BOOL[a..b]	Массив данных типа <code>BOOL</code> , из которого будет выполняться копирование. Допускается произвольная допустимая размерность массива.
data_out	BOOL[c..d]	Массив данных типа <code>BOOL</code> , в который будет выполняться копирование. Допускается произвольная допустимая размерность массива
offset_in	DINT	Индекс столбца в массиве <code>data_in</code> , начиная с которого будет выполняться копирование
offset_out	DINT	Индекс столбца в массиве <code>data_out</code> , начиная с которого будут помещаться скопированные данные
count	DINT	Число копируемых элементов массива <code>BOOL</code>

```
res := SafeWordArrayCopy(data_in, data_out, offset_in, offset_out, count);
```

Табл. 1.45– Параметры и возвращаемые значения функции SafeWordArrayCopy

Имя параметра	Тип параметра	Описание
res	INT	Результат выполнения преобразования: ≥ 0 – число скопированных переменных типа WORD, -1 – попытка скопировать 0 или меньше переменных типа WORD, -2 – одно из значений индекса массива находится за пределами границ массива
data_in	WORD[a..b]	Массив данных типа WORD, из которого будет выполняться копирование. Допускается произвольная допустимая размерность массива
data_out	WORD[c..d]	Массив данных типа WORD, в который будет выполняться копирование. Допускается произвольная допустимая размерность массива
offset_in	DINT	Индекс столбца в массиве data_in, начиная с которого будет выполняться копирование
offset_out	DINT	Индекс столбца в массиве data_out, начиная с которого будут помещаться скопированные данные
count	DINT	Число копируемых элементов массива WORD

```
(* bArray1 - BOOL[1..100], bArray2 - BOOL[1..200] *)
res := SafeBoolArrayCopy(bArray1, bArray2, 1, 101, 100);
```

```
(* wArray1 - WORD[1..20], wArray2 - WORD[1..5] *)
res := SafeWordArrayCopy(wArray1, wArray2, 5, 1, 5);
```

Листинг 4.6 – Использование функций SafeBoolArrayCopy и SafeWordArrayCopy

4.7 Функции SafeCopyFromModbusREGsArray и SafeCopyToModbusREGsArray

Функции `SafeCopyFromModbusREGsArray` и `SafeCopyToModbusREGsArray` выполняют копирование данных из массива `ModbusREGs` в двумерный массив для данных типа `WORD` и копирование данных из двумерного массива для данных типа `WORD` в массив `ModbusREGs`.

Параметры и возвращаемые значения функций `SafeCopyFromModbusREGsArray` и `SafeCopyToModbusREGsArray` приведены в [Табл. 1.46](#) и [Табл. 1.47](#) соответственно, пример использования функций приведён в [Листинг 4.7](#).

```
res := SafeCopyFromModbusREGsArray (data_in, data_out, row_in, column_in,
row_out, column_out, count);
```

Табл. 1.46– Параметры и возвращаемые значения функции SafeCopyFromModbusREGsArray

Имя параметра	Тип параметра	Описание
res	DINT	Результат выполнения преобразования: Если $\Rightarrow 0$ – число скопированных переменных типа <code>WORD</code> массива <code>data_in</code> , -1 – массив <code>data_in</code> не одномерный или <code>data_out</code> не двумерный, -2 – одно из значений номера строки/столбца находится за пределами границ массива
data_in	ModbusREGs	Массив данных типа <code>ModbusREGs</code> , из которого будет выполняться копирование
data_out	WORD	Двумерный массив данных типа <code>WORD</code> , в который будет выполняться копирование
row_in	DINT	Индекс строки в массиве <code>data_in</code> , начиная с которого будет выполняться копирование
column_in	DINT	Индекс столбца в массиве <code>data_in</code> , начиная с которого будет выполняться копирование
row_out	DINT	Индекс строки в массиве <code>data_out</code> , начиная с которого будут помещаться скопированные данные
column_out	DINT	Индекс столбца в массиве <code>data_out</code> , начиная с которого будут помещаться скопированные данные
count	DINT	Число копируемых регистров <code>WORD</code>

```
res:= SafeCopyToModbusREGsArray (data_in, data_out, row_in, column_in, row_out,
column_out, count);
```

Табл. 1.47– Параметры и возвращаемые значения функции SafeCopyToModbusREGsArray

Имя параметра	Тип параметра	Описание
res	DINT	Результат выполнения преобразования: Если => 0 – число скопированных переменных типа WORD массива data_in, -1 – массив data_in не двумерный или data_out не одномерный, -2 – одно из значений номера строки/столбца находится за пределами границ массива
data_in	WORD	Массив данных типа WORD, из которого будет выполняться копирование
data_out	ModbusREGs	Массив данных типа ModbusREGs, в который будет выполняться копирование
row_in	DINT	Первый индекс смещение в массиве data_in, начиная с которого будет выполняться копирование
column_in	DINT	Второй индекс смещение в массиве data_in, начиная с которого будет выполняться копирование.
row_out	DINT	Первый индекс смещение в массиве data_out, начиная с которого будут помещаться скопированные данные
column_out	DINT	Второй индекс смещение в массиве data_out, начиная с которого будут помещаться скопированные данные
count	DINT	Число копируемых регистров WORD

```
(* data_in_fmr - ModbusREGs [0..3], data_out_fmr - WORD [0..3,1..1024] *)
data_in_fmr[0].regs[1]:=1;
data_in_fmr[1].regs[1024]:=1025;
data_in_fmr[3].regs[1024]:=1027;
res_fmr:=SafeCopyFromModbusREGsArray(data_in_fmr,data_out_fmr,0,1,0,1,4096);
(* res_fmr = 4096, data_out_fmr[0,1] = 1, data_out_fmr[1,1024] = 1025,
data_out_fmr [3,1024] = 1027 *)
```

```
(* data_in_tmr - WORD [0..3,1..1024], data_out_tmr - ModbusREGs [0..3] *)
data_in_tmr [0,0]:=5;
data_in_tmr [1,1024]:=6;
data_in_tmr [3,1024]:=7;
res := SafeCopyToModbusREGsArray (data_in_tmr,data_out_tmr,1,2,0,1,2039);
(* res_tmr = 4096, data_out_tmr[0,1] = 5, data_out_tmr[1,1024] = 6,
data_out_tmr [3,1024] = 7 *)
```

Листинг 4.7 – Использование функций SafeCopyFromModbusREGsArray и SafeCopyToModbusREGsArray

4.8 Функции WriteReal2DimArray и ReadReal2DimArray

Функции `WriteReal2DimArray` и `ReadReal2DimArray` выполняют копирование данных из двумерного массива `REAL` в файл данных и копирование из файла данных в двумерный массив `REAL`.

Функции `WriteReal2DimArray` и `ReadReal2DimArray` выполняются параллельно и на основном, и на резервном CPU.

Параметры и возвращаемые значения функций `WriteReal2DimArray` и `ReadReal2DimArray` приведены в [Табл. 1.48](#) и [Табл. 1.49](#) соответственно, пример использования функций приведён в [Листинг 4.8](#).

```
res := WriteReal2DimArray(filename, rArray);
```

Табл. 1.48– Параметры и возвращаемые значения функции WriteReal2DimArray

Имя параметра	Тип параметра	Описание
res	DINT	Результат выполнения записи: 0 – успешно, -1 – недопустимое имя файла, -2 – ошибка открытия файла, -3 – ошибка записи данных в файл.
filename	STRING	Переменная типа <code>STRING</code> . Название файла, в который будет скопирован двумерный массив данных типа <code>REAL</code> .
rArray	REAL	Двумерный массив данных типа <code>REAL</code> , который будет скопирован в файл.

```
res := ReadReal2DimArray(RetainReal, rArray_out);
```

Табл. 1.49– Параметры и возвращаемые значения функции ReadReal2DimArray

Имя параметра	Тип параметра	Описание
res	DINT	Результат выполнения чтения: 0 – успешно, -1 – файла с таким именем нет в папке, -2 – ошибка открытия файла, -4 – ошибка чтения данных из файла, -5 - размер файла не совпадает с размером массива данных типа <code>REAL</code>
filename	STRING	Переменная типа <code>STRING</code> . Название файла, из которого будет производиться копирование в двумерный массив данных типа <code>REAL</code> .
rArray	REAL	Двумерный массив данных типа <code>REAL</code> , в который будут скопированы данные из файла.

```
(* rArray_wr - REAL [0..9,0..9], rArray_rr - REAL [0..9,0..9] *)
rArray_wr[0,0]:=10.0;
rArray_wr[4,9]:=20.0;
rArray_wr[9,9]:=30.0;
res_wr:=WriteReal2DimArray('RETAIN',rArray_wr);
res_rr:=ReadReal2DimArray('RETAIN',rArray_rr);

(* res_wr = 0, res_rr = 0, rArray_rr [0,0] = 10.0, rArray_rr [4,9] = 20.0,
rArray_rr [9,9] = 30.0 *)
```

Листинг 4.8 – Использование функций WriteReal2DimArray и ReadReal2DimArray

4.9 Функция CompareWordArrays

Функция `CompareWordArrays` выполняет сравнение значений двух массивов `WORD`.

Параметры и возвращаемые значения функции `CompareWordArrays` приведены в [Табл. 1.50](#), пример использования функции приведён в [Листинг 4.9](#).

```
res := CompareWordArrays(wArray_1, wArray_2, result);
```

Табл. 1.50– Параметры и возвращаемые значения функции CompareWordArrays

Имя параметра	Тип параметра	Описание
res	DINT	Число пар сравниваемых элементов массивов.
wArray_1	WORD	Первый сравниваемый массив данных типа WORD.
wArray_2	WORD	Второй сравниваемый массив данных типа WORD.
result	BOOL	Массив данных типа BOOL, в который записываются результаты сравнений двух массивов данных типа WORD.

Сравнение двух массивов данных начинается с первого элемента. Результат сравнения записывается в соответствующий элемент массива данных `result`, если элементы массивов данных типа `WORD` равны, в соответствующий элемент массива данных `result` запишется значение `TRUE`. Сравнивается минимально общее число элементов во всех трёх массивах.

```
(* wArray_1 - WORD [1..5], wArray_1 - WORD [20..40], result - BOOL [0..99] *)
wArray_1[1]:=1;
wArray_2[20]:=1;
wArray_1[2]:=2;
wArray_2[21]:=2;
wArray_1[3]:=3;
wArray_2[22]:=10;
wArray_1[4]:=4;
wArray_2[23]:=4;
wArray_1[5]:=5;
wArray_2[24]:=5;
res := CompareWordArrays(wArray_1, wArray_2, result);
(* res = 5, result[0] = TRUE, result[1] = TRUE, result[2] = FALSE, result[3] =
TRUE, result[4] = TRUE *)
```

Листинг 4.9 – Использование функции CompareWordArrays

4.10 Функция CRC16ForModbusRegs

Функция `CRC16ForModbusRegs` выполняет считывание контрольной суммы CRC16 для подмножества элементов типа `WORD` массива данных типа `ModbusREGs`.

Параметры и возвращаемые значения функции `CRC16ForModbusRegs` приведены в [Табл. 1.51](#), пример использования функции приведён в [Листинг 4.10](#).

```
res := CRC16ForModbusRegs (MB_TCP_sl4k_HR, row, column, count);
```

Табл. 1.51– Параметры и возвращаемые значения функции CRC16ForModbusRegs

Имя параметра	Тип параметра	Описание
res	UINT	Значение контрольной суммы CRC16. Если массив данных <code>data</code> не одномерный или индексы <code>row</code> или <code>column</code> лежат за пределами массива данных, функция вернет значение <code>0xffff</code>
data	ModbusREGs	Массив данных типа <code>ModbusREGs</code> , для подмножества элементов типа <code>WORD</code> которого будет рассчитана контрольная сумма CRC16
row	DINT	Первый индекс смещение в массиве <code>MB_TCP_sl4k_HR</code> , начиная с которого будет выполняться расчет контрольной суммы CRC16
column	DINT	Второй индекс смещение в массиве <code>MB_TCP_sl4k_HR</code> , начиная с которого будет выполняться расчет контрольной суммы CRC16
count	DINT	Число элементов типа <code>WORD</code> массива данных типа <code>ModbusREGs</code> для которых будет рассчитана контрольная сумма CRC16

```
(* MB_TCP_sl4k_HR - ModbusREGs [1..4] *)
res := CRC16ForModbusRegs (MB_TCP_sl4k_HR, 0, 0, 4096);
(* res = 65535 *)
```

Листинг 4.10 – Использование функции CRC16ForModbusRegs

4.11 Функция CRC16ForWords

Функция `CRC16ForWords` выполняет расчет контрольной суммы CRC16 для подмножества элементов массива данных типа `WORD`.

Параметры и возвращаемые значения функции `CRC16ForWords` приведены в [Табл. 1.52](#), пример использования функции приведен в [Листинг 4.11](#).

```
res := CRC16ForWords (wArray, count);
```

Табл. 1.52– Параметры и возвращаемые значения функции CRC16ForWords

Имя параметра	Тип параметра	Описание
<code>res</code>	<code>UINT</code>	Значение контрольной суммы CRC16. Если массив данных <code>wArray</code> не одномерный или нулевой размерности функция вернет значение <code>0xffff</code>
<code>wArray</code>	<code>WORD</code>	Массив данных типа <code>WORD</code> для <i>подмножества</i> элементов которого будет рассчитываться контрольная сумма CRC16
<code>count</code>	<code>DINT</code>	Число элементов массива данных типа <code>WORD</code> для которых будет рассчитана контрольная сумма CRC16

```
(* wArray - WORD [1..10] *)
res := CRC16ForWords(wArray, 5);
(* res = 1904 *)
```

Листинг 4.11 – Использование функции CRC16ForWords

4.12 Функция `InitRS485ModuleModbus`

Функция `InitRS485ModuleModbus` устарела и не рекомендуется к использованию.

С 13.05.2022 функция `InitRS485ModuleModbus` удалена из системного ПО модуля CPU.

4.13 Функция `SendMessage`

Функция `SendMessage` устарела и не рекомендуется к использованию.

4.14 Функция SetDateTime

Функция `SetDateTime` выполняет установку даты и времени для модуля CPU. Параметры даты и времени указываются и применяются без учёта часового пояса. Перед применением функции необходимо отключить NTP-сервер с помощью плагина МК500 IODevice ([Рис. 4.4](#)), иначе значения даты и времени скорректируются в соответствии с NTP-сервером.

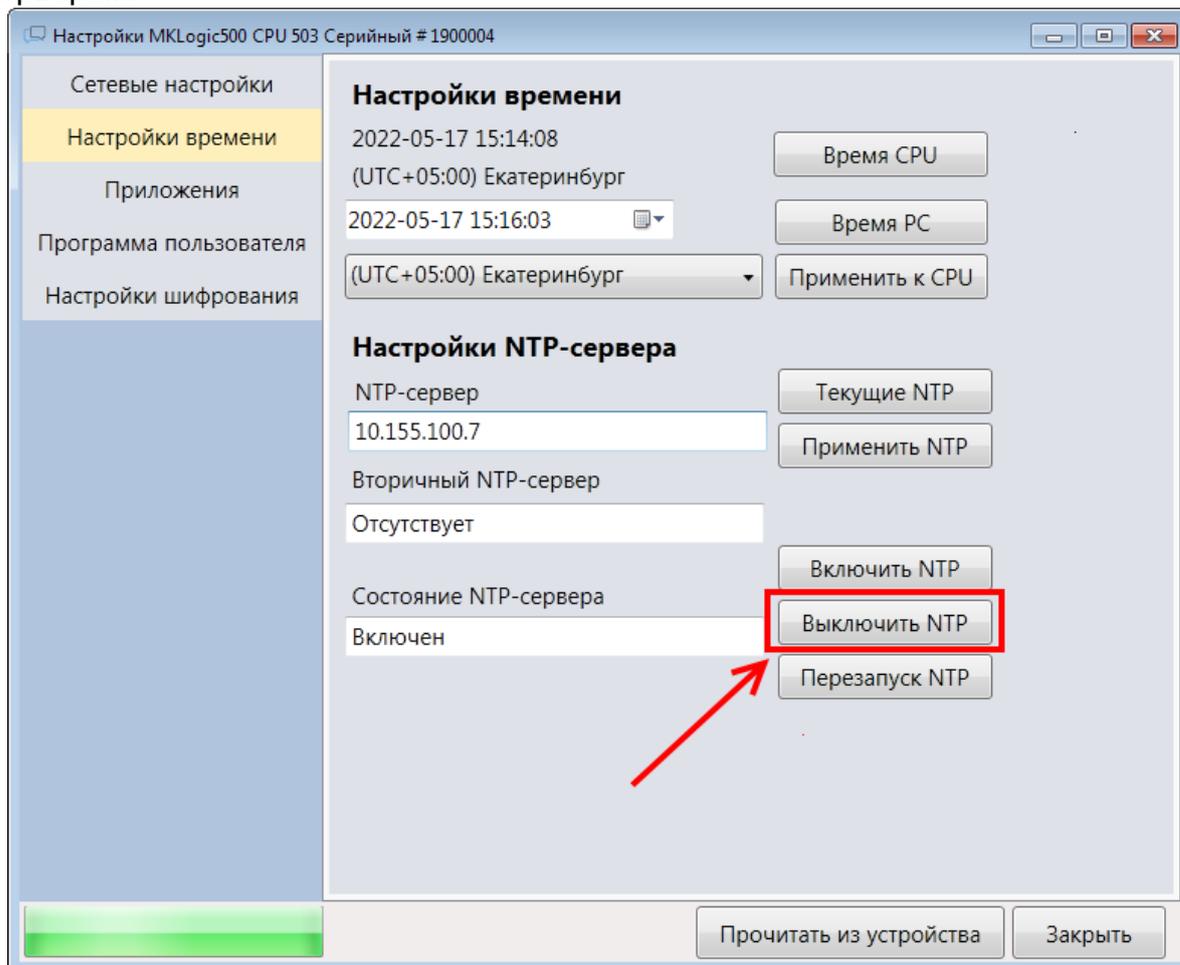


Рис. 4.4 – Окно настроек времени модуля CPU

Параметры и возвращаемые значения функции `SetDateTime` приведены в [Табл. 1.53](#), пример использования функции приведён в [Листинг 4.12](#).

```
res := SetDateTime (year, month, day, hour, minute, second);
```

Табл. 1.53– Параметры и возвращаемые значения функции SetDateTime

Имя параметра	Тип параметра	Описание
res	BOOL	Результат выполнения функции. TRUE - удалось применить время, FALSE - не удалось применить время, в том числе и по причине некорректно введённых входных значений
year	DINT	Год
month	DINT	Месяц

Имя параметра	Тип параметра	Описание
day	DINT	День
hour	DINT	Часы
minute	DINT	Минуты
second	DINT	Секунды

```
res := SetDateTime(2019, 1, 1, 0, 0, 0);  
(* res = TRUE *)
```

Листинг 4.12 – Использование функции SetDateTime

Для перевода формата времени IEC 60870-5-104 в аргументы функции для установки системного времени SetDateTime нужно прибавить к аргументу year значение 2000 и поделить на 1000 поле IEC104Time.msec (см. [Листинг 4.13](#)).

```
SetDateTime(syncTime.year+2000, syncTime.month, syncTime.monthday,  
            syncTime.hour, syncTime.minute, syncTime.msec / 1000);
```

Листинг 4.13 – Перевод формата времени IEC 60870-5-104 в системное

4.15 Функция UdpMessage

Функция `UdpMessage` выполняет отправку отладочных сообщений в специальную программу-приёмник сообщений `udp_debug` (на 16.12.2021 актуальная версия 1.0.3) из текста программы пользователя. Метка времени CPU автоматически добавляется к тексту сообщения.

Параметры и возвращаемые значения функции `UdpMessage` приведены в [Табл. 1.54](#), пример использования функции приведён в [Листинг 4.14](#).

```
res := UdpMessage(udp_address, message);
```

Табл. 1.54– Параметры и возвращаемые значения функции UdpMessage

Имя параметра	Тип параметра	Описание
res	BOOL	Результат выполнения функции. TRUE - удалось отправить сообщение, FALSE - не удалось отправить сообщение (некорректный UDP-адрес, либо ошибка функции отправки сообщения)
udp_address	STRING	UDP-адрес компьютера, на котором запущена программа <code>udp_debug</code> , принимающая сообщения
message	STRING	Текст сообщения

```
res := UdpMessage('10.155.27.1', 'error');
(* res = TRUE *)
```

Листинг 4.14 – Использование функции UdpMessage

4.16 Функция SwapActiveCPU

Функция `SwapActiveCPU` выполняет рестарт среды исполнения ISaGRAF для проектов с поддержкой Failover на модуле CPU в режиме «Running» при наличии рабочего модуля CPU в режиме «Standby». В результате выполнения этой функции меняются ролями модули CPU: резервный модуль CPU переходит в режим «Running», а перезапущенный модуль после старта среды исполнения переходит в режим «Standby».

Ограничением функции `SwapActiveCPU` является то, что на время перезапуска модуля CPU проект работает без резерва, что может быть недопустимо по условиям эксплуатации.

Параметры и возвращаемые значения функции `SwapActiveCPU` приведены в [Табл. 1.55](#), пример использования функции приведён в [Листинг 4.15](#).

(* Все проверки на допустимость вызова выполняются внутри функции *)

```
res := SwapActiveCPU();
```

Листинг 4.15 – Использование функции SwapActiveCPU

```
res := SwapActiveCPU();
```

Табл. 1.55– Параметры и возвращаемые значения функции SwapActiveCPU

Имя параметра	Тип параметра	Описание
res	INT	Результат выполнения: 1 – выполняется рестарт, 0 – команда проигнорирована, -1 или -2 – системная ошибка, безопасное выполнение рестарта невозможно

4.17 Функции GetBitFromDInt, GetBitFromDWord и GetBitFromWord

Функции `GetBitFromDInt`, `GetBitFromDWord` и `GetBitFromWord` **выполняют** извлечение бита из переменной типа `DINT`, `DWORD` и `WORD`.

Параметры и возвращаемые значения функций `GetBitFromDInt`, `GetBitFromDWord` и `GetBitFromWord` приведены в [Табл. 1.56](#), [Табл. 1.57](#) и [Табл. 1.58](#) соответственно, пример использования функций приведён в [Листинг 4.16](#).

```
ResBitDINT := GetBitFromDInt(InBitDINT, NoBitDINT);
```

Табл. 1.56– Параметры и возвращаемые значения функции `GetBitFromDInt`

Имя параметра	Тип параметра	Описание
<code>ResBitDINT</code>	<code>BOOL</code>	Извлечённый бит
<code>InBitDINT</code>	<code>DINT</code>	Переменная, из которой извлекается бит
<code>NoBitDINT</code>	<code>DINT</code>	Номер бита (от 0 до 31)

```
ResBitDWORD := GetBitFromDWord(InBitDWORD, NoBitDWORD);
```

Табл. 1.57– Параметры и возвращаемые значения функции `GetBitFromDWord`

Имя параметра	Тип параметра	Описание
<code>ResBitDWORD</code>	<code>BOOL</code>	Извлечённый бит
<code>InBitDWORD</code>	<code>DWORD</code>	Переменная, из которой извлекается бит
<code>NoBitDWORD</code>	<code>DINT</code>	Номер бита (от 0 до 31)

```
ResBitWORD := GetBitFromWord(InBitWORD, NoBitWORD);
```

Табл. 1.58– Параметры и возвращаемые значения функции `GetBitFromWord`

Имя параметра	Тип параметра	Описание
<code>ResBitWORD</code>	<code>BOOL</code>	Извлечённый бит
<code>InBitWORD</code>	<code>WORD</code>	Переменная, из которой извлекается бит
<code>NoBitWORD</code>	<code>DINT</code>	Номер бита (от 0 до 15)

```
(* InBitDINT - DINT, NoBitDINT - DINT *)
InBitDINT:= 1081410080;
NoBitDINT:=5;
ResBitDINT := GetBitFromDInt(InBitDINT,NoBitDINT);
(* ResBitDINT = 1 *)

(* InBitDWORD - DWORD, NoBitDWORD - DINT *)
InBitDWORD:= 1081410080;
NoBitDWORD:=5;
ResBitDWORD:=GetBitFromDWord(InBitDWORD,NoBitDWORD);
(* ResBitDWORD = 1 *)

(* InBitWORD - WORD, NoBitWORD - DINT *)
InBitWORD:= 33312;
NoBitWORD:=5;
ResBitWORD:=GetBitFromWord(InBitWORD,NoBitWORD);
(* ResBitWORD = 1 *)
```

Листинг 4.16 – Использование функций GetBitFromDInt, GetBitFromDWord и GetBitFromWord

4.18 Функции GetByteFromDWord и GetByteFromWord

Функции `GetByteFromDWord` и `GetByteFromWord` выполняют извлечение байта из переменной типа `DWORD` и `WORD`.

Параметры и возвращаемые значения функций `GetByteFromDWord` и `GetByteFromWord` приведены в [Табл. 1.59](#) и [Табл. 1.60](#) соответственно, пример использования функций приведён в [Листинг 4.17](#).

```
ResByteDWord:=GetByteFromDWord(InByteDWord,NoByteDWord);
```

Табл. 1.59– Параметры и возвращаемые значения функции GetByteFromDWord

Имя параметра	Тип параметра	Описание
ResByteDWord	BYTE	Извлечённый байт
InByteDWord	DWORD	Переменная, из которой извлекается байт
NoByteDWord	DINT	Номер байта (от 0 до 3)

```
ResByteWord:=GetByteFromWord(InByteWord,NoByteWord);
```

Табл. 1.60– Параметры и возвращаемые значения функции GetByteFromWord

Имя параметра	Тип параметра	Описание
ResByteDWord	BYTE	Извлечённый байт
InByteDWord	WORD	Переменная, из которой извлекается байт
NoByteDWord	DINT	Номер байта (от 0 до 1)

```
(* InByteDWord - DWORD, NoByteDWord - DINT *)
InByteDWord:= 11141120;
NoByteDWord:=2;
ResByteDWord:=GetByteFromDWord(InByteDWord,NoByteDWord);
(* ResByteDWord = 170 *)
```

```
(* InByteWord - WORD, NoByteWord - DINT *)
InByteWord:= 43520;
NoByteWord:=1;
ResByteWord:=GetByteFromWord(InByteWord,NoByteWord);
(* ResByteWord = 170 *)
```

Листинг 4.17 – Использование функций GetByteFromDWord и GetByteFromWord

4.19 Функция GetWordFromDWord

Функция `GetWordFromDWord` выполняет извлечение слова из переменной типа `DWORD`.

Параметры и возвращаемые значения функции `GetWordFromDWord` приведены в [Табл. 1.61](#), пример использования функций приведён в [Листинг 4.18](#).

```
ResWordDWORD:=GetWordFromDWord(InWordDWORD,NoWordDWORD);
```

Табл. 1.61– Параметры и возвращаемые значения функции GetWordFromDWord

Имя параметра	Тип параметра	Описание
ResWordDWORD	WORD	Извлеченное слово
InWordDWORD	DWORD	Переменная, из которой извлекается слово
NoWordDWORD	DINT	Номер слова (от 0 до 1)

```
(* InWordDWORD - DWORD, NoWordDWORD - DINT *)
InWordDWORD:= 2863267840;
NoWordDWORD:=1;
ResWordDWORD:=GetWordFromDWord(InWordDWORD,NoWordDWORD);
(* ResWordDWORD = 43690 *)
```

Листинг 4.18 – Использование функции GetWordFromDWord

4.20 Функции SetBitToDInt, SetBitToDWord и SetBitToWord

Функции `SetBitToDInt`, `SetBitToDWord` и `SetBitToWord` выполняют замену указанного бита в переменной типа `DINT`, `DWORD` и `WORD`.

Параметры и возвращаемые значения функций `SetBitToDInt`, `SetBitToDWord` и `SetBitToWord` приведены в [Табл. 1.62](#), [Табл. 1.63](#) и [Табл. 1.64](#) соответственно, пример использования функций приведён в [Листинг 4.19](#).

```
SResBitDINT:=SetBitToDInt(InSBitDINT,NoSBitDINT,SrcSBitDINT);
```

Табл. 1.62– Параметры и возвращаемые значения функции SetBitToDInt

Имя параметра	Тип параметра	Описание
SResBitDINT	DINT	Измененная переменная с заменой указанного бита, либо исходная переменная при некорректном выборе бита
InSBitDINT	BOOL	Значение бита
NoSBitDINT	DINT	Номер бита (от 0 до 31)
SrcSBitDINT	DINT	Исходная переменная

```
SResBitDWORD:=SetBitToDWord(InSBitDWORD,NoSBitDWORD,SrcSBitDWORD);
```

Табл. 1.63– Параметры и возвращаемые значения функции SetBitToDWord

Имя параметра	Тип параметра	Описание
SResBitDWORD	DWORD	Измененная переменная с заменой указанного бита, либо исходная переменная при некорректном выборе бита
InSBitDWORD	BOOL	Значение бита
NoSBitDWORD	DINT	Номер бита (от 0 до 31)
SrcSBitDWORD	DWORD	Исходная переменная

```
SResBitWORD:=SetBitToWord(InSBitWORD,NoSBitWORD,SrcSBitWORD);
```

Табл. 1.64– Параметры и возвращаемые значения функции SetBitToWord

Имя параметра	Тип параметра	Описание
SResBitWORD	WORD	Измененная переменная с заменой указанного бита, либо исходная переменная при некорректном выборе бита
InSBitWORD	BOOL	Значение бита
NoSBitWORD	DINT	Номер бита (от 0 до 15)
SrcSBitWORD	WORD	Исходная переменная

```
(* InSBitDINT - BOOL, NoSBitDINT - DINT, SrcSBitDINT - DINT *)
InSBitDINT:= TRUE;
NoSBitDINT:=8;
SrcSBitDINT:=65279;
SResBitDINT:=SetBitToDInt (InSBitDINT,NoSBitDINT,SrcSBitDINT);
(* SResBitDINT = 65535*)

(* InSBitDWORD - BOOL, NoSBitDWORD - DINT, SrcSBitDWORD - DWORD *)
InSBitDWORD:= TRUE;
NoSBitDWORD:=8;
SrcSBitDWORD:=65279;
SResBitDWORD:=SetBitToDWORD (InSBitDWORD,NoSBitDWORD,SrcSBitDWORD);
(* SResBitDWORD = 65535 *)

(* InSBitWORD - BOOL, NoSBitWORD - DINT, SrcSBitWORD - WORD *)
InSBitWORD:= TRUE;
NoSBitWORD:=8;
SrcSBitWORD:=65279;
SResBitWORD:=SetBitToWORD (InSBitWORD,NoSBitWORD,SrcSBitWORD);
(* SResBitWORD = 65535 *)
```

Листинг 4.19 – Использование функций SetBitToDInt, SetBitToDWord и SetBitToWord

4.21 Функции SetByteToDWord и SetByteToWord

Функции SetByteToDWord и SetByteToWord выполняют замену указанного байта в переменной типа DWORD и WORD.

Параметры и возвращаемые значения функций SetByteToDWord и SetByteToWord приведены в [Табл. 1.65](#) и [Табл. 1.66](#) соответственно, пример использования функций приведён в [Листинг 4.20](#).

```
SResByteDWord:=SetByteToDWord(InByteDWord,NoByteDWord,SrcByteDWord);
```

Табл. 1.65– Параметры и возвращаемые значения функции SetByteToDWord

Имя параметра	Тип параметра	Описание
SResByteDWord	DWORD	Измененная переменная с заменой указанного байта, либо исходная переменная при некорректном выборе байта
InByteDWord	BYTE	Значение байта
NoByteDWord	DINT	Номер байта (от 0 до 3)
SrcByteDWord	DWORD	Исходная переменная

```
SResByteWord:=SetByteToWord(InByteWord,NoByteWord,SrcByteWord);
```

Табл. 1.66– Параметры и возвращаемые значения функции SetByteToWord

Имя параметра	Тип параметра	Описание
SResByteWord	WORD	Измененная переменная с заменой указанного байта, либо исходная переменная при некорректном выборе байта
InByteWord	BYTE	Значение байта
NoByteWord	DINT	Номер байта (от 0 до 1)
SrcByteWord	WORD	Исходная переменная

```
(* InByteDWord - BYTE, NoByteDWord - DINT, SrcByteDWord - DWORD *)
InByteDWord:= 225;
NoByteDWord:=1;
SrcByteDWord:=65535;
SResByteDWord:=SetByteToDWord(InByteDWord,NoByteDWord,SrcByteDWord);
(* SResByteDWord = 57855 *)
```

```
(* InByteWord - BYTE, NoByteWord - DINT, SrcByteWord - WORD *)
InByteWord:= 225;
NoByteWord:=1;
SrcByteWord:=65535;
SResByteWord:=SetByteToWord(InByteWord,NoByteWord,SrcByteWord);
(* SResByteWord = 57855 *)
```

Листинг 4.20 – Использование функций SetByteToDWord и SetByteToWord

4.22 Функция SetWordToDWord

Функция SetWordToDWord выполняют замену указанного слова в переменной типа DWORD.

Параметры и возвращаемые значения функции SetWordToDWord приведены в [Табл. 1.67](#), пример использования функции приведён в [Листинг 4.21](#).

```
SResWordDWord:=SetWordToDWord(InSWordDWord,NoSWordDWord,SrcSWordDWord);
```

Табл. 1.67– Параметры и возвращаемые значения функции SetWordToDWord

Имя параметра	Тип параметра	Описание
SResWordDWord	DWORD	Изменённая переменная с заменой указанного слова, либо исходная переменная при некорректном выборе слова
InSWordDWord	WORD	Значение слова
NoSWordDWord	DINT	Номер слова (от 0 до 1)
SrcSWordDWord	DWORD	Исходная переменная

```
(* InSWordDWord - WORD, NoSWordDWord - DINT, SrcSWordDWord - DWORD *)
InSWordDWord:= 43690;
NoSWordDWord:=1;
SrcSWordDWord:= 4294967295;
SResWordDWord:=SetWordToDWord(InSWordDWord,NoSWordDWord,SrcSWordDWord);
(* SResWordDWord = 2863333375 *)
```

Листинг 4.21 – Использование функции SetWordToDWord

4.23 Функция SynchronizeFtpFiles

Функция `SynchronizeFtpFiles` используется для синхронизации пользовательских данных в папке FTP. Функция `SynchronizeFtpFiles` выполняет копирование пользовательских данных из папки FTP на CPU в режиме «Running» и размещает их в папке FTP на CPU в режиме «Standby».

Не будут синхронизированы следующие файлы и папки:

- системные папки (`log`, `result` и `RETAIN`) и их содержимое;
- `tdb`-файлы;
- файлы обновлений (`*.enc` и `*.sign`);
- файлы логов обновлений (вида `logpatch-50x-datetime-SN=xxxxxx.txt`);

Параметры и возвращаемые значения функции `SynchronizeFtpFiles` приведены в [Табл. 1.68](#), пример использования функции приведён в [Листинг 4.22](#).

```
res := SynchronizeFtpFiles();
```

Табл. 1.68– Параметры и возвращаемые значения функции SynchronizeFtpFiles

Имя параметра	Тип параметра	Описание
<code>res</code>	BOOL	Результат выполнения: всегда TRUE

```
res := SynchronizeFtpFiles();
```

Листинг 4.22 – Использование функции SynchronizeFtpFiles

4.24 Функция PredictByDerivative

Функция `PredictByDerivative` выполняет прогнозирование достижения определённого значения в течении заданного времени.

Параметры и возвращаемые значения функции `PredictByDerivative` приведены в [Табл. 1.69](#), пример использования функции приведён в [Листинг 4.23](#).

```
res:=PredictByDerivative(curVal,prevVal,vSet,tSet,direct,deltaT,bCalc);
```

Табл. 1.69– Параметры и возвращаемые значения функции PredictByDerivative

Имя параметра	Тип параметра	Описание
res	INT	Результат выполнения: 0 – значение не будет достигнуто, 1 – значение будет достигнуто, -1 – неверно указано одно из значений времени (значение меньше либо равно нулю), -2 – значение направления не совпадает со знаком отношения текущего и ожидаемого значения (например, <code>curVal > vSet</code> и <code>direct = TRUE</code>), -3 – параметр <code>bCalc</code> равен <code>FALSE</code>
curVal	REAL	Текущее значение
prevVal	REAL	Предыдущее значение
vSet	REAL	Ожидаемое значение
tSet	REAL	Ожидаемое время для достижения ожидаемого значения
direct	BOOL	Направление ожидаемого значения, <code>TRUE</code> – значение вверх, <code>FALSE</code> – значение вниз
deltaT	REAL	Время, через которое будут происходить замеры текущего значения
bCalc	BOOL	Вычислять, будет ли достигнуто ожидаемое значение за определённое время (<code>TRUE</code>) или нет (<code>FALSE</code>)

```
(* curVal - REAL, prevVal - REAL, vSet - REAL, tSet - REAL, direct - BOOL,
deltaT - REAL, bCalc - BOOL *)
curVal := 10.0;
prevVal := 9.0;
vSet := 200.0;
tSet := 300.0;
direct := TRUE;
deltaT := 1;
bCalc := TRUE;
res := PredictByDerivative(curVal,prevVal,vSet,tSet,direct,deltaT,bCalc);
(* res = 1 *)
```

Листинг 4.23 – Использование функции PredictByDerivative

4.25 Функция GetDevInfo

Функция `GetDevInfo` выполняет поиск устройства с указанными значениями `openPowerLink ID`, номера стойки и позиции модуля, и копирует диагностические данные этого устройства в передаваемую переменную типа `IOModuleDiag`.

Параметры и возвращаемые значения функции `GetDevInfo` приведены в [Табл. 1.70](#), пример использования функции приведён в [Листинг 4.24](#).

```
res:=GetDevInfo(opwl_id, rack, slot, IOModulDiagIn);
```

Табл. 1.70– Параметры и возвращаемые значения функции GetDevInfo

Имя параметра	Тип параметра	Описание
res	INT	Результат выполнения: 0 – не удалось прочитать данные, с указанными параметрами, 1 – удалось прочитать данные
opwl_id	WORD	openPowerLink ID модуля
rack	WORD	Номер стойки
slot	WORD	Позиция модуля в стойке
IOModulDiagIn	IOModuleDiag	Структура куда будет копироваться результат (переменная-структура подробно описана в п. 3.1.6). Поля структуры будут иметь нулевое значение, если не удалось найти модуль с указанными параметрами

```
(* opwl_id - WORD, rack - WORD, slot - WORD, IOModulDiagIn - IOModuleDiag *)
opwl_id := 240;
rack := 1;
slot := 1;
res:=GetDevInfo(opwl_id, rack, slot, IOModulDiagIn)
(* res = 1 *)
```

Листинг 4.24 – Использование функции GetDevInfo

4.26 Функция GetFastModuleState

Функция `GetFastModuleState` выполняет сбор диагностики по всем модулям в конфигурации проекта (в порядке возрастания значений `Index` модулей, см. [п. 3.1.2](#)) и копирует эти данные в передаваемый массив. Элементы передаваемого массива должны иметь тип `FastModuleState`.

Следует обратить внимание, что если входной массив имеет размерность меньше чем количество модулей в стойке, то диагностика будет собрана не полностью; если элементов в массиве больше, то оставшиеся значения массива будут нулевыми.

Параметры и возвращаемые значения функции `GetFastModuleState` приведены в [Табл. 1.71](#), пример использования функции приведён в [Листинг 4.25](#).

```
res := GetFastModuleState(resArr);
```

Табл. 1.71– Параметры и возвращаемые значения функции GetFastModuleState

Имя параметра	Тип параметра	Описание
res	INT	Количество модулей, для которых были почитаны диагностические данные
resArr	FastModuleState[a..b]	Массив структур, куда будет копироваться результат. Каждый элемент массива содержит краткую диагностику по модулю. Поля структуры: pwl_id – идентификатор для OpenPowerlink, rack – номер стойки, slot – позиция модуля в стойке, state – статус диагностики: -1 – не соответствует, 0 – отсутствует, 1 – соответствует, 2 – виртуальный

```
(* resArr – FastModuleState[0..2] *)
res := GetFastModuleState(resArr);
(* res = 3 *)
```

Листинг 4.25 – Использование функции GetFastModuleState

4.27 Функция FTPEnable

Функция FTPEnable выполняет запуск либо остановку встроенного в модуль CPU FTP-сервера.

Работа функции FTPEnable имеет две особенности, которые следует учитывать:

1. Остановка и запуск FTP-сервера выполняется в течение определённого времени, обычно 1-2 секунды. Текущее состояние FTP-сервера следует контролировать через переменную Services.FTPStatus устройства system_info ([п. 3.7.6](#)).
2. Остановка и запуск FTP-сервера выполняется на обоих CPU резервированной пары, поэтому функция FTPEnable всегда возвращает TRUE.

Параметры и возвращаемые значения функции FTPEnable приведены в [Табл. 1.72](#), пример использования функции приведён в [Листинг 4.26](#).

```
res := FTPEnable(cmd);
```

Табл. 1.72– Параметры и возвращаемые значения функции FTPEnable

Имя параметра	Тип параметра	Описание
res	BOOL	Результат выполнения функции: всегда TRUE.
cmd	BOOL	TRUE – команда запустить FTP-сервер; FALSE – команда остановить FTP-сервер.

```
(* system_info.Services.FTPStatus = 1 *)
res := FTPEnable(FALSE);
(* res = TRUE, system_info.Services.FTPStatus = -2 *)

(* через 1-2 секунды: system_info.Services.FTPStatus = 0 *)

res := FTPEnable(FALSE);
(* res = TRUE, system_info.Services.FTPStatus = 0 *)
res := FTPEnable(TRUE);
(* res = TRUE, system_info.Services.FTPStatus = -1 *)

(* через 1-2 секунды: system_info.Services.FTPStatus = 1 *)
```

Листинг 4.26 – Использование функции FTPEnable

4.28 Функция UpdateOutputData

Функция UpdateOutputData предназначена для немедленного обновления данных всех выходных простых устройств в составе определённого модуля.

Параметры и возвращаемые значения функции UpdateOutputData приведены в [Табл. 1.73](#), пример использования функции приведён в [Листинг 4.27](#).

```
res:= UpdateOutputData(opwl_id, rack, slot);
```

Табл. 1.73– Параметры и возвращаемые значения функции UpdateOutputData

Имя параметра	Тип параметра	Описание
res	BOOL	Результат выполнения функции: всегда TRUE
pwl_id	WORD	openPowerLink ID модуля
rack	WORD	Номер стойки
slot	WORD	Позиция модуля в стойке

```
res := UpdateOutputData(240,1,6);
```

Листинг 4.27 – Использование функции UpdateOutputData



ВНИМАНИЕ

Частый вызов функции UpdateOutputData приведёт к ухудшению производительности и превышению времени цикла.

4.29 Функция ConvertToIec104Time

Функция `ConvertToIec104Time` предназначена для преобразования структуры `NowTime` в `IEC104Time`.

Параметры и возвращаемые значения функции `ConvertToIec104Time` приведены в [Табл. 1.74](#), пример использования функции приведён в [Листинг 4.28](#).

```
res := ConvertToIec104Time(currentTime, iecTimestamp);
```

Табл. 1.74– Параметры и возвращаемые значения функции ConvertToIec104Time

Имя параметра	Тип параметра	Описание
res	BOOL	Если преобразование успешно, то TRUE, иначе FALSE
convertTime	NowTime	Время для преобразования
returnTime	IEC104Time	Преобразованное время

```
res := ConvertToIec104Time(currentTime, iecTimestamp);
```

Листинг 4.28 – Использование функции ConvertToIec104Time

4.30 Функции WriteLogToFile и DeleteLogFile

Функция WriteLogToFile предназначена для записи сообщений в пользовательский файл журнала. Пользовательский файл журнала помещается в папку FTP/log. Общий размер пользовательских файлов в папке FTP/log ограничен 50 МБ, проверка выполняется в ходе вызова функции.

Функция WriteLogToFile выполняется параллельно и на основном, и на резервном CPU.

Параметры и возвращаемые значения функции WriteLogToFile приведены в [Табл. 1.74](#), пример использования функции приведён в [Листинг 4.28](#).

```
res := WriteLogToFile(filename, timestamp, message);
```

Табл. 1.75– Параметры и возвращаемые значения функции WriteLogToFile

Имя параметра	Тип параметра	Описание
Res	BOOL	Всегда TRUE
filename	STRING	Имя файла
timestamp	STRING	Текстовое представление метки времени
message	STRING	Текстовое сообщение

```
res := WriteLogToFile('journal.txt', '2022-06-07 10:12:74.123', 'Pump start');
```

Листинг 4.29 – Использование функции WriteLogToFile

Функция DeleteLogFile предназначена для удаления указанного пользовательского файла журнала.

Функция DeleteLogFile выполняется параллельно и на основном, и на резервном CPU.

Параметры и возвращаемые значения функции DeleteLogFile приведены в [Табл. 1.74](#), пример использования функции приведён в [Листинг 4.30](#).

```
res := DeleteLogFile(filename);
```

Табл. 1.76– Параметры и возвращаемые значения функции DeleteLogFile

Имя параметра	Тип параметра	Описание
Res	BOOL	Всегда TRUE
filename	STRING	Имя файла

```
res := DeleteLogFile('journal.txt');
```

Листинг 4.30 – Использование функции DeleteLogFile

Глава 5 Рекомендации по работе с модулями CPU

5.1 Оптимизация времени выполнения программ пользователя

5.1.1 Уменьшение числа параметров функций и функциональных блоков

Каждый входной параметр функции/функционального блока требует дополнительного времени на обработку. Уменьшение количества входных переменных приводит к ускорению работы функции/функционального блока.

5.1.2 Влияние параметра ресурса «Function Internal State Enabled»

Если параметр ресурса «Function Internal State Enabled» ([п. 2.5.2](#)) установлен в FALSE, то при каждом вызове функции все внутренние переменные будут принудительно устанавливаться в начальные значения (в 0 или FALSE если начального значения нет). В этом случае уменьшение количества внутренних переменных ведёт к ускорению работы функции

Если параметр ресурса «Function Internal State Enabled» установлен в TRUE, то при каждом вызове функции все внутренние переменные будут сохранять своё значение с прошлого вызова. В этом случае число внутренних переменных не влияет на время выполнения функции.

5.1.3 Работа с сохраняемыми (RETAIN) переменными

При работе с сохраняемыми (RETAIN) переменными в составе структур следует отмечать как Retained только те переменные, которые реально необходимо сохранять. Запись сохраняемых переменных требует определённого времени, добавление лишних переменных в RETAIN снижает производительность.

При обновлении программы пользователя RETAIN-файлы обновляются в случае, если в программе пользователя изменялось число или типы RETAIN-переменных.

5.1.4 Использование оператора WHILE вместо FOR

Внутренняя реализация цикла WHILE имеет в 2,5 раза меньше накладных расходов, чем реализация цикла FOR. Это ошибка в реализации среды исполнения ISaGRAF, и в последующих версиях она будет исправлена.

В среде исполнения ISaGRAF 6.5 рекомендуется использовать цикл WHILE вместо FOR, особенно для циклов с большой вложенностью либо в циклах с большим числом итераций ([Листинг 5.1](#)).

При использовании циклов WHILE следует соблюдать аккуратность, так как среда исполнения ISaGRAF не определяет заикливание внутри WHILE при неправильной работе с условием и не прерывает его работу, что приводит к зависанию среды исполнения и перегрузке модуля CPU.

```
(* 65 мс *)
FOR i := 1 TO 1000 DO
  FOR j := 1 TO 40 DO
    b := ANY_TO_REAL(i);
    c := ANY_TO_REAL(j);
    a := (1.1*b + c)*c/b*2.22;
  END_FOR;
END_FOR;
```

```
(* 45 мс *)
i := 0;
WHILE i < 1000 DO
  i := i+1;
  j := 0;
  WHILE j < 40 DO
    j := j+1;
    b := ANY_TO_REAL(i);
    c := ANY_TO_REAL(j);
    a := (1.1*b + c)*c/b*2.22;
  END_WHILE;
END_WHILE;
```

Листинг 5.1 – Использование оператора WHILE вместо FOR

5.1.5 Использование промежуточных переменных

Затраты времени на доступ к вложенным переменным растут пропорционально степени вложенности. Поэтому при циклической обработке вложенных массивов целесообразно использовать вспомогательные переменные вместо прямого обращения к переменной с использованием нескольких индексов ([Листинг 5.2](#)).

 **ВНИМАНИЕ**

Если вложенным в массив элементом является массив либо структура большого размера, затраты на копирование в промежуточную переменную начинают превышать экономию от её использования. Данный совет следует применять осторожно.

```
(* regs_2d - массив переменных типа ModbusREGs
   regs - промежуточная переменная типа Words1k
   ModbusREGs - тип-структура из одной переменной regs типа Words1k
   Words1k - тип-массив из 1024 WORD *)
(* 145 мс *)
i := 1;
WHILE i <= 100 DO
  i := i+1;
  j := 1;
  WHILE j <= 1024 DO
    j := j+1;
    regs_2d[i].regs[j] := ANY_TO_WORD(i+j);
  END_WHILE;
END_WHILE;
(* 96 мс *)
i := 1;
WHILE i <= 100 DO
  i := i+1;
  j := 1;
  regs := regs_2d[i].regs;
  WHILE j <= 1024 DO
    j := j+1;
    regs[j] := ANY_TO_WORD(i+j);
  END_WHILE;
  regs_2d[i].regs := regs;
END_WHILE;
```

Листинг 5.2 – Использование промежуточных переменных

5.1.6 Выход за пределы массива

Если в настройках ресурса не включена опция «Check array index» (см. [п. 2.5.2](#)), среда исполнения не контролирует выход за пределы массива. Следует быть особенно внимательными при работе с циклами While (см. [п.5.1.4](#)).

Выход за пределы массива приводит к непредсказуемому поведению программы пользователя, вплоть до её зависания. Следует избегать подобных ситуаций.

5.2 Правила написания программ для модулей CPU с поддержкой режима Failover

Работа программы пользователя в системах с поддержкой режима Failover имеет особенности, описанные в документе «Механизм обеспечения отказоустойчивости» (fvr_ISa6_ru).

Из этого документа следует, что несовпадение контрольных сумм оперативной памяти программы пользователя после окончания цикла выполнения на основном и на резервном модулях CPU влечёт выполнение синхронизации всей оперативной памяти в направлении от основного модуля CPU к резервному. Данная операция может занимать от нескольких десятков до нескольких сотен миллисекунд, почти наверняка приводит к нарушению заданного времени цикла и всегда приводит к повышению нагрузки на процессор модуля CPU.

Следовательно, программы пользователя для систем с поддержкой режима Failover должны быть написаны так, чтобы минимизировать число синхронизаций. В идеале должна выполняться единственная операция синхронизации, сразу после запуска.

Ниже перечислены рекомендации, нарушение которых приводит к незапланированному выполнению синхронизаций:

1. Обязательно привязывать все каналы ввода-вывода всех устройств ввода-вывода к пользовательским переменным. Наличие даже одного непривязанного канала ввода-вывода может привести к постоянным вызовам синхронизации при работе программы в системе с поддержкой Failover.
2. Избегать использования в программе (присвоения локальным переменным, сравнение и т.п.) системных переменных, их значения могут отличаться на различных модулях CPU.
3. Избегать использования функций и функциональных блоков, получающих данные не через каналы ввода-вывода, так как высока вероятность получения разных данных разными модулями CPU.
4. Избегать использования текущего времени в программе, например, полученного из функции `getCurrentTime`.
5. Избегать использования функциональных блоков для SFC-программ.

5.2.1 Диагностика проблем при работе процессорных модулей с поддержкой режима Failover

Диагностика программы пользователя в системах с поддержкой режима Failover описана в документе «Механизм обеспечения отказоустойчивости» (fvr_ISa6_ru).

Диагностировать наличие постоянной синхронизации пользовательских данных удобнее всего по значению системной переменной `__SYSVA_FO_DATASYNCCNT` (как это сделать, описано в [п. 2.8.2](#)). Её значение должно быть равно 1, и не должно увеличиваться. Постоянный рост значения этой переменной говорит о наличие проблемы, описанной в [п. 5.2](#).

Также следует обратить внимание на значение системной переменной `__SYSVA_FO_DATASYNCTIME`. Её значение в ходе работы зависит от числа используемых переменных ввода-вывода и может достигать десятков миллисекунд при работе с большими устройствами Modbus или IEC-104. Но её значение также резко возрастает при синхронизации пользовательских данных и может служить признаком проблем, описанных в [п. 5.2](#).

5.3 Диагностика среды выполнения и программы пользователя

5.3.1 Журнал сообщений программы пользователя

При возникновении важных событий/ошибок в программе пользователя среда исполнения ISaGRAF либо системное ПО модуля CPU помещает их в журнал ([п. 2.3.2.5](#)). В [Табл. 1.77](#) приводится их текст и расшифровка (коды и параметры сообщений – в шестнадцатеричной системе счисления).

Табл. 1.77– Текст и расшифровка сообщений журнала программы пользователя

Код сообщ.	Параметр сообщения	Текст сообщения	Расшифровка
1	<error code>	Startup error.	Ошибка при запуске ISaGRAF, код ошибки <error code>. Сообщение с точным кодом и текстом ошибки обычно предшествует этому сообщению
3	<error code>	Resource Restore error.	Ошибка восстановления сохранённого ресурса (обычно при старте), код ошибки <error code>. Сообщение с точным кодом и текстом ошибки обычно предшествует этому сообщению
4	0	Kernel Retain: Init error.	Ошибка инициализации RETAIN-переменных при старте ресурса. Может быть вызвано несовпадением файла RETAIN-переменных и программы пользователя
6	0	Kernel Retain: CRC error	Ошибка контрольной суммы при чтении файла RETAIN-переменных
7	0	Kernel Retain: Read error	Ошибка чтения файла RETAIN-переменных
8	0	Kernel Retain: Write error	Ошибка записи файла RETAIN-переменных
A	1	Resource Start Report	Ресурс запущен
B	1	Resource Stop Report	Ресурс остановлен
C	<function number>	Standard function not implemented	Программа пользователя не может найти и вызвать стандартную функцию. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
D	<function block number>	Standard function block instance init not implemented.	Программа пользователя не может найти функцию инициализации экземпляра стандартного функционального блока. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
E	<function block number>	Standard function block instance exit not implemented.	Программа пользователя не может найти функцию завершения экземпляра стандартного функционального блока. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF

Код сообщ.	Параметр сообщения	Текст сообщения	Расшифровка
F	<function block number>	Standard function block call not implemented.	Программа пользователя не может найти рабочую функцию экземпляра стандартного функционального блока. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
10	<function number>	Function not implemented	Программа пользователя не может найти и вызвать пользовательскую функцию. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
11	<function block number>	Function block instance init required but not implemented.	Программа пользователя не может найти функцию инициализации экземпляра пользовательского функционального блока. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
12	<function block number>	Function block instance exit required but not implemented	Программа пользователя не может найти функцию завершения экземпляра пользовательского функционального блока. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
13	<function block number>	Function block call not implemented	Программа пользователя не может найти рабочую функцию экземпляра пользовательского функционального блока. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
14	<function number>	Function not implemented.	Программа пользователя не может найти и вызвать пользовательскую функцию преобразования данных канала ввода-вывода. Может быть вызвано несоответствием tdb-файла программы пользователя и версией ISaGRAF
15	0	Kernel IOs: Init error	Ошибка инициализации пользовательского драйвера
16	<driver number>	Kernel IOs: Driver Init/Exit fct(s) not found.	Не найдена функция инициализации либо деинициализации пользовательского драйвера
17	<device number>	Kernel IOs: Device Open/Close fct(s) not found	Не найдена функция открытия либо закрытия устройства пользовательского драйвера
18	<device number>	Kernel IOs: Device Read fct not found.	Не найдена функция чтения устройства пользовательского драйвера
19	<device number>	Kernel IOs: Device Write fct not found	Не найдена функция записи устройства пользовательского драйвера
1A	<device number>	Kernel IOs: Device Control fct not	Не найдена функция управления устройством пользовательского драйвера

Код сообщ.	Параметр сообщения	Текст сообщения	Расшифровка
		found	
1B	<driver number>	Kernel IOs: Driver Init fct failure	Ошибка при вызове функции инициализации пользовательского драйвера
1C	<device number>	Kernel IOs: Device Open fct failure	Ошибка при вызове функции открытия устройства пользовательского драйвера. Может возникать, если в программе пользователя выбрано не поддерживаемое устройство (например, cpu502 для модуля CPU МК-501-022)
1D	<driver number>	Kernel Binding: Driver not found.	Ошибка при связывании переменной программы пользователя и устройства: не найден драйвер
21	<variable index>	Kernel TIC: Boundary check error.	Выход за пределы массива, при включённой опции «Check array index» в настройках ресурса (см раздел 2.5.2)
22	<variable index>	Kernel TIC: SINT divided by zero.	Выполнена операция деления на ноль над переменной типа SINT
23	<variable index>	Kernel TIC: DINT divided by zero.	Выполнена операция деления на ноль над переменной типа DINT
24	<variable index>	Kernel TIC: REAL divided by zero.	Выполнена операция деления на ноль над переменной типа REAL
27	<cycle time>	Cycle Time Overflow	Превышение текущего времени цикла программы пользователя над заданным. Текущее время цикла в мс приводится в параметре
29	<variable index>	Kernel TIC: INT divided by zero	Выполнена операция деления на ноль над переменной типа SINT
30	<variable index>	Kernel TIC: LINT divided by zero	Выполнена операция деления на ноль над переменной типа LINT
31	<variable index>	Kernel TIC: USINT divided by zero.	Выполнена операция деления на ноль над переменной типа SINT
32	<variable index>	Kernel TIC: UINT divided by zero.	Выполнена операция деления на ноль над переменной типа UINT
33	<variable index>	Kernel TIC: UDINT divided by zero.	Выполнена операция деления на ноль над переменной типа UDINT
34	<variable index>	Kernel TIC: ULINT divided by zero.	Выполнена операция деления на ноль над переменной типа ULINT
35	<variable index>	Kernel TIC: LREAL divided by zero.	Выполнена операция деления на ноль над переменной типа LREAL
36	0	Kernel TIC: Call stack overflow	Переполнение стека вызовов функций
50	1	Failover: Change of state	Смена состояния режима работы Failover, например обнаружение ошибки Failover или переключение с активной роли в резервную
52	<CanID>	Project: Virtual simple in non-virtual complex device	В составе сложного устройства с идентификатором на шине CAN равным CanID есть простое устройство, помеченное как

Код сообщ.	Параметр сообщения	Текст сообщения	Расшифровка
			виртуальное
53	<CanID>	Project: Multiple use of id.	В проекте есть два или более устройств с одинаковым идентификатором на шине CAN, равным CanID
54	<Scan number>	Resource cycling	Программа пользователя зациклилась на скане <Scan number>
55	0	Waiting for new Resource	Модуль CPU ожидает новой программы пользователя, либо сброса по питанию, чтобы снова начать выполнять текущую программу пользователя.
56	<port_id>	port_id=<n> incorrect	Порт с port_id=<n> некорректно настроен, недопустимое значение <n>
57	0	cpu_id incorrect	Драйвер какого-то устройства добавлен выше драйвера CPU и не может получить из драйвера CPU данные для работы
58	0	serial port on МК-503 is not available	Попытка добавить последовательный порт на МК-503-120. Но в CPU МК-503-120 нет последовательных портов.
59	0	port_id=<n> has no IP address, 127.0.0.1 assigned	Не настроен IP-адрес для порта с port_id=<n>, присвоен адрес 127.0.0.1
5A	0	iec104 EthernetPort should be upper than all iec104 devices	Устройство EthernetPort, на которое ссылаются устройства iec104_* (см. п. 3.7.3), должно находиться в Resource I/O выше всех этих устройств iec104_*.
5B	<mode>	Port mode=<n> incorrect	Порт с mode=<n> некорректно настроен, недопустимое значение <n>
120	0	KVB: Memory allocated is too short	Недостаточно памяти для хранения переменных.
180	0	KER: Slave number not allowed.	Конфликт ключей запуска ISaGRAF (например, на обоих CPU запущены ISaGRAF с ключом -PR)
200	0	SRV: Cannot allocate memory for server	Не удаётся создать внутренние объекты ISaGRAF. Может возникать при обновлении программы пользователя, если прежняя версия программы закрывалась долго (см. п. 5.4.3)
1001	-	Applying to CPU network settings from other CPU	Применяются сетевые настройки соседнего CPU (как правило, появляется при запуске на резервном CPU).
1002	-	Deleting isagraf project	Удаляется программа пользователя (при запуске на резервном CPU, либо при удалении программы пользователя из плагина).
2002	-	Isagraf not started: too many CPU detected	Обнаружено больше одного модуля CPU (всего больше двух модулей CPU) на CAN-шине. ISaGRAF не будет запущен
2003	-	Isagraf not started: CPU	Модуль CPU находится в запрещённой позиции (неправильный CanID-адрес, см. РЭ). ISaGRAF

Код сообщ.	Параметр сообщения	Текст сообщения	Расшифровка
		placed in wrong slot	не будет запущен.
2004	-	Isagraf not started: ADDRESS or BITRATE on PSU are incorrect	Неверно установлены значения переключателей ADDRESS и/или BITRATE на модуле питания МК-550-024. ISaGRAF не будет запущен
2005	-	SE Isagraf not started: Bridge is missing	Второй модуль CPU не может получить данные о настройках сети от первого модуля CPU в ходе инициализации. ISaGRAF не будет запущен
3001	2	route <n>: <error message>	В ходе применения маршрута с порядковым номером <n> возникла ошибка <error message>
3002	-	opwl config filename not defined in powerlinkMN device	В устройстве powerlinkMN не указан конфигурационный файл
3003	-	opwl config file '<filename>' not exists	В папке ftp не найден конфигурационный файл <filename>, указанный в устройстве powerlinkMN
3004	0	opwl CNsStates (n1 n2 n3) has too large bandwidth and skiped	В Powerlink-стойках с opwl-адресами n1, n2, n3 модули ввода-вывода превышают бюджет по вводу-выводу. Все стойки с перечисленными opwl-адресами будут игнорироваться
3005	-	cdc file not create	Ошибка при попытке создания конфигурации powerlinkMN

5.3.2 Журнал запуска среды выполнения и программы пользователя

При возникновении ошибок среды выполнения или программы пользователя может быть создан журнал запуска, в котором будет указана причина ошибки. Файл с журналом `/log/log.txt` можно скачать из модуля CPU проекта по протоколу FTP ([п. 2.3.1](#)).

5.3.3 Оценка времени исполнения отдельных секций цикла ISaGRAF

Оценивать время исполнения отдельных секций цикла ISaGRAF (input, синхронизация, output...) можно с использованием встроенных функций среды выполнения, которые позволяют отправлять отладочные сообщения с меткой времени в специальную программу-приёмник сообщений `udp_debug`. Для этого требуется сформировать и передать во все модули CPU проекта по протоколу FTP ([п. 2.3.1](#)) конфигурационный файл `isagraf.ini`. Отсутствие конфигурационного файла, либо ошибки в формате конфигурационного файла отключает данную возможность, не влияя на остальной функционал.

Конфигурационный файл должен быть составлен в следующем формате:

1. Секция конфигурации UDP отладки открывается заголовком `[UDPdebug]`. Далее следуют пары `ключ=значение`. Порядок следования ключей не критичен, между знаком равенства и ключом/значением допускается наличие пробела. Имена ключей и заголовков критичны к регистру. Все ключи, кроме IP имеют два значения: 1 – вкл., 0 – выкл.

Список ключей:

- 1.1 `Enabled` – включение/выключение отладки;
- 1.2 `IP` – сетевой адрес, на который отправляются UDP сообщения.

Пример:

```
IP=10.155.26.100;
```

- 1.3 `ShowPrSe` – добавлять или нет к сообщению метку `Pr/Se` (Primary/Secondary);

2. Секция конфигурации меток UDP сообщений открывается заголовком `[UDPlabel]`. К ней применимы те же требования, что и к секции `Udpdebug`.

Список ключей:

- 2.1 `KERSTART` – старт ядра среды выполнения;
- 2.2 `KERSTOP` – остановка ядра среды выполнения;
- 2.3 `RSTART` – старт ресурса;
- 2.4 `RSTOP` – остановка ресурса;
- 2.5 `BEGCYCEXEC` – начало цикла;
- 2.6 `NOCYCEXEC` – отсутствует выполнение цикла;
- 2.7 `ENDCYCEXEC` – окончание цикла;
- 2.8 `BEGINP` – начало фазы input;
- 2.9 `ENDINP` – окончание фазы input;
- 2.10 `BEGINP` – начало фазы output;
- 2.11 `BEGOUT` – окончание фазы output;
- 2.12 `ONLINECHANGESAVE` – произведен Online Change;
- 2.13 `BEGSCAN` – начало скана;
- 2.14 `SHOWWARN` – дублировать по UDP ошибки, отправленные в журнал запуска ([п. 5.3.1](#));

Пример содержимого конфигурационного файла приведён в [Листинг 5.3](#).

```
[UDPdebug]
Enabled=1
IP=10.155.26.25
ShowPrSe=0

[UDPlabel]
KERSTART=1
KERSTOP=1
RSTART=1
RSTOP=1
BEGCYCEXEC=0
BEGINP=0
ENDINP=0
BEGOUT=0
ENDOUT=0
NOCYCEXEC=0
ENDCYCEXEC=0
ONLINECHANGESAVE=1
BEGSCAN=0
SHOWWARN=1
```

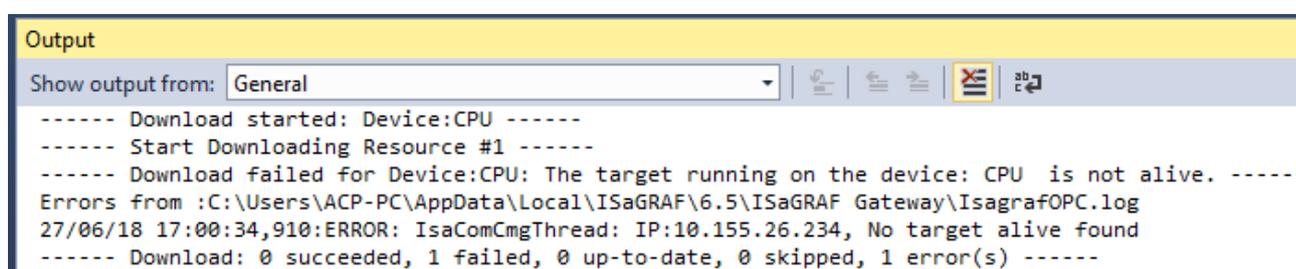
Листинг 5.3 – Пример файла конфигурации UDP отладки

5.4 Часто встречающиеся проблемы и меры по их устранению

5.4.1 Проблемы при загрузке программы пользователя

Самой частой ошибкой при попытке загрузить программу пользователя в модуль CPU является несовпадение сетевых параметров проекта и модуля CPU, проблема с сетевым подключением компьютера с установленной средой разработки ACP, либо невозможность подключиться к модулю CPU.

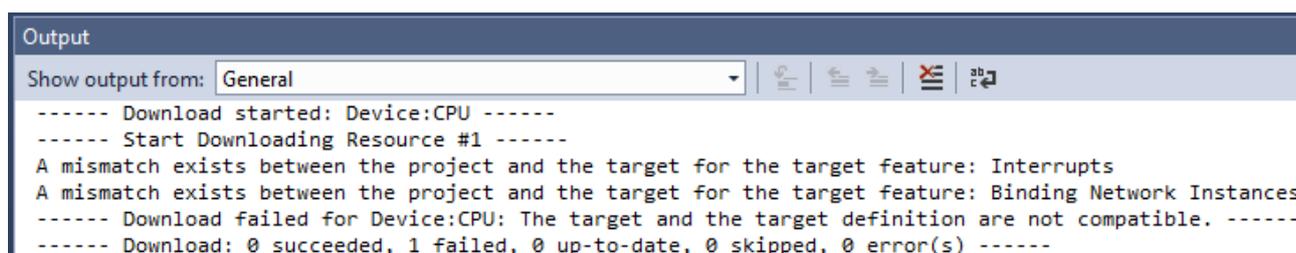
Признаком такой ошибки является сообщение об ошибке, приведённое на [Рис. 5.1](#). При её возникновении следует проверить сетевое подключение компьютера с установленной средой разработки ACP, корректность сетевых настроек компьютера и модуля CPU ([п. 2.3](#)), сетевые настройки проекта ([п. 2.4](#)).



```
Output
Show output from: General
----- Download started: Device:CPU -----
----- Start Downloading Resource #1 -----
----- Download failed for Device:CPU: The target running on the device: CPU is not alive. -----
Errors from :C:\Users\ACP-PC\AppData\Local\ISaGRAF\6.5\ISaGRAF Gateway\IsagrafOPC.log
27/06/18 17:00:34,910:ERROR: IsaComCmgThread: IP:10.155.26.234, No target alive found
----- Download: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped, 1 error(s) -----
```

Рис. 5.1 – Пример сообщения об ошибке при проблемах загрузки программы пользователя

Загрузка программы пользователя, собранной с неподходящим для модуля CPU tdb-файлом ([п. 2.5](#)) заканчивается ошибкой, приведённой на [Рис. 5.2](#). В случае её возникновения следует уточнить версии системного ПО используемого модуля CPU и его модификацию (с поддержкой режиме Failover или без, [п. 2.3](#)), и скорректировать проект (см. [п. 2.2](#)).



```
Output
Show output from: General
----- Download started: Device:CPU -----
----- Start Downloading Resource #1 -----
A mismatch exists between the project and the target for the target feature: Interrupts
A mismatch exists between the project and the target for the target feature: Binding Network Instances
----- Download failed for Device:CPU: The target and the target definition are not compatible. -----
----- Download: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped, 0 error(s) -----
```

Рис. 5.2 – Пример сообщения об ошибке при несовпадении tdb-файла либо версии среды исполнения ISaGRAF

5.4.2 Зацикливание программы пользователя

Среда исполнения ISaGRAF имеет сторожевой таймер (см. [п. 2.5.2](#)) для защиты программы пользователя от зацикливания (например, в случае ошибочно записанного вызова цикла типа WHILE).

При обнаружении зацикливания:

- исполнение программы пользователя в CPU прерывается;
- изменяется индикация CPU (начинает мигать светодиод RUN, см. КДСА.426471.004 РЭ);
- активный CPU отправляет сообщения с кодами 54 и 55 в журнал сообщений (см. [п. 5.3.1](#) и [Рис. 5.3](#));
- оба CPU переходят в режим ожидания загрузки программы пользователя.

В режиме ожидания программы пользователя среда исполнения ожидает загрузки программы пользователя. После загрузки программы пользователя CPU переходят в обычный режим работы с исполнением программы пользователя.

Также CPU переходят в обычный режим работы при перезагрузке по питанию. Этот вариант оставлен на случай, если зацикливанием программы пользователя происходит при определённом стечении обстоятельств и невозможности быстро исправить ошибку. В таком случае при перезагрузке по питанию зациклившихся модулей CPU можно попробовать продолжить работу.

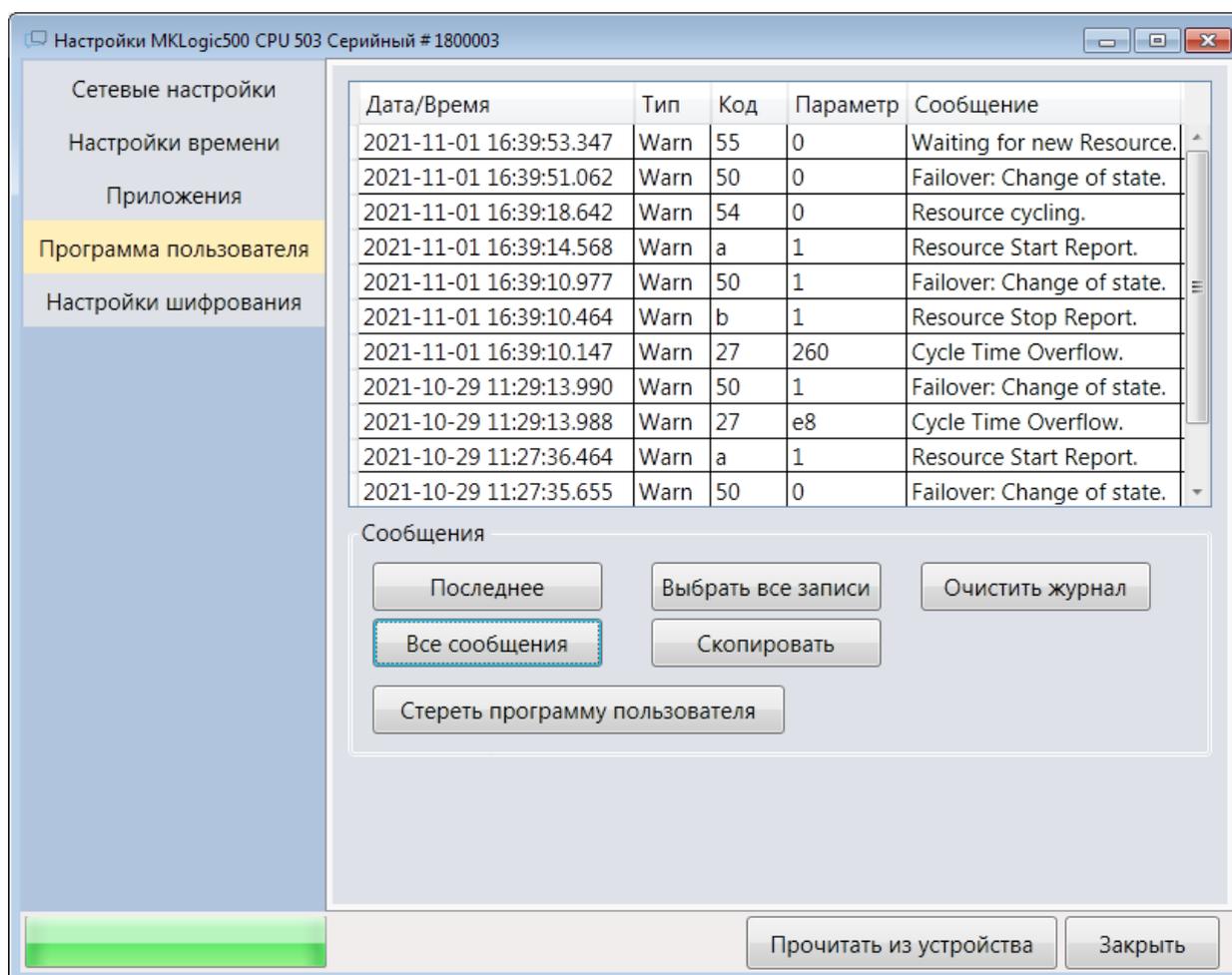


Рис. 5.3 – Сообщения об ошибках при зацикливании программы пользователя

5.4.3 Ошибка запуска программы пользователя

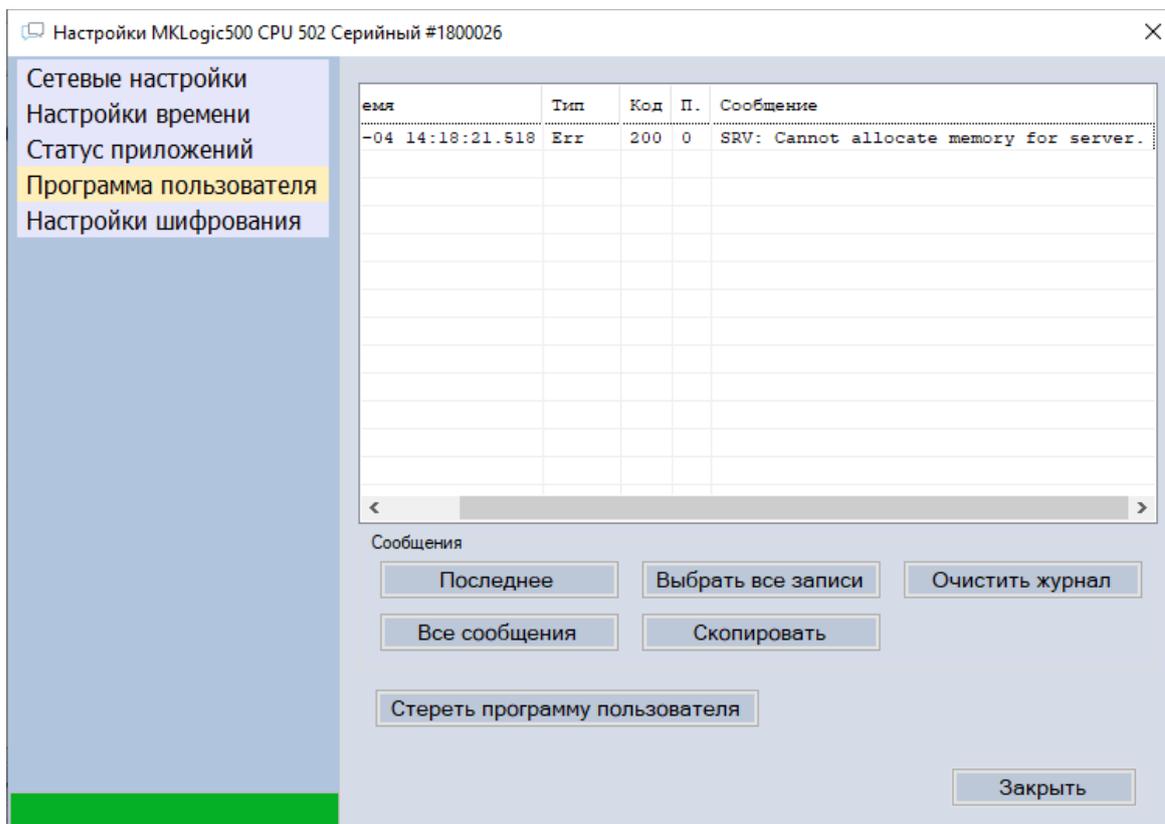


Рис. 5.4 – Диагностическое сообщение при ошибке запуска программы пользователя

Иногда при обновлении программы пользователя возникает ошибка «200 SRV: Cannot allocate memory for server» ([Рис. 5.4](#)). Она происходит в случае, если при обновлении программы пользователя прежняя версия программы закрывалась долго (45 секунд и более).

Способ устранения ошибки: с помощью плагина остановить приложение (согласно [п.2.3.2.3](#)), подождать не менее 1 минуты, затем стереть программу пользователя, дождаться перезапуска модуля CPU и загрузить в него программу пользователя заново.

5.4.4 Ложные ошибки диагностики

Диагностический флаг «Check Array Index» ([п. 2.5.2](#)) приводит к ложному срабатыванию защиты по выходу за пределы массива при использовании следующей конструкции ([Листинг 5.4](#)):

(*Не вызывает проблем*)

```
SomeIOdataArray[index] := 123;  
SomeIOdataArray[index + 1] := 456;
```

(*Приводит к остановке программы пользователя во время исполнения*)

```
SomeIOdataArray[index + 0] := 123; (*на этой строке сработает защита*)  
SomeIOdataArray[index + 1] := 456;
```

Листинг 5.4 – Пример кода, вызывающего остановку при установленном флаге «Check Array Index»

Ложное срабатывание возникает при доступе к массиву (привязанному к IODevice) с добавлением к переменной-индексу 0. При добавлении к переменной-индексу других констант срабатывание защиты возникает только при реальном выходе за пределы массива.

5.4.5 Ошибки при сборке проекта

При сборке проекта часто возникает ошибка вида <Имя_функции>: unexpected statement при попытке использовать функцию, не присваивая переменной возвращаемое функцией значение. Пример возникновения подобной ошибки приведён в [Листинг 5.5](#).

```
(*правильно res := SafeBoolArrayCopy(bArray1, bArray2);*)
SafeBoolArrayCopy(bArray1, bArray2);
(*При попытке скомпилировать будет ошибка:
SAFEBOOLARRAYCOPY: unexpected statement*)
```

Листинг 5.5 – Пример игнорирования возвращаемого значения функции

Также к интересным сообщениям об ошибках приводят пропуски закрывающих точек с запятой в строке, предшествующей строке с ключевым словом ST. В этом случае компилятор «склеивает» проблемную строку со следующей, и выдаёт ошибки, соответствующие отсутствию следующей строки, но не обязательно указывающие на пропуск точки с запятой. Пример подобного поведения приведён в [Листинг 5.6](#).

```
(*в следующей строке пропущена точка с запятой*)
regs := regs_2d[i].regs
WHILE j <= 1024 DO
    j := j+1;
END_WHILE;
(*При попытке скомпилировать возникают следующие ошибки:
=:must precede an expression having the same type as the assignment variable
END_WHILE:expected
WHILE:unexpected statement*)
```

Листинг 5.6 – Пример пропуска точки с запятой перед ключевым словом ST

5.4.6 Проблемы с работой модулей на CAN-шине

При проблемах с модулями ввода-вывода на шине CAN (полная неработоспособность, невозможность прочитать данные, необоснованные смены состояния и пр.) следует проверить следующие настройки:

1. На всех модулях блоков питания МК-550-024:
 - 1.1. Значения переключателей «ADDRESS», должны быть уникальны на каждой стойке. Если в стойке два блока питания – должны быть одинаковы для обоих. Не могут принимать значения 0 и 9.
 - 1.2. Значения переключателей «BITRATE», должны быть одинаковым в каждой стойке. Не могут принимать значения 0, 1 и 2. Значение 7 поддерживается только для модулей CPU МК-501-022.
 - 1.3. Положение переключателей «terminator» для каждой шины. На одной CAN-шине должен быть включен (состояние «1») как минимум один переключатель. Для длинной (более 10 метров) CAN-шины необходимо включать два переключателя, в начале и в конце шины. Если в стойке два блока питания – следует включать на каждом по одному переключателю, разному в каждом из двух блоков (для возможности горячей замены). Более подробно см. раздел 17.3.1 КДСА.426471.004 РЭ
 - 1.4. Качество монтажа проводов на разъёмах CAN1 и CAN2, отсутствие обрывов, коротких замыканий и перекрестных (CAN1 на CAN2) связей.
2. На модулях CPU:
 - 2.1. Модули CPU должны совпадать по типу с модулями CPU в конфигурации ввода-вывода, и должны располагаться на тех же позициях в стойках.
 - 2.2. Позиции модулей CPU в стойках должны быть разрешёнными: только сразу после блоков питания, не больше двух в первой стойке и одного во всех остальных, не более двух модулей CPU всего. Более подробно см. раздел 17.3.1 КДСА.426471.004 РЭ
 - 2.3. Резервированные модули CPU должны быть одного типа.
 - 2.4. Резервированные модули CPU должны иметь «перемычку» (прямую связь по Ethernet) на одном из Ethernet-портов (кроме Eth1, на нём «перемычку» поднимать запрещено), общем для обоих CPU.
 - 2.5. Индикаторы BusA/BusB на модулях CPU не должны мигать. Если мигают – это признак нерабочей шины CAN (короткое замыкание, отсутствие терминаторов...).
 - 2.6. Если на модуле/модулях CPU мигают индикаторы Run и Err, а из остальных горят только BusA и BusB, это значит, что программа пользователя отказалась запускаться на модуле CPU. Чаще всего это возникает при радикально не совпадающих tdb-файлах программы пользователя и модуля CPU, либо при зацикливании программы пользователя на основном CPU.
 - 2.7. Если при одновременном запуске двух модулей CPU в работу (включением питания шкафа, например) при старте программы пользователя наблюдается кратковременное (2-3 раза) мигание всей индикации модулей CPU - значит, на самом деле они друг друга не видят по CAN и по «перемычке», и стартуют по отдельности. Если модуль CPU стартует без резерва - такое поведение индикации нормально и ожидаемо. Но уже второй модуль CPU, запускаемый после первого, так вести себя не должен.

2.8. После запуска и первого прохода бегущей строки модуля CPU не должно быть адресов вида 0.0.0.0, такой адрес говорит о проблеме с соответствующим сетевым интерфейсом. Чаще всего случается при проблемах с модулями МК-544-040 и МК-546-010.

3. Самая частая ошибка монтажа – не до конца защёлкнутые задние шины. Эта ошибка проявляется как спонтанные пропадания по питанию либо по шине CAN одного или нескольких модулей ввода-вывода. Визуально определяется по небольшому зазору между рядом стоящими модулями, при нормальном подключении шин зазора практически нет.
4. Также для определения проблем на шинах CAN можно использовать возможности диагностического WEB-сервера (см. [п.2.3.2.3](#)).
5. В журнале сообщений (см. [п. 5.3.1](#)) следует поискать сообщение с кодом 53 «Project: Multiple use of id». Если оно есть, следует проверить параметры модулей ввода-вывода в программе пользователя на предмет дублирующихся адресов rack и slot.
6. Неправильно выбранный в I/O Device модуль (например, МК-514-008А вместо МК-514-008) не будет определяться на шине CAN (на нём будет гореть только светодиод RUN).
7. Если в стойке есть разрыв (один или несколько модулей ввода-вывода в середине стойки не установлен) и модуль ввода-вывода справа от разрыва заменяется другим модулем, есть вероятность, что новый модуль раньше имел другой адрес на шине CAN. В этом случае этот модуль и все модули справа от него получают другие адреса, не обязательно совпадающие с проектными. Это происходит потому, что адреса на шине CAN не раздаются через разрыв.
Чтобы этого избежать, следует избегать разрывов. Если разрыв неизбежен, лучше поставить в разрыв любой другой модуль ввода-вывода (кроме блока питания МК-550-024).

5.4.7 Проблемы при работе по протоколу Modbus

При проблемах в работе с протоколом Modbus (не удаётся подключиться клиентом, не выполняются запросы) следует проверить следующие настройки:

1. Должно присутствовать устройство `modbusXXX_xxx_xxx`. Расположено оно должно быть ниже всех модулей CPU.
2. В устройстве `modbusXXX_mst_xxx` для начала выполнения запросов следует записать 1 в выходное устройство `mb_control_`. Для устройств `modbusXXX_sl_xxx` это не обязательно, они работают всегда.
3. Частой ошибкой бывает привязка устройства не к тому порту (параметр `port_id` в дополнительном устройстве `EthernetPort_` или `SerialPort_`). Режим работы (параметр `mode`) можно не устанавливать, для modbus-устройств он выбирается автоматически.
4. Для master-устройства следует убедиться, что у необходимых запросов установлено в TRUE поле `Enable`.
5. При подключении по Modbus к модулю CPU в резерве (в режиме `Secondary`) на все запросы будет возвращаться код ошибки 6 – `Device busy`. Это нормальное поведение резервного модуля CPU.
6. При подключении по ModbusRTU к модулю CPU следует помнить, что `slaveID` для ModbusRTU резервного CPU увеличен на 32 (см. [п. 3.7.2.1](#)).

5.4.8 Проблемы при работе по протоколу IEC 60870-5-104

При проблемах в работе с протоколом IEC 60870-5-104 (не удаётся подключиться клиентом, нет обновления данных и пр.) следует проверить следующие настройки:

1. Должно присутствовать устройство `EthernetPort_` с параметром `mode=3` (`iec104`). Также должен быть правильно указан TCP-порт, правильно указано имя файла конфигурации. Само устройство `EthernetPort_` должно находиться ниже всех модулей CPU, но выше всех устройств `iec104_xxx`.
2. Файл конфигурации обязателен, находится он должен в корне папки FTP.
3. В файле конфигурации не должно быть расхождений с конфигурацией переменных устройств `iec104_xxx`. Если типы, адреса (`ioa`) и/или количество переменных не совпадает – сервер не запустится. Следует обращать внимание на адреса возле границ диапазонов.
4. При подключении к модулю CPU в резерве (в режиме `Secondary`) данные не будут изменяться. Это нормальное поведение резервного модуля CPU для `iec104`.

5.4.9 Проблемы при работе по протоколу OPC-UA

При проблемах в работе с протоколом OPC-UA (не удаётся подключиться клиентом, нет обновления данных и пр.) следует проверить следующие настройки:

1. Должно присутствовать устройство `opcua_server`. Расположено оно должно быть ниже всех модулей CPU.
2. В дополнительном устройстве `EthernetPort_` устройства `opcua_server` должен быть выбран режим OPC-UA server, правильно указан TCP-порт, правильно указано имя файла конфигурации.
3. Файл конфигурации обязателен, находится он должен в корне папки FTP.
4. В файле конфигурации не должно быть ссылок на неподдерживаемые типы переменных (массивы структур, массивы переменных-массивов, структуры с массивами).
5. Подключение к модулю CPU в резерве (в режиме Secondary) будет блокироваться. Это нормальное поведение резервного модуля CPU для OPC-UA.

5.4.10 Проблемы при работе по протоколу Powerlink

При проблемах в работе с протоколом Powerlink (нет обмена по Powerlink, нет обмена с одной из стоек за CN, постоянно пропадает одна из стоек за CN) следует проверить следующие настройки:

1. В сетевых настройках модулей CPU (см. [п. 2.3.2.1](#)) должна быть выбрана роль PowerlinkMN.
2. Не должно быть сбоя при инициализации МК-546-010 (в бегущей строке на модуле CPU для модуля МК-546-010 не должен выводиться 0.0.0.0 в качестве IP-адреса).
3. Поддерживается работа только с одним модулем расширения МК-546-010, при наличии в слотах расширения одного модуля CPU двух и более модулей расширения МК-546-010 нормальная работа системы не гарантируется.
4. В шине Powerlink все кабеля должны быть хорошо обжаты. Проверить либо заменить кабеля нужно при спонтанных потерях связи по Powerlink.
5. В шине Powerlink не должно быть Ethernet-трафика: не должно быть пересечения сетей общего назначения с сетью Powerlink, в сети Powerlink не должны присутствовать модули МК-545-010 (далее CN) в режиме настройки (см. раздел 15.6 КДСА.426471.004 РЭ).
6. В конфигурации должно присутствовать устройство PowerlinkMN. Расположено оно должно быть ниже всех модулей CPU.
7. В дополнительном устройстве EthernetPort_ устройства PowerlinkMN должен быть выбран режим OpenPowerlink и правильно указано имя файла конфигурации. Параметр port_id указывать не обязательно, рабочий порт Powerlink определяется автоматически.
8. В конфигурации должны присутствовать устройства cn545 с корректными настройками параметров (pwl_id, rack и slot соответствующей стойки).
9. Ожидается, что в стойках CN есть модули ввода-вывода. У этих модулей ввода-вывода должны быть правильно настроены параметры pwl_id, rack и slot.
10. Если хотя бы один из модулей ввода-вывода в стойке CN – модуль МК-541-002 (или в стойке CN просто очень много модулей ввода-вывода), то в корне папки FTP должен находиться файл конфигурации с настроенными коэффициентами кадрирования
11. Адреса Powerlink на модулях CN должны быть уникальными, и они должны совпадать с проектными.
12. Адреса CAN для всех стоек CN должны начинаться с 1.
13. Стойки разных CN не должны быть связаны между собой по CAN. Если в рамках одной CN заложены в проекте несколько стоек CAN, они должны быть связаны по CAN.
14. Также для определения проблем на шине Powerlink можно использовать возможности диагностического WEB-сервера (см. [п. 2.3.2.3](#)).

5.5 Сетевая безопасность

При работе модулей CPU используется ряд TCP-портов (см. [Табл. 1.78](#)). Для нормальной работы доступ к ним со стороны среды разработки не должен блокироваться средствами сетевой безопасности.

Табл. 1.78– TCP-порты, открытые на модулях CPU

TCP-порт	Назначение
21	FTP-сервер (см. п. 2.3.1 , п.4.27)
123, 323	NTP-сервер (см. п.2.3.2.2)
502	Сервер ModbusTCP (см. п.3.7.2.1)
2404	Сервер IEC 60870-5-104 (см. п.3.7.3)
4840	Сервер OPC-UA (см. п.3.7.4.1)
1131	Загрузка проектов в среду исполнения ISaGRAF (см. п.2.7)
6670	Настройка параметров модулей CPU с помощью плагина MK500 IODevice (см. п. 2.3.1)

Приложение А. Инструкция по интеграции в PDM-систему

Модули ввода-вывода МК-574-008А, МК-576-008А и МК-576-016А могут интегрироваться в PDM-системы, поддерживающие формат DTM (PactWare, FDT Container, PRM Yokogawa). Модули МК-574-008А и МК-576-008А по сути являются 8-ми канальными HART-модемам, модуль МК-576-016А – 16-ти канальным HART-модемом.

Для того чтобы подключить несколько модулей в одну шину, модули необходимо предварительно сконфигурировать. Конфигурация заключается в задании базового смещения адресации (0...7) каналов Hart.

Конфигурация осуществляется через среду разработки ACP ISaGRAF заданием параметра `addrOffset` устройства `hart8_` (для МК-576-016А – `hart16_`) от 0 до 7, по умолчанию смещение равно 0.

Базовое смещение модуля соответствуют следующим адресам:

- 0 – адреса 0, 1, 2, 3, 4, 5, 6, 7
- 1 – адреса 8, 9, 10, 11, 12, 13, 14, 15
- 2 – адреса 16, 17, 18, 19, 20, 21, 22, 23
- 3 – адреса 24, 25, 26, 27, 28, 29, 30, 31
- 4 – адреса 32, 33, 34, 35, 36, 37, 38, 39
- 5 – адреса 40, 41, 42, 43, 44, 45, 46, 47
- 6 – адреса 48, 49, 50, 51, 52, 53, 54, 55
- 7 – адреса 56, 57, 58, 59, 60, 61, 62, 63

Для работы, необходимо установить DTM с поддержкой многоканального HART-modem, например, бесплатный Hart Communication фирмы CodeWrights. Также необходимо установить DTM подключенного датчика (на сайте производителя датчика). После чего необходимо обновить библиотеку PDM системы. Все подключенные датчики должны иметь базовый адрес 0.

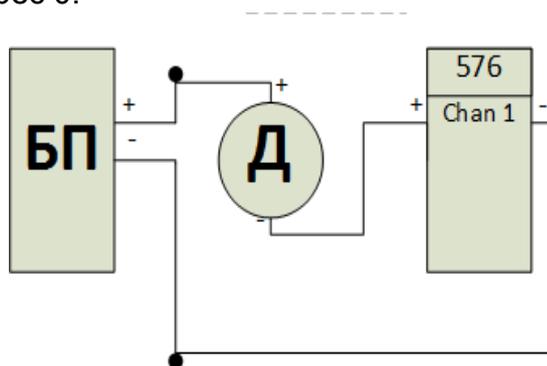


Рис. А.1 – Схема подключения МК576 с неактивным датчиком

Пример настройки будет рассмотрен ниже на примере модуля МК-576-008А (работа с МК-574-008А и МК-576-016А выполняется аналогично), ПО FDT container, DTM Hart Communication CodeWrights и датчика давления Siemens Sitrans P DSIII. Модуль МК-576-008А подключен к ПК через преобразователь Moxa UPort1150.

В ПО FDT Container выбираем Hart Communication (Рис. А.2).

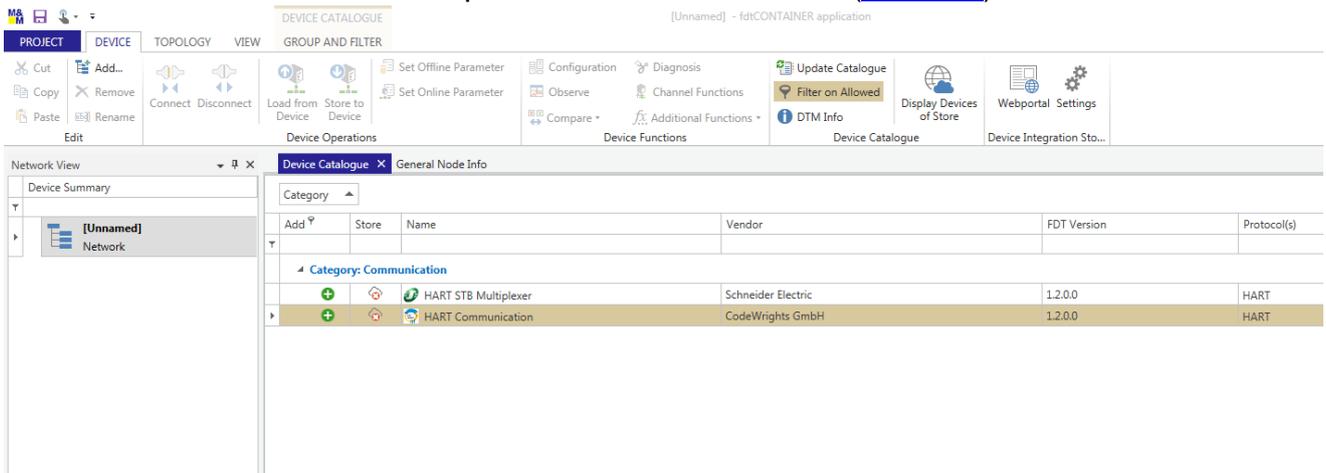


Рис. А.2 – Окно подключения устройств

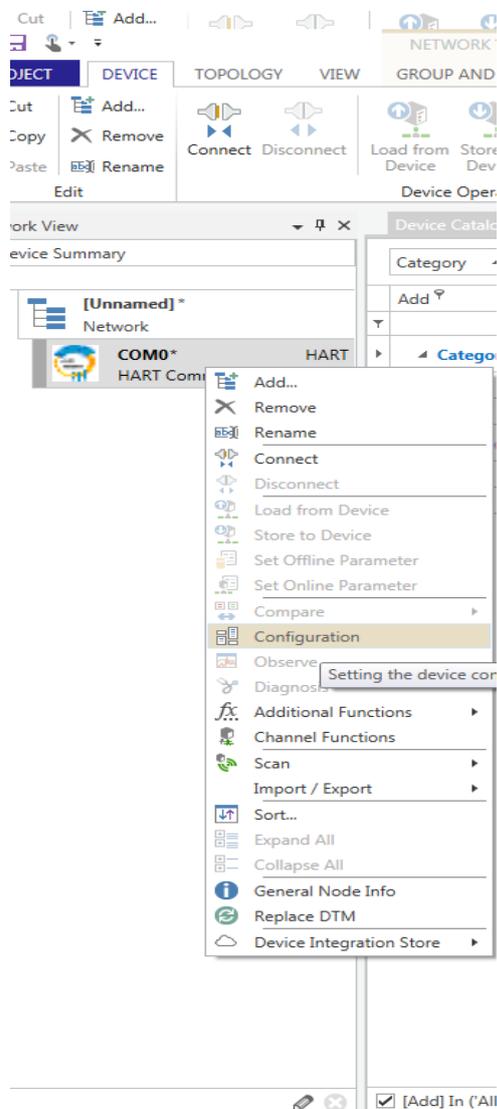


Рис. А.3 – Настройки Hart Communication

Настраиваем в соответствии с [Рис. А.4](#), номер COM порта должен указывать на адаптер USB-RS485 (на [Рис. А.4](#) COM5 Uport1150).

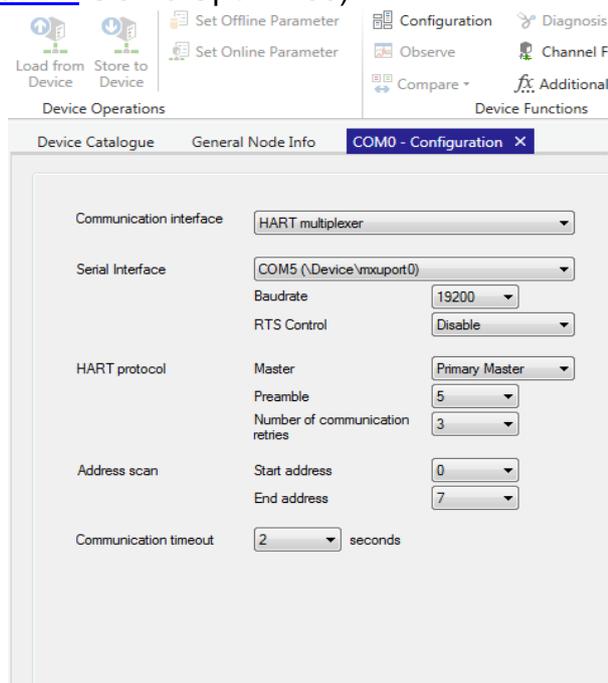


Рис. А.4 – Настройки коммуникаций

Принимаем настройки и выбираем сканирование Hart ([Рис. А.5](#)).

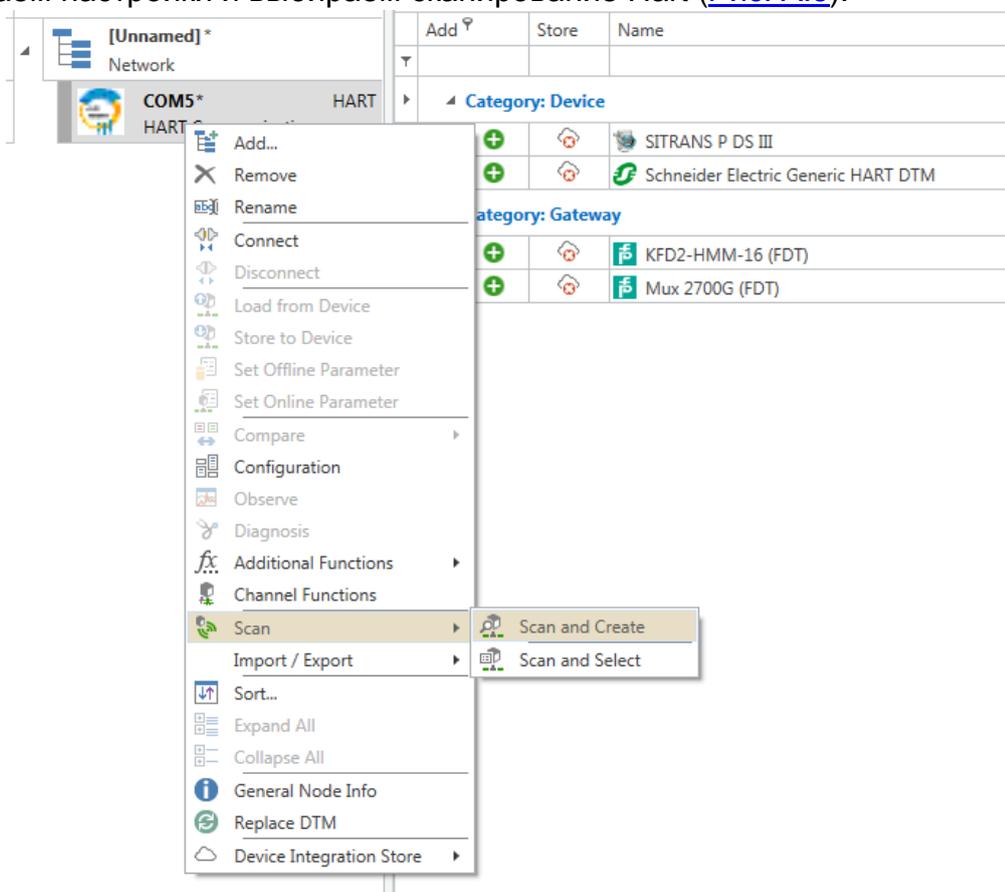


Рис. А.5 – Контекстное меню

Датчик будет найден под номером канала, к которому он подключен. Если датчик найден, то схема собрана верно, Hart канал работает корректно и RS485 канал работает корректно.

Датчик на 0-м канале будет иметь адрес 0, 8, 16, 24, 32, 40, 48 или 56 в зависимости от базового смещения. Например, если модуль имеет базовое смещение 5, датчик подключен на 3 канал, то в FDT Container датчик будет иметь адрес 42.

Приложение Б. Индикация модулей МКLogic-500

Табл 1. Индикаторы модуля центрального процессора МК-501-022

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Не запущен процесс isagraf
	Мигает 1 раз в секунду	Запущен процесс isagraf, программа пользователя отсутствует
	Мигает 2 раза в секунду	-
	Горит	Запущен процесс isagraf и запущена программа пользователя в isagraf
Err (красный)	Не горит	Нет ошибок в конфигурации и при выполнении программы пользователя. Не запущен процесс isagraf, либо программа пользователя отсутствует.
	Мигает 1 раз в секунду	Ошибка конфигурации одного или нескольких модулей ввода-вывода — либо отсутствует предусмотренный текущей конфигурацией модуль, либо установлен модуль, не соответствующий текущей конфигурации.
	Мигает 2 раза в секунду	Ошибка при старте или в ходе выполнения isaVM (программа пользователя не запустилась/ остановлена).
	Горит	Отказ одного из процессов встроенного ПО самого контроллера (PLC*).
Prim (зелёный)	Не горит	Не запущен процесс isagraf.
	Мигает 1 раз в секунду	Запущен isagraf в режиме SE («ведомый»).
	Мигает 2 раза в секунду	-
	Горит	Запущен isagraf в режиме PR («ведущий»), либо отключён режим резервирования и восстановления после отказа (режим Failover)
ACT (зелёный)	Не горит	Не запущен процесс isagraf, либо программа пользователя отсутствует
	Мигает 1 раз в секунду	Запущен isagraf, программа пользователя запущена, но её время выполнения выходит за пределы заданного в пользовательской программе времени опроса
	Мигает 2 раза в секунду	Запущен isagraf, программа пользователя запущена, но центральный процессор модуля загружен более чем на 90%
	Горит	Запущен isagraf, программа пользователя запущена
FO Err (красный)	Не горит	Нет ошибок режима резервирования и восстановления после отказа (режим Failover) либо он не настроен/не используется
	Мигает 1 раз в секунду	-
	Горит	Запущен isagraf в режиме резервирования и восстановления после отказа (режим Failover), и вышел из строя второй модуль центрального процессора.
BusA (жёлтый)	Не горит	Не используется шина А
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе сап-драйвера шины А/неисправность шины А
	Горит	Шина А работает в штатном режиме
BusB (жёлтый)	Не горит	Не используется шина В
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе CAN-драйвера шины В/ неисправность шины В
	Горит	Шина В работает в штатном режиме

Индикатор	Состояние	Описание
Rx (жёлтый или зелёный)	Мигает с интенсивностью приёма	Ход приёма данных по интерфейсу RS-485
Tx (зелёный)	Мигает с интенсивностью передачи	Ход передачи данных по интерфейсу RS-485
Поле семисегментных индикаторов	Бегущая строка	IP-адреса модуля центрального процессора

Табл 2. – Индикаторы модуля центрального процессора МК-502-142

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Не запущен процесс isagraf
	Мигает 1 раз в секунду	Запущен процесс isagraf, программа пользователя отсутствует
	Мигает 2 раза в секунду	-
	Горит	Запущен процесс isagraf и запущена программа пользователя в isagraf
Err (красный)	Не горит	Нет ошибок в конфигурации и при выполнении программы пользователя. Не запущен процесс isagraf, либо программа пользователя отсутствует
	Мигает 1 раз в секунду	Ошибка конфигурации одного или нескольких модулей ввода-вывода — либо отсутствует предусмотренный текущей конфигурацией модуль, либо установлен модуль, не соответствующий текущей конфигурации
	Мигает 2 раза в секунду	Ошибка при старте или в ходе выполнения IsaVM (программа пользователя не запустилась/ остановлена)
	Горит	Отказ одного из процессов встроенного ПО самого контроллера (PLC*)
Prim (зелёный)	Не горит	Не запущен процесс isagraf
	Мигает 1 раз в секунду	Запущен isagraf в режиме SE («ведомый»).
	Мигает 2 раза в секунду	-
	Горит	Запущен isagraf в режиме PR («ведущий»), либо отключён режим резервирования и восстановления после отказа (режим Failover)
ACT (зелёный)	Не горит	Не запущен процесс isagraf, либо программа пользователя отсутствует
	Мигает 1 раз в секунду	Запущен isagraf, программа пользователя запущена, но её время выполнения выходит за пределы заданного в пользовательской программе времени опроса
	Мигает 2 раза в секунду	Запущен isagraf, программа пользователя запущена, но центральный процессор модуля загружен более чем на 90%
	Горит	Запущен isagraf, программа пользователя запущена
FO Err (красный)	Не горит	Нет ошибок режима резервирования и восстановления после отказа (режим Failover) либо он не настроен/не используется
	Мигает 1 раз в секунду	-
	Горит	Запущен isagraf в режиме резервирования и восстановления после отказа (режим Failover), и вышел из строя второй модуль центрального процессора
BusA (жёлтый)	Не горит	Не используется шина А
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе can-драйвера шины А/неисправность шины А
	Горит	Шина А работает в штатном режиме
BusB (жёлтый)	Не горит	Не используется шина В
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе CAN-драйвера шины В/ неисправность шины В
	Горит	Шина В работает в штатном режиме
Rx (жёлтый или зелёный)	Мигает с интенсивностью приёма	Ход приёма данных по интерфейсу RS-485
Tx (зелёный)	Мигает с интенсивностью передачи	Ход передачи данных по интерфейсу RS-485
Поле семисегментных индикаторов	Бегущая строка	IP-адреса модуля центрального процессора

Табл 3. – Индикаторы модуля центрального процессора МК-502-142 DCS

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Среда исполнения NaftaProcess не загружена или работает неправильно
Err (красный)	Не горит	Нет ошибок в конфигурации и в работе среды исполнения
	Горит	Отказ одного из процессов среды исполнения или самого контроллера (PLC), ошибка в конфигурации или в работе среды исполнения.
Prim (зелёный)	Не горит	Ошибка в работе среды исполнения или база функциональных блоков не загружена
	Мигает 1 раз в секунду	Среда исполнения запущена в режиме «Ведомый»
	Горит	Среда исполнения запущена в режиме «Ведущий»
ACT (зелёный)	Не горит	Ошибка в работе среды исполнения или база функциональных блоков не загружена
	Мигает 1 раз в секунду	Среда исполнения запущена в режиме «Резервный»
	Горит	Среда исполнения запущена в режиме «Активный»
FO Err (красный)	Не горит	Среда исполнения в состоянии «Синхронизированно»
	Горит	Среда исполнения в состоянии «Идет синхронизация»
BusA (жёлтый)	Не горит	Не используется шина А / Отсутствуют или неисправны все модули на шине
	Мигает 1 раз в секунду	Часть модулей на шине А отсутствует или неисправна
	Горит	Шина А работает в штатном режиме
BusB (жёлтый)	Не горит	Не используется шина В / Отсутствуют или неисправны все модули на шине
	Мигает 1 раз в секунду	Часть модулей на шине В отсутствует или неисправна
	Горит	Шина В работает в штатном режиме
Rx (жёлтый или зелёный)	Мигает с интенсивностью приёма	Ход приёма данных по интерфейсу RS-485
Tx (зелёный)	Мигает с интенсивностью передачи	Ход передачи данных по интерфейсу RS-485
Поле семисегментных индикаторов	Заставка	Идет загрузка модуля центрального процессора
	Бегущая строка	IP-адреса модуля центрального процессора

Табл 4. – Индикаторы модулей центрального процессора МК-503-120 и МК-504-120

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Не запущен процесс isagraf/CODESYS
	Мигает 1 раз в секунду	Запущен процесс isagraf/CODESYS, программа пользователя отсутствует
	Мигает 2 раза в секунду	-
	Горит	Запущен процесс isagraf/CODESYS и запущена программа пользователя в isagraf
Err (красный)	Не горит	Нет ошибок в конфигурации и при выполнении программы пользователя. Не запущен процесс isagraf/CODESYS, либо программа пользователя отсутствует
	Мигает 1 раз в секунду	Ошибка конфигурации одного или нескольких модулей ввода-вывода — либо отсутствует предусмотренный текущей конфигурацией модуль, либо установлен модуль, не соответствующий текущей конфигурации
	Мигает 2 раза в секунду	Ошибка при старте или в ходе выполнения (программа пользователя не запустилась/ остановлена)
	Горит	Отказ одного из процессов встроенного ПО самого контроллера (PLC*)
Prim (зелёный)	Не горит	Не запущен процесс isagraf/CODESYS
	Мигает 1 раз в секунду	Запущен isagraf/CODESYS в режиме SE («ведомый»)
	Мигает 2 раза в секунду	-
	Горит	Запущен isagraf/CODESYS в режиме PR («ведущий»), либо отключён режим резервирования и восстановления после отказа (режим Failover)
ACT (зелёный)	Не горит	Не запущен процесс isagraf/CODESYS, либо программа пользователя отсутствует
	Мигает 1 раз в секунду	Запущен isagraf/CODESYS, программа пользователя запущена, но её время выполнения выходит за пределы заданного в пользовательской программе времени опроса
	Мигает 2 раза в секунду	Запущен isagraf/CODESYS, программа пользователя запущена, но центральный процессор модуля загружен более чем на 90%
	Горит	Запущен isagraf, программа пользователя запущена
FO Err (красный)	Не горит	Нет ошибок режима резервирования и восстановления после отказа (режим Failover) либо он не настроен/не используется
	Мигает 1 раз в секунду	Нет ошибок резервирования isagraf/CODESYS, но ещё не синхронизирован обмен по Powerlink между основным и резервным модулем центрального процессора
	Горит	Запущен isagraf/CODESYS в режиме резервирования и восстановления после отказа (режим Failover), и вышел из строя резервный модуль центрального процессора
BusA (жёлтый)	Не горит	Не используется шина А
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе CAN-драйвера шины А/неисправность шины А
	Горит	Шина А работает в штатном режиме

Индикатор	Состояние	Описание
BusB (жёлтый)	Не горит	Не используется шина В
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе CAN-драйвера шины В/ неисправность шины В
	Горит	Шина В работает в штатном режиме
Поле семисегментных индикаторов	Заставка	Идет загрузка модуля центрального процессора
	Бегущая строка	IP-адреса модуля центрального процессора

Табл 5. Индикаторы модуля МК-545-010

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Не загружена конфигурация Powerlink
	Мигает 1 раз в секунду	Ошибка коммуникации с Powerlink MN (потеря связи после начала работы)
	Мигает 2 раза в секунду	-
	Горит	Загружена корректная конфигурация Powerlink
Err (красный)	Не горит	Модуль исправен и не имеет ошибок
	Мигает 1 раз в секунду	Ошибка конфигурации одного или нескольких модулей ввода-вывода — либо отсутствует предусмотренный текущей конфигурацией модуль, либо установлен модуль, не соответствующий текущей конфигурации
	Мигает 2 раза в секунду	Недопустимый адрес модуля (недопустимое положение переключателей Node Address)
	Горит	Отказ одного из процессов встроенного ПО самого коммуникационного модуля (PLC*)
ACT (зелёный)	Не горит	Модуль не активен (не выполняет опрос по шинам CAN)
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Центральный процессор модуля загружен более чем на 90%
	Горит	Модуль активен и выполняет опрос по шинам CAN
BusA (жёлтый)	Не горит	Не используется шина А
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе сап-драйвера шины А/неисправность шины А
	Горит	Шина А работает в штатном режиме
BusB (жёлтый)	Не горит	Не используется шина В
	Мигает 1 раз в секунду	-
	Мигает 2 раза в секунду	Ошибки в работе CAN-драйвера шины В/ неисправность шины В
	Горит	Шина В работает в штатном режиме
PWL (зеленый)	Не горит	Не поступают данные формата POWERLINK
	Горит	Поступают данные формата POWERLINK
ETH (зеленый)	Не горит	Не поступают данные формата Ethernet
	Горит	Поступают данные формата Ethernet
RING (зеленый)	Не горит	-
	Горит	Работа в режиме кольцевой топологии сети POWERLINK
LINE (зеленый)	Не горит	-
	Горит	Работа в режиме линейной топологии сети POWERLINK

Табл 6. Индикаторы модуля блока питания МК-550-024

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Входное напряжение отсутствует или ниже допустимого (видно только при резервировании питания). Или внутренний отказ
	Мигает	<ul style="list-style-type: none"> - переключатели адреса или скорости в неверном положении; - у модуля питания, расположенного слева неверное положение переключателей; - нет связи с модулем питания, расположенным слева; - положение переключателей на основном и резервном блоке питания допустимые, но не совпадают
BusA BusB (жёлтый)	Горит непрерывно	Модуль питания проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль питания не проинициализирован модулем центрального процессора

Табл 7. Индикаторы модулей дискретного ввода МК-521-032, МК-521-032 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль дискретного ввода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль дискретного ввода не проинициализирован модулем центрального процессора
Индикаторы входов (зелёный)	Горит непрерывно	Дискретный вход в состоянии логической единицы
	Не горит	Нет сигнала на дискретном входе, либо вход отключён

Табл 8. Индикаторы модуля дискретного ввода МК-523-032 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль дискретного ввода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль дискретного ввода не проинициализирован модулем центрального процессора
Индикаторы входов (зелёный)	Горит	Дискретный вход в состоянии логической единицы
	Мигает	Устранимый отказ цепи (обрыв либо короткое замыкание)
	Не горит	Дискретный вход в состоянии логического нуля, либо вход отключён

Табл 9. – Индикаторы модулей дискретного вывода МК-531-032, МК-531-032 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль дискретного вывода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль дискретного вывода не проинициализирован модулем центрального процессора
Индикаторы выходов (зелёный)	Горит непрерывно	Дискретный выход находится в состоянии логической единицы
	Не горит	Нет сигнала на дискретном выходе, либо выход отключён

Табл 10. Индикаторы модуля дискретного вывода МК-532-032 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль дискретного ввода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль дискретного ввода не проинициализирован модулем центрального процессора
Индикаторы выходов (зелёный)	Горит	Дискретный выход в состоянии логической единицы
	Мигает	Устранимый отказ цепи (обрыв, короткое замыкание либо отсутствие внешнего напряжения)
	Не горит	Дискретный выход в состоянии логического нуля, либо выход отключён

Табл 11. Индикаторы модулей аналогового ввода МК-513-016, МК-513-016 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит	Модуль аналогового ввода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль аналогового ввода не проинициализирован модулем центрального процессора
Индикаторы входов (зелёный)	Горит	Входной токовый сигнал в пределах допустимого диапазона 0...20 (4...20) мА. Допустимый диапазон настраивается программно
	Мигает	Уровень входного токового сигнала превышает 20 мА
	Не горит	Уровень входного токового сигнала ниже 4 мА. Используется, если настроен рабочий диапазон сигнала (4...20) мА

Табл 12. Индикаторы модуля аналогового ввода МК-516-008 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль аналогового ввода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль аналогового ввода не проинициализирован модулем центрального процессора
Индикаторы входов (зелёный)	Горит непрерывно	Входной токовый сигнал в пределах допустимого диапазона 0...20 (4...20) мА. Допустимый диапазон настраивается программно
	Мигает	Уровень входного токового сигнала превышает 20 мА
	Не горит	Уровень входного токового сигнала ниже 4 мА. Используется, если настроен рабочий диапазон сигнала (4...20) мА

Табл 13. – Индикаторы модулей аналогового вывода МК-514-008, МК-514-008 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Внутренний отказ
	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине); – диагностирован обрыв цепи канала
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль аналогового вывода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль аналогового вывода не проинициализирован модулем центрального процессора
Индикаторы выходов (зелёный)	Горит непрерывно	Выходной токовый сигнал в пределах диапазона 0...20 (4...20) мА
	Мигает	Диагностирован обрыв цепи канала

Табл 14. Индикаторы модуля аналогового вывода МК-574-008 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Горит	Отсутствует напряжение питания или напряжение питания за пределами допустимого диапазона (18...30 В)
	Мигает	<ul style="list-style-type: none"> – ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине); – несоответствие допустимому значению уровня тока на входе
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль аналогового вывода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль аналогового ввода не проинициализирован модулем центрального процессора
Индикаторы входов (зелёный)	Горит непрерывно	Выходной токовый сигнал в пределах диапазона 0...20 (4...20) мА
	Мигает	Диагностирован обрыв цепи канала
	Не горит	Выход программно выключен
Индикаторы работы HART (зелёный)	Горит непрерывно	Канал сконфигурирован для работы по HART
	Мигает	Диагностирован аппаратный отказ HART канала
	Не горит	Канал не сконфигурирован для работы по HART

Табл 15. Индикаторы модулей аналогового ввода МК-576-008 А, МК-576-016 А

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Мигает	<ul style="list-style-type: none"> – ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине); – несоответствие допустимому значению уровня тока на входе
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Модуль аналогового ввода проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Модуль аналогового ввода не проинициализирован модулем центрального процессора
Индикаторы входов (зелёный)	Горит непрерывно	Входной токовый сигнал в пределах допустимого диапазона 0-20 (4-20) мА. Допустимый диапазон настраивается программно
	Мигает	Уровень входного токового сигнала превышает 20 мА
	Не горит	Уровень входного токового сигнала ниже 4 мА. Используется, если настроен рабочий диапазон сигнала (4-20) мА
Индикаторы работы HART (зелёный)	Горит непрерывно	Канал сконфигурирован для работы по HART
	Мигает	Диагностирован аппаратный отказ HART канала
	Не горит	Канал не сконфигурирован для работы по HART

Табл 16. – Индикаторы коммуникационного модуля МК-541-002

Индикатор	Состояние	Описание
Run (зелёный)	Горит непрерывно	Встроенное ПО запущено и работает
	Мигает	Обнаружена критическая аппаратная ошибка либо отказ встроенного ПО. Модуль выведен из работы. Ошибка не может быть сброшена перезагрузкой модуля.
	Не горит	Встроенное ПО не запущено
Err (красный)	Мигает	– ошибка конфигурации (по интерфейсной шине в модуль загружены некорректные параметры); – конфигурация отсутствует (модуль не получил параметры по интерфейсной шине)
	Не горит	Ошибки не обнаружены
BusA BusB (жёлтый)	Горит непрерывно	Коммуникационный модуль проинициализирован модулем центрального процессора и есть связь с модулем центрального процессора
	Мигает	Потеряна связь с модулем центрального процессора
	Не горит	Коммуникационный модуль не проинициализирован модулем центрального процессора
Tx (зелёный)	Не горит	Не осуществляется передача данных по интерфейсу RS-485
	Мигает/горит непрерывно	Осуществляется передача данных по интерфейсу RS-485
Rx (жёлтый или зелёный)	Не горит	Не осуществляется приём данных по интерфейсу RS-485
	Мигает/горит непрерывно	Осуществляется приём данных по интерфейсу RS-485
Поле семисегментных индикаторов (зелёный)	Бегущая строка	Скорость передачи данных интерфейсов RS-485

Табл 17. – Индикаторы коммуникационного модуля МК-544-040

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Модуль не запущен
	Горит	Модуль запущен
Err (красный)	Не горит	Модуль исправен
	Горит	Отказ модуля

Табл 18. – Индикаторы коммуникационного модуля МК-546-010

Индикатор	Состояние	Описание
Run (зелёный)	Не горит	Модуль не запущен
	Горит	Модуль запущен
Err (красный)	Не горит	Модуль исправен
	Горит	Отказ модуля
PWL (зелёный)	Не горит	Не поступают данные формата POWERLINK
	Горит	Поступают данные формата POWERLINK
ETH (зелёный)	Не горит	Не поступают данные формата Ethernet
	Горит	Поступают данные формата Ethernet
RING (зелёный)	Не горит	-
	Горит	Работа в режиме кольцевой топологии сети POWERLINK
LINE (зелёный)	Не горит	-
	Горит	Работа в режиме линейной топологии сети POWERLINK

