



Руководство  
пользователя  
Nexto Series CPU  
NX3008

MU214620 Рев. Е 05  
05 мая 2023 года

Никакая часть этого документа не может быть скопирована или воспроизведена в любой форме без предварительного письменного согласия Altus Sistemas de Automação S.A., которая оставляет за собой право вносить изменения без предварительного уведомления.

Согласно действующему законодательству Бразилии, Кодексу защиты прав потребителей, мы предоставляем следующую информацию клиентам, использующим наши продукты, в отношении личной безопасности и помещений.

Оборудование промышленной автоматизации, производимое Altus, отличается прочностью и надежностью благодаря строгому контролю качества, которому оно подвергается. Однако любое электронное промышленное управляющее оборудование (программируемые контроллеры, числовые команды и т. д.) может повредить машины или процессы, управляемые ими, при наличии дефектных компонентов и/или при возникновении ошибки программирования или установки. Это может даже поставить под угрозу жизни людей. Пользователь должен учитывать возможные последствия дефектов и должен предусмотреть дополнительные внешние установки из соображений безопасности. Эта проблема возрастает при первоначальном вводе в эксплуатацию и тестировании.

Оборудование, производимое Altus, не подвергает окружающую среду непосредственно опасности, так как при использовании не выделяет никаких загрязняющих веществ. Однако, что касается утилизации оборудования, важно отметить, что встроенная электроника может содержать материалы, которые при неправильной утилизации могут нанести вред природе. Поэтому рекомендуется, чтобы всякий раз, когда выбрасывается этот тип продукта, он направлялся на заводы по переработке, которые гарантируют надлежащее обращение с отходами.

Очень важно прочитать и понять документацию по продукту, такую как руководства и технические характеристики, перед его установкой или использованием. Примеры и рисунки, представленные в этом документе, предназначены исключительно для иллюстративных целей. Из-за возможных обновлений и улучшений продуктов, Altus не несет ответственности за использование этих примеров и рисунков в реальных приложениях. Их следует использовать только для обучения пользователей и улучшения опыта работы с продуктами и их функциями.

Altus предоставляет гарантию на свое оборудование, как описано в Общих условиях поставки, прилагаемых к коммерческим предложениям.

Altus гарантирует, что их оборудование работает в соответствии с четкими инструкциями, содержащимися в их руководствах и/или технических характеристиках, не гарантируя успеха любого конкретного типа применения оборудования.

Altus не признает никаких других гарантий, прямых или подразумеваемых, в основном, когда конечные клиенты имеют дело со сторонними поставщиками. Запросы дополнительной информации о поставке, характеристиках оборудования и/или любых других услугах Altus должны быть сделаны в письменной форме. Altus не несет ответственности за предоставление информации о своем оборудовании без официального запроса. Эти продукты могут использовать технологию EtherCAT® ([www.ethercat.org](http://www.ethercat.org)).

#### **АВТОРСКИЕ ПРАВА**

Nexto, MasterTool, Grano и WebPLC являются зарегистрированными товарными знаками Altus Sistemas de Automação S.A. Windows, Windows NT и Windows Vista являются зарегистрированными товарными знаками Microsoft Corporation.

#### **УВЕДОМЛЕНИЕ О ПРОГРАММНОМ ОБЕСПЕЧЕНИИ ОТКРЫТОГО ИСТОЧНИКА**

Чтобы получить исходный код под GPL, LGPL, MPL и другими лицензиями с открытым исходным кодом, который содержится в этом продукте, обращайтесь по адресу [opensource@altus.com.br](mailto:opensource@altus.com.br). В дополнение к исходному коду все упомянутые условия лицензии, отказ от гарантий и уведомления об авторских правах могут быть раскрыты по запросу.

# Содержание

1.	Введение.....	1
1.1.	Серии Nexto .....	1
1.2.	Инновационные особенности.....	3
1.3.	Документы, относящиеся к данному руководству.....	5
1.4.	Визуальный осмотр.....	7
1.5.	Техническая поддержка.....	7
1.6.	Предупреждающие сообщения, используемые в данном руководстве .....	7
2.	Техническое описание .....	8
2.1.	Панели и соединения .....	8
2.2.	Особенности продукта.....	9
2.2.1.	Основные особенности.....	9
2.2.2.	Память.....	12
2.2.3.	Протоколы .....	13
2.2.4.	Серийный интерфейс.....	14
2.2.4.1.	COM 1 .....	14
2.2.5.	CAN-интерфейс.....	14
2.2.6.	USB интерфейс.....	15
2.2.7.	Ethernet-интерфейсы.....	15
2.2.7.1.	NET 1 15	
2.2.7.2.	NET 2 16	
2.2.7.3.	NET 3 16	
2.2.8.	Интерфейс карты памяти.....	17
2.2.9.	Источник питания .....	17
2.2.10.	Характеристики окружающей среды .....	18
2.3.	Совместимость с другими продуктами.....	19
2.4.	Производительность .....	19
2.4.1.	Время применения .....	19
2.4.2.	Время выполнения инструкций.....	19
2.4.3.	Время инициализации .....	20
2.4.4.	Интервал времени .....	20
2.5.	Физические размеры.....	20
2.6.	Данные о покупке .....	20
2.6.1.	Включенные пункты.....	20
2.6.2.	Код продукта .....	21
2.7.	Связанные продукты.....	21
3.	Установка .....	22
3.1.	Механическая установка .....	22
3.2.	Электрическая установка.....	22
3.3.	Сетевое соединение Ethernet.....	24
3.3.1.	IP Адрес.....	24
3.3.2.	Бесплатный ARP .....	24
3.3.3.	Установка сетевого кабеля.....	24
3.4.	Последовательное сетевое соединение RS-485 и CAN .....	26

3.5. Установка карты памяти .....	26
3.6. Установка архитектуры .....	28
3.6.1. Установка модуля в стойку основной объединительной платы.....	28
3.7. Установка программатора .....	28
4. Первоначальное программирование.....	29
4.1. Организация памяти и доступ.....	29
4.2. Профили проектов .....	30
4.2.1. Единый профиль проекта.....	31
4.2.2. Базовый профиль проекта .....	32
4.2.3. Обычный профиль .....	32
4.2.4. Экспертные задачи профиля .....	32
4.2.5. Пользовательский профиль проекта .....	33
4.2.6. Профиль машины.....	34
4.2.7. Основная таблица.....	34
4.2.8. Максимальное количество заданий.....	35
4.3. Конфигурация процессора .....	36
4.4. Библиотеки.....	37
4.5. Вставка экземпляра протокола .....	37
4.5.1. MODBUS RTU .....	37
4.5.2. MODBUS Ethernet.....	39
4.6. Поиск устройства .....	41
4.7. Логин .....	43
4.8. Режим работы .....	45
4.9. Режим остановки.....	46
4.10. Запись и принудительное использование переменных .....	46
4.11. Выход .....	47
4.12. Загрузка проекта.....	47
4.13. Рабочие состояния процессора .....	49
4.13.1. Запуск .....	49
4.13.2. Остановка.....	49
4.13.3. Точка останова .....	49
4.13.4. Исключение .....	49
4.13.5. Сброс настройки тепла .....	49
4.13.6. Сброс настройки холода.....	49
4.13.7. Сбросить происхождение.....	49
4.14. Программы (POU) и списки глобальных переменных (GVL) .....	49
4.14.1. Программа MainPrg .....	49
4.14.2. Программа StartPrg .....	50
4.14.3. Программа UserPrg .....	50
4.14.4. Система GVL_ Diagnostics .....	50
4.14.5. Отключение GVL .....	51
4.14.6. GVL IOQualities.....	52
4.14.7. Module_Diagnostics GVL .....	53
4.14.8. Качества GVL .....	54
4.14.9. GVL ReqDiagnostics .....	55
5. Конфигурация.....	57
5.1. Конфигурация ЦП.....	57
5.1.1. Общие параметры .....	57

5.1.1.1. Горячая замена .....	59
5.1.1.1.1. Горячая замена отключена, только для заявленных модулей .....	59
5.1.1.1.2. Горячая замена отключена .....	59
5.1.1.1.3. Горячая замена отключена, без согласованности при запуске .....	59
5.1.1.1.4. Включена горячая замена, с согласованностью запуска только для заявленных модулей .....	60
5.1.1.1.5. Горячая замена включена с согласованностью запуска .....	60
5.1.1.1.6. Горячая замена включена без согласованности при запуске .....	60
5.1.1.1.7. Как сделать горячую замену .....	60
5.1.1.2. Сохраняемые и постоянные области памяти .....	62
5.1.1.3. Параметры проекта .....	63
5.1.1.4. Параметры CAN Port .....	64
5.1.2. Конфигурация внешнего события .....	64
5.1.3. Синхронизация времени .....	66
5.1.3.1. SNTP .....	67
5.1.3.2. Летнее время (DST) .....	68
5.1.4. Внутренние точки .....	68
5.1.4.1. Качественные конверси .....	70
5.1.4.1.1. Внутреннее качество .....	70
5.1.4.1.2. Внутреннее качество MODBUS .....	71
5.1.4.1.3. Модули качество ввода/вывода локальной шины .....	72
5.1.4.1.4. Качество модулей ввода/вывода PROFIBUS .....	72
5.1.4.1.5. Качество цифровых входов PROFIBUS .....	73
5.1.4.1.6. Качество цифрового вывода PROFIBUS .....	74
5.1.4.1.7. Качество аналоговых входов PROFIBUS .....	75
5.1.4.1.8. Качество аналоговых выходов PROFIBUS .....	76
5.2. Конфигурация последовательных интерфейсов .....	78
5.2.1. COM 1 .....	78
5.2.1.1. Расширенные конфигурации .....	79
5.3. Конфигурация интерфейса ЛСК (Локальная сеть контроллера) .....	80
5.3.1. ЛСК .....	80
5.4. Доступ к веб-странице управления .....	80
5.4.1. Страница системы .....	80
5.4.2. Страница сети .....	81
5.4.1.1. Конфигурация страницы сети .....	81
5.4.1.1.1. Определяется приложением .....	81
5.4.1.1.2. Определяется веб-страницей .....	82
5.4.1.2. Сетевой анализатор трафика .....	83
5.5. Конфигурация интерфейса USB .....	84
5.5.1. Устройство хранения .....	86
5.5.1.1. Общее хранилище .....	86
5.5.1.2. Приложение не загружается при запуске .....	87
5.5.1.3. Перенос приложения с USB-устройства .....	87
5.5.2. Преобразователь USB в RS-232 .....	88
5.5.3. Модемные устройства .....	89
5.5.4. Адаптеры WiFi .....	92
5.6. Конфигурация Интерфейсов Ethernet .....	95
5.6.1. Внутренние интерфейсы Ethernet .....	95

5.6.1.1. NET 1.....	95
5.6.1.2. NET 2.....	96
5.6.1.3. NET 3.....	96
5.6.1.4. Конфигурация внутренних интерфейсов Ethernet .....	96
5.6.1.4.1. Одиночный режим .....	96
5.6.1.4.2. Резервный режим .....	97
5.6.1.4.3. Режим переключения.....	98
5.6.1.4.4. Режим переключения — отключен .....	98
5.6.1.4.5. Режим переключения — RSTP .....	98
5.6.2. Удаленный Ethernet-интерфейс NX5000 .....	99
5.6.2.1. NET 1.....	99
5.6.2.2. Режимы работы удаленного интерфейса Ethernet NX5000.....	99
5.6.2.2.1. Резервный режим .....	99
5.6.3. Зарезервированные порты TCP/UDP .....	99
5.7. Конфигурация протоколов .....	100
5.7.1. Поведение протокола х состояние ЦП.....	102
5.7.2. Очередь событий процессора .....	103
5.7.2.1. Потребители .....	104
5.7.2.2. Принципы работы очереди.....	104
5.7.2.2.1. Знак переполнения.....	104
5.7.2.3. Производители .....	104
5.7.3. Перехват команд, поступающих из Центра управления .....	105
5.7.4. Ведущее устройство MODBUS RTU.....	109
5.7.4.1. Конфигурация главного протокола MODBUS с помощью символьного отображения.....	110
5.7.4.1.1. Основные параметры основного протокола MODBUS — символическое отображение Конфигурация.....	110
5.7.4.1.2. Конфигурация устройств — Конфигурация символического отображения.....	113
5.7.4.1.3. Конфигурация сопоставлений – Настройки символического сопоставления.....	114
5.7.4.1.4. Конфигурация запросов — Настройки символического сопоставления.....	115
5.7.4.2. Конфигурация главного протокола MODBUS для прямого представления (%Q) .....	120
5.7.4.2.1. Общие параметры ведущего протокола MODBUS - настройка прямым представлением (%Q) .....	120
5.7.4.2.2. Конфигурация устройств — Конфигурация для прямого представительства (%Q) ..	121
5.7.4.2.3. Конфигурация сопоставлений — конфигурация для прямого представления (%Q) ..	121
5.7.5. Ведомое устройство MODBUS RTU.....	123
5.7.5.1. Конфигурация ведомого протокола MODBUS с помощью символьного отображения .....	124
5.7.5.1.1. Общие параметры ведомого протокола MODBUS – Конфигурация через символическое отображение.....	124
5.7.5.1.2. Конфигурация отношений — настройка символического отображения.....	128
5.7.5.2. Конфигурация ведомого протокола MODBUS через прямое представление (%Q) .....	129
5.7.5.2.1. Общие параметры ведомого протокола MODBUS — настройка через прямое представление (%Q).....	129
5.7.5.2.2. Конфигурация отображений — конфигурация через прямое представление (%Q) ..	130
5.7.6. MODBUS Ethernet.....	132
5.7.7. Ethernet-клиент MODBUS .....	134
5.7.7.1. Конфигурация клиента MODBUS Ethernet через символическое отображение.....	134
5.7.7.1.1. Общие параметры клиентского протокола MODBUS – конфигурация с помощью символьного отображения.....	134

5.7.7.1.2. Конфигурация устройства — Конфигурация через символическое сопоставление .	136
5.7.7.1.3. Конфигурация сопоставлений — конфигурация через символическое сопоставление .....	137
5.7.7.1.4. Конфигурация запросов — конфигурация через символическое сопоставление.....	139
5.7.7.2. Конфигурация клиента MODBUS Ethernet через прямое представление (%Q) .....	143
5.7.7.2.1. Общие параметры клиента протокола MODBUS - конфигурация для прямого представления (%Q).....	144
5.7.7.2.2. Конфигурация устройства — Конфигурация через прямое представление (%Q).....	144
5.7.7.2.3. Сопоставление Конфигурация – Конфигурация через прямое представление (%Q)	145
5.7.7.3. Запуск отношений клиента MODBUS в циклической форме.....	147
5.7.8. Ethernet-сервер MODBUS.....	147
5.7.8.1. Конфигурация протокола Ethernet сервера MODBUS для символического отображения .....	147
5.7.8.1.1. Общие параметры протокола сервера MODBUS – Конфигурация с помощью символьного отображения.....	148
5.7.8.1.2. Диагностика сервера MODBUS – Конфигурация с помощью символьного отображения .....	149
5.7.8.1.3. Конфигурация отображения – Конфигурация через символическое отображение...	151
5.7.8.2. Конфигурация протокола Ethernet сервера MODBUS через прямое представление (%Q) .....	152
5.7.8.2.1. Общие параметры протокола сервера MODBUS – Конфигурация через прямое представление (%Q).....	152
5.7.8.2.2. Сопоставление Конфигурация – Конфигурация через прямое представление (%Q)	154
5.7.9. Сервер Доступа к коммуникационным данным открытой платформы.....	156
5.7.9.1. Создание проекта для связи OPC DA.....	158
5.7.9.2. Настройка ПЛК на сервере OPC DA.....	161
5.7.9.2.1. Импорт конфигурации проекта .....	163
5.7.9.3. Статус связи OPC DA и переменные качества.....	163
5.7.9.4. Ограничения связи с сервером OPC DA.....	165
5.7.9.5. Доступ к данным через клиент OPC DA.....	166
5.7.10. Сервер OPC UA .....	168
5.7.10.1. Создание проекта для связи OPC UA.....	169
5.7.10.2. Типы поддерживаемых переменных.....	170
5.7.10.3. Ограничение количества подключенных клиентов на сервере OPC UA .....	171
5.7.10.4. Ограничение коммуникационных переменных на сервере OPC UA.....	171
5.7.10.5. Настройки шифрования.....	171
5.7.10.6. Основные параметры связи, настроенные в клиенте OPC UA.....	171
5.7.10.6.1. Конечная точка URL.....	172
5.7.10.6.2. Интервал публикации (мс) и интервал выборки (мс).....	172
5.7.10.6.3. Счетчик жизненного цикла и счетчик поддержания активности.....	172
5.7.10.6.4. Размер очереди и удаление самых старых.....	172
5.7.10.6.5. Тип фильтра и тип зоны нечувствительности.....	173
5.7.10.6.6. PublishingEnabled, MaxNotificationsPerPublish и Priority.....	173
5.7.10.7. Доступ к данным через клиент OPC UA.....	173
5.7.11. Мастер EtherCAT .....	175
5.7.11.1. Установка и подключение устройств EtherCAT .....	175
5.7.11.1.1. Скан устройства EtherCAT.....	176
5.7.11.2. Основные настройки EtherCAT .....	176
5.7.11.2.1. Мастер EtherCAT — Общие .....	176

5.7.11.2.2. EtherCAT Master — назначение устройства синхронизации .....	179
5.7.11.2.3. Мастер EtherCAT — Обзор .....	179
5.7.11.2.4. EtherCAT Master — сопоставление входов/выходов .....	179
5.7.11.2.5. EtherCAT Master — объекты IEC .....	179
5.7.11.2.6. EtherCAT Master — вкладки состояния/информации .....	179
5.7.11.3. Конфигурация EtherCAT Slave .....	179
5.7.11.3.1. Ведомое устройство EtherCAT — Общее .....	179
5.7.11.3.2. Ведомое устройство EtherCAT — данные процесса .....	183
5.7.11.3.3. Ведомое устройство EtherCAT — редактирование списка PDO .....	185
5.7.11.3.4. Ведомое устройство EtherCAT — параметры запуска .....	185
5.7.11.3.5. Ведомое устройство EtherCAT — сопоставление ввода-вывода .....	185
5.7.11.3.6. Ведомое устройство EtherCAT — статус и информация .....	186
5.7.12. EtherNet/IP .....	186
5.7.12.1. EtherNet/IP-интерфейс .....	187
5.7.12.2. Конфигурация сканера EtherNet/IP .....	188
5.7.12.2.1. Общие .....	188
5.7.12.2.2. Соединения .....	189
5.7.12.2.3. Сборки .....	191
5.7.12.2.4. Сопоставление ввода/вывода EtherNet/IP .....	191
5.7.12.3. Конфигурация адаптера EtherNet/IP .....	191
5.7.12.3.1. Общая информация .....	192
5.7.12.3.2. Адаптер EtherNet/IP: сопоставление входов/выходов .....	192
5.7.12.4. Конфигурация модуля EtherNet/IP .....	192
5.7.12.4.1. Сборки .....	192
5.7.12.4.2. Модуль EtherNet/IP: сопоставление ввода-вывода .....	193
5.7.13. Менеджер CANOpen .....	193
5.7.13.1. Установка и подключение устройств CANOpen .....	193
5.7.13.2. Конфигурация менеджера CANOpen .....	194
5.7.13.3. Конфигурация ведомого устройства CANOpen .....	195
5.8. Коммуникационная производительность .....	195
5.8.1. Сервер MODBUS .....	195
5.8.1.1. Локальные интерфейсы ЦП .....	196
5.8.1.2. Удаленные интерфейсы .....	197
5.8.2. Сервер OPC-UA .....	197
5.9. Производительность системы .....	197
5.9.1. Время сканирования входа/выхода .....	197
5.9.2. Карта памяти .....	198
5.10. Часы реального времени .....	198
5.10.1. Функциональные блоки для чтения и записи RTC .....	199
5.10.1.1. Функциональные блоки для чтения RTC .....	199
5.10.1.1.1. GetDateAndTime .....	199
5.10.1.1.2. GetTimeZone .....	200
5.10.1.1.3. GetDayOfWeek .....	201
5.10.1.2. Функции записи RTC .....	202
5.10.1.2.1. SetDateAndTime .....	202
5.10.1.2.2. SetTimeZone .....	203
5.10.2. Структура данных RTC .....	204
5.10.2.1. EXTENDED_DATE_AND_TIME .....	204

5.10.2.2. DAYS_OF_WEEK .....	204
5.10.2.3. RTC_STATUS.....	205
5.10.2.4. TIMEZONESETTINGS .....	205
5.11. Память пользовательских файлов .....	205
5.12. Карта памяти .....	207
5.12.1. Подготовка проекта .....	208
5.12.2. Перенос проекта.....	208
5.12.3. Доступ MasterTool.....	210
5.13. Информативность процессора и Конфигурация меню.....	210
5.14. Функциональные блоки и функции .....	212
5.14.1. Специальные функциональные блоки для последовательных интерфейсов .....	212
5.14.1.1. SERIAL_CFG.....	216
5.14.1.2. SERIAL_GET_CFG.....	217
5.14.1.3. SERIAL_GET_RX_QUEUE_STATUS.....	219
5.14.1.4. SERIAL_PURGE_RX_QUEUE .....	221
5.14.1.5. SERIAL_RX.....	222
5.14.1.6. SERIAL_RX_EXTENDED.....	225
5.14.1.7. SERIAL_SET_CTRL .....	227
5.14.1.8. SERIAL_TX .....	229
5.14.2. Обновление входов и выходов .....	231
5.14.2.1. REFRESH_INPUT .....	231
5.14.2.2. REFRESH_OUTPUT .....	232
5.14.3. Функциональный блок PID .....	233
5.14.4. Сохранение таймера .....	234
5.14.4.1. TOF_RET .....	234
5.14.4.2. TON_RET.....	235
5.14.4.3. TP_RET .....	236
5.14.5. Журнал пользователя.....	238
5.14.5.1. UserLogAdd.....	239
5.14.5.2. UserLogDeleteAll .....	240
5.14.6. ClearRtuDiagnostic.....	241
5.14.7. ClearEventQueue .....	241
5.15. Веб страницы пользователей .....	241
5.16. Устройство.....	241
5.16.1. Управление пользователями и права доступа.....	241
5.16.2. Настройки ПЛК.....	242
5.17. Брандмауэр.....	244
5.17.1. Конфигурация .....	244
5.17.2. Общая конфигурация.....	244
5.17.3. Правила пользователя.....	246
5.18. OpenVPN .....	248
5.18.1. Импорт Конфигурация .....	248
5.18.2. Конфигурация OpenVPN.....	249
5.18.2.1. Общие конфигурации .....	250
5.18.2.1.1. Режим .....	250
5.18.2.1.2. Протокол.....	250
5.18.2.1.3. Уровень журналов.....	250
5.18.2.1.4. Пакет проверки активности .....	250

5.18.2.1.5. Тайм-аут поддержания активности .....	250
5.18.2.1.6. Файлы безопасности .....	250
5.18.2.1.7. Ключ ТА.....	251
5.18.2.2. Эксклюзивная конфигурация сервера.....	251
5.18.2.2.1. Сетевой адрес .....	251
5.18.2.2.2. Общение между клиентами.....	251
5.18.2.2.3. Максимальное количество подключенных клиентов.....	251
5.18.2.2.4. Частные сети.....	251
5.18.2.3. Эксклюзивные клиентские конфигурации .....	253
5.18.2.3.1. Удаленный IP.....	253
5.18.2.4. Настройки приложения .....	253
5.18.3. Файлы безопасности .....	253
5.18.4. Таблица статуса.....	254
5.18.5. Файлы для загрузки .....	256
5.18.6. Конфигурация архитектуры.....	256
5.18.6.1. Хост-хост .....	257
5.18.6.2. Хост объект.....	257
5.18.6.3. Объект-объект .....	258
6. Обслуживание.....	259
6.1. Диагностика модуля .....	259
6.1.1. Диагностика одним касанием .....	259
6.1.2. Диагностика через светодиод .....	261
6.1.2.1. DG (диагностический) .....	261
6.1.2.2. WD (Таймер контрольной системы) .....	262
6.1.2.3. Индикаторы разъема RJ45.....	262
6.1.3. Диагностики через ВЕБ.....	262
6.1.4. Обозреватель диагностики .....	264
6.1.5. Диагностика через переменные .....	265
6.1.5.1. Обобщенная диагностика.....	265
6.1.5.2. Детальная диагностика.....	269
6.1.6. Диагностика через функциональные блоки .....	283
6.1.6.1. GetTaskInfo .....	283
6.2. Графический экран.....	284
6.3. Системный журнал.....	286
6.4. Не загружается приложение при запуске.....	286
6.5. Общие проблемы.....	286
6.6. Поиск неисправностей .....	287
6.7. Профилактика .....	287
7. Приложения .....	288
7.1. Управление ключами TLS и сертификатами.....	288
7.1.1. Создание сертификата Easy-RSA .....	288
7.1.2. Создание сертификата OpenSSL.....	292
7.1.3. Генерация ключей ТА с помощью OpenVPN.....	294

## 1. Введение

Программируемые контроллеры серии Nexto — идеальное решение для промышленной автоматизации и системного управления. Благодаря высокотехнологичным встроенным технологиям продукты семейства способны управлять распределенным и избыточным образом сложными промышленными системами, машинами, высокопроизводительными производственными линиями и самыми передовыми процессами Индустрии 4.0. Современная и высокоскоростная серия Nexto использует передовые технологии для обеспечения надежности и возможности подключения, помогая повысить производительность различных предприятий.

Компактные, прочные и с высокой доступностью продукты этой серии обладают превосходной производительностью обработки и возможностями расширения стойки. Его архитектура обеспечивает простую интеграцию с системами наблюдения, управления и полевыми сетями, а также с резервированием ЦП и источника питания. Оборудование семейства также предлагает расширенную диагностику и горячую замену, сводя к минимуму или исключая время простоя на техническое обслуживание и обеспечивая непрерывный производственный процесс.

Благодаря мощному 64-разрядному процессору ARM с тактовой частотой 1 ГГц процессор NX3008 идеально подходит для управления малыми и большими промышленными машинами и процессами. Помимо расширенной диагностики и разнообразия консолидированных протоколов связи, имеет ресурсы кибербезопасности, межсетевой экран, удаленное управление и возможность настройки приложения пользователя с установкой внешних программ, благодаря платформе Docker и используемой операционной системе Linux на всех контроллерах серии Nexto.



Рисунок 1: NX3008

### 1.1. Серии Nexto

Серия Nexto — это мощная и полная серия программируемых контроллеров (ПЛК) с эксклюзивными и инновационными характеристиками. Благодаря своей гибкости, функциональному дизайну, расширенным диагностическим ресурсам и модульной архитектуре ПЛК Nexto можно использовать для управления системами в малых, средних и крупных приложениях.

Архитектура серии Nexto имеет большое разнообразие модулей ввода и вывода. Эти модули в сочетании с мощным 32-битным процессором и высокоскоростной шиной на основе Ethernet подходят для нескольких типов приложений, таких как высокоскоростное управление небольшими машинами, сложные распределенные процессы, резервные приложения и системы с большим количеством операций ввода-вывода, такие как автоматизация зданий. Кроме того, в серии Nexto есть модули для управления движением, коммуникационные модули, охватывающие, среди прочего, самые популярные полевые сети.

Серия Nexto использует в своей шине передовую технологию, основанную на высокоскоростном интерфейсе Ethernet, позволяющую обмениваться входной и выходной информацией и данными между несколькими контроллерами внутри одной системы. Систему можно легко разделить и распределить по всему полю, что позволяет использовать шинное расширение с той же производительностью, что и локальный модуль, что делает возможным использование каждого модуля в локальной раме или в рамках расширения без ограничений. Для

## 1. ВВЕДЕНИЕ

---

соединения между модулями расширения используется простой стандартный кабель Ethernet.



Рисунок 2: Серия Nexto — Обзор

### 1.2. Инновационные особенности

Серия Nexto предлагает пользователю множество инноваций, касающихся использования, контроля и обслуживания системы. Эти функции были разработаны с учетом новой концепции промышленной автоматизации.



**VPN:** Продукты Nexto имеют встроенную службу VPN, которая создает частный туннель, который напрямую подключается к ЦП. Эта функциональность, доступная на некоторых моделях семейства, позволяет удаленно и полностью безопасно получить доступ к сети управления.



**FTP:** Поддерживая соединения типа FTP, серийное оборудование может обмениваться данными с сервером, использующим ту же технологическую модель. Эта функция позволяет удаленно получать доступ к файлам, сгенерированным контроллером, таким как журналы, собранные с помощью функции регистратора данных.



**Linux:** Еще одна инновационная особенность серии — встроенная платформа Linux. Эта функция делает возможной виртуализацию программного обеспечения, разработанного для операционных систем с технологией Unix. Эта функция делает работу системы более универсальной и быстрой, поскольку позволяет обрабатывать несколько данных внутри самого ЦП.



**Работа без батарей:** серия Nexto не требует каких-либо батарей для обслуживания памяти и работы часов реального времени. Эта функция чрезвычайно важна, поскольку она снижает потребность в обслуживании системы и позволяет использовать ее в удаленных местах, где обслуживание может быть затруднено. Кроме того, эта функция является экологически чистой.



**Система Easy Plug:** серия Nexto имеет эксклюзивный метод подключения и отключения клеммных блоков ввода-вывода. Клеммные колодки легко снимаются одним движением и без специальных инструментов. Чтобы подключить клеммную колодку обратно к модулю, передняя крышка облегчает процедуру установки, подгоняя клеммную колодку к модулю.



**Множественное блочное хранилище:** В процессорах серии Nexto пользователю доступны несколько типов памяти, что обеспечивает наилучший вариант для любых нужд пользователя. Эти воспоминания делятся на энергозависимые и энергонезависимые. Для энергозависимой памяти ЦП серии Nexto предлагают адресный вход (%I), адресный выход (%Q), адресуемую память (%M), память данных и резервную память данных. Для приложений, требующих энергонезависимой функциональности, ЦП серии Nexto обеспечивают сохраняемую адресуемую память (%Q), сохраняемую память данных, постоянную адресуемую память (%Q), постоянную память данных, память программ, память исходного кода, файловую систему ЦП (doc, PDF, данные) и интерфейс карты памяти.



**One Touch Diag:** One Touch Diag — это эксклюзивная функция, которую серия Nexто привносит в ПЛК. Благодаря этой новой концепции пользователь может проверять диагностическую информацию о любом модуле, присутствующем в системе, непосредственно на графическом дисплее ЦП одним нажатием диагностического переключателя соответствующего модуля. OTD — это мощный диагностический инструмент, который можно использовать в автономном режиме (без супервизора или программиста), сокращая время обслуживания и ввода в эксплуатацию

**OFD — полная документация на ходу:** ЦП серии Nexто способны хранить полную проектную документацию в собственной памяти. Эта функция может быть очень удобна для целей резервного копирования и обслуживания, поскольку вся информация хранится в одном надежном месте.

**ETD —Электронная метка на дисплее:** Еще одна эксклюзивная функция, которую серия Nexто привносит в ПЛК, — это электронная метка на дисплее. Эта новая функциональность переносит процесс проверки имен тегов любого контакта ввода-вывода или модуля, используемого в системе, непосредственно на графический дисплей ЦП. Наряду с этой информацией пользователь также может проверить описание. Эта функция чрезвычайно полезна при обслуживании и устранении неполадок.

**ГВС-Двойная аппаратная ширина:** Модули серии Nexто были разработаны для экономии места в шкафах пользователя или машинах. По этой причине серия Nexто предлагает модули двух разных размеров: двойной ширины (требуется два слота для объединительной платы) и одинарной ширины (требуется только один слот для объединительной платы). Эта концепция позволяет использовать компактные модули ввода/вывода с высокой плотностью точек ввода/вывода вместе со сложными модулями, такими как ЦП, ведущие устройства полевой шины и модули электропитания.

**Высокоскоростной ЦП:** Все ЦП серии Nexто были разработаны для обеспечения выдающейся производительности для пользователя, что позволяет удовлетворить широкий спектр требований приложений.

### 1.3. Документы, относящиеся к данному руководству

Чтобы получить дополнительную информацию о серии Nexto, кроме этого, можно получить доступ к другим документам (руководствам и техническим характеристикам). Эти документы доступны в последней версии на сайте <https://www.altus.com.br/en/>.

Каждый продукт имеет документ, разработанный техническими характеристиками (CE), в котором описаны функции продукта. Кроме того, продукт может иметь руководства по эксплуатации (коды руководств указаны в CE). Например, модуль NX2020 содержит информацию об использовании функций и покупке на своем CE. С другой стороны, NX5001 имеет, помимо CE, руководство пользователя (MU).

В качестве дополнительного источника информации рекомендуется использовать следующие документы:

Код	Описание	Язык
<b>CE114000</b> <b>CT114000</b> <b>CS114000</b>	Nexto Series – технические характеристики Série Nexto – Características Técnicas Serie Nexto – Características Técnicas	английский португальский испанский
<b>CE114109</b> <b>CT114109</b> <b>CS114109</b>	NX3008 технические характеристики Características Técnicas NX3008 Especificaciones y Configuraciones NX3008	английский португальский испанский
<b>CE114700</b> <b>CT114700</b> <b>CS114700</b>	Технические характеристики стоек объединительной платы серии Nexto Características Técnicas dos Bastidores da Série Nexto Características Técnicas de los Bastidores de la Serie Nexto	английский португальский испанский
<b>CE114810</b> <b>CT114810</b> <b>CS114810</b>	Технические характеристики для аксессуаров серии Nexto для стоек объединительных плат Características Técnicas Acessórios para Bastidor Série Nexto Características Técnicas del Cierres Laterales para el Bastidor	английский португальский испанский
<b>CE114902</b> <b>CT114902</b> <b>CS114902</b>	Технические характеристики PROFIBUS-DP Master серии Nexto Características Técnicas do Mestre PROFIBUS-DP da Série Nexto Características Técnicas del Módulo Profibus-DP Maestro	английский португальский испанский
<b>CE114908</b> <b>CT114908</b> <b>CS114908</b>	Технические характеристики головок NX5110 и NX5210 PROFIBUS-DP Características Técnicas Interfaces Cabeça PROFIBUSDP NX5110 e NX5210 Especificaciones y Configuraciones PROFIBUS-DP Interfaz Cabezas NX5110 y NX5210	английский португальский испанский
<b>MU214600</b> <b>MU214000</b>	Руководство пользователя серии Nexto Manual de Utilização Série Nexto	английский португальский
<b>MU214620</b> <b>MU214109</b>	Руководство пользователя процессора NX3008 Manual de Utilização UCP NX3008	английский португальский
<b>MU299609</b>	MasterTool IEC XE User Manual	английский

Код	Описание	Язык
MU299048	Manual de Utilização MasterTool IEC XE	португальский
MP399609 MP399048	Руководство по программированию MasterTool IEC XE Manual de Programação MasterTool IEC XE	английский португальский
MU214601 MU214001	Руководство пользователя NX5001 PROFIBUS DP Master Manual de Utilização Mestre PROFIBUS-DP NX5001	английский португальский
MU214608 MU214108	Руководство по использованию головки Nexto PROFIBUS-DP Manual de Utilização da Cabeça PROFIBUS-DP Nexto	английский португальский
MU214603	Руководство по протоколу HART для серии Nexto	английский
MU214610	Руководство пользователя по расширенным функциям управления	английский
NAP151	Utilização do Tunneller OPC	португальский
NAP165 NAP165_ing	Comunicação OPC UA com Controladores ALTUS Связь OPC UA с контроллерами ALTUS	португальский английский

Таблица 1: Связанные документы

### 1.4. Визуальный осмотр

Перед возобновлением процесса монтажа рекомендуется внимательно визуально осмотреть оборудование, убедившись в наличии транспортных повреждений. Убедитесь, что все запрошенные детали находятся в идеальном состоянии. В случае повреждения сообщите об этом ближайшей к вам транспортной компании или дистрибьютору

#### ПРЕДУПРЕЖДЕНИЕ

Перед снятием модулей с корпуса важно снять возможную статическую энергию, накопившуюся в корпусе. Для этого прикоснитесь (голыми руками) к любой заземленной металлической поверхности, прежде чем брать за модули. Такая процедура гарантирует, что пределы статической энергии модуля не будут

Altus.

Важно регистрировать каждый полученный серийный номер оборудования, а также версии программного обеспечения, если они существуют. Эта информация необходима на случай обращения в службу технической поддержки Altus.

### 1.5. Техническая поддержка

Для обращения в службу технической поддержки Altus в Сан-Леопольдо, RS, позвоните по телефону +55 51 3589-9500. Для получения дополнительной информации о технической поддержке Altus, существующей в других местах, посетите <https://www.altus.com.br/en/> или отправьте электронное письмо по адресу [altus@altus.com.br](mailto:altus@altus.com.br).

Если оборудование уже установлено, на момент обращения в службу поддержки у вас должна быть следующая информация:

- Модель используемого оборудования и конфигурация установленной системы
- Серийный номер продукта
- Ревизия оборудования и версия исполнительного ПО, написанные на бирке, закрепленной на стороне продукта. Информация о режиме работы ЦП, полученная через MasterTool IEC XE.
- Содержимое прикладного программного обеспечения, полученное с помощью MasterTool IEC XE
- Используемая версия программатора

### 1.6. Предупреждающие сообщения, используемые в данном руководстве

В данном руководстве предупреждающие сообщения будут представлены в следующих форматах и значениях:

#### ОПАСНО

Сообщает о потенциальной опасности, которая, если ее не обнаружить, может нанести вред людям, материалам, окружающей среде и производству.

#### ПРЕДУПРЕЖДЕНИЕ

Сообщает о конфигурации, приложении или деталях установки, которые необходимо учитывать, чтобы избежать любого случая, который может привести к сбою системы и последующему воздействию.

#### ВНИМАНИЕ

Определяет детали конфигурации, приложения и установки, направленные на достижение максимальной производительности системы.

## 2. Техническое описание

В этой главе представлены все технические характеристики процессора Nexto Series NX3008.

### 2.1. Панели и соединения

На следующем рисунке показана передняя панель ЦП.



Рисунок 3: NX3008

Как видно на рисунке, в верхней части лицевой панели расположен графический дисплей, на котором отображается состояние всей системы и диагностика, включая конкретную диагностику каждого модуля. Графический дисплей также предлагает простое в использовании меню, которое предоставляет пользователю быстрый режим для чтения или определения параметров, таких как: внутренняя температура (только чтение) и местное время (только чтение).

Непосредственно под графическим дисплеем расположены 2 светодиода, которые используются для индикации диагностики сигнализации и схемы сторожевого устройства. В таблице ниже показано описание светодиодов. Дополнительную информацию о состоянии и значении светодиодов см. в разделе «Диагностика с помощью светодиодов».

LED	Описание
DG	Светодиод диагностики
WD	Сторожевой светодиод

Таблица 2: Описание светодиодов

ЦП серии Nexto имеет два переключателя, доступных пользователю. В таблице ниже показано описание этих переключателей. Дополнительную информацию о диагностическом переключателе см. в разделах One Touch Diag.

Ключи	Описание
Переключатель диагностики	Переключатель расположен в верхней части модуля. Используется для визуализации диагностики на графическом дисплее или для навигации по информативному меню и конфигурации ЦП.

Таблица 3: Описание ключей

На передней панели доступны интерфейсы подключения процессоров серии Nexto. В таблице ниже представлено краткое описание этих интерфейсов.

Интерфейсы	Описание
<b>NET 1.NET 3</b>	Разъем типа RJ45 для стандартной связи 10/100/1000Base-TX. Обеспечивает двухточечную или сетевую связь. Дополнительные сведения об использовании см. в разделе Конфигурация интерфейсов Ethernet.
<b>COM 1</b>	Дополнительные сведения об использовании см. в разделе «Конфигурация последовательных интерфейсов».
<b>CAN</b>	Дополнительные сведения об использовании см. в разделе «Конфигурация интерфейса CAN (локальная сеть контроллеров)».
<b>MEMORY SLOT</b>	Разъем карты памяти. Позволяет использовать карту памяти для хранения различных типов данных, таких как журналы пользователей, веб-страницы, проектная документация и исходные файлы. Дополнительные сведения об использовании см. в разделе Карта памяти.
<b>USB HOST</b>	Дополнительные сведения об использовании см. в разделе «Конфигурация USB-интерфейса».

Таблица 4: Интерфейсы подключения

## 2.2. Особенности продукта

### 2.2.1. Основные особенности

	<b>NX3008</b>
<b>Занятие стойки объединительной платы</b>	2 последовательных слота
<b>Встроенный блок питания</b>	Да
<b>Локальный интерфейс Ethernet TCP/IP</b>	3
<b>Последовательный интерфейс</b>	1

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

<b>CAN-интерфейс</b>	1
<b>Хост USB-порта</b>	1
<b>Интерфейс карты памяти</b>	1
<b>Часы реального времени (RTC)</b>	Да Разрешение 1 мс и максимальное отклонение 2 с в день.
<b>Таймер контрольной системы</b>	Да
<b>Состояние и диагностическая индикация</b>	Светодиоды графического дисплея веб-страница внутренняя память процессора
<b>Языки программирования</b>	Структурированный текст (ST) Лестничная диаграмма (LD) Последовательная функциональная схема (SFC) Диаграмма функциональных блоков (FBD) Непрерывная функциональная схема (CFC)
<b>Задания</b>	Циклический (периодический) Событие (программное прерывание) Внешнее (аппаратное прерывание) Случайное (непрерывное) Состояние (программное прерывание)
<b>Онлайн изменения</b>	Да
<b>Максимальное количество задач</b>	24
<b>Максимальное количество шин расширения</b>	24
<b>Поддержка резервирования расширения шины</b>	Да
<b>Максимальное количество модулей ввода/вывода на шине</b>	128
<b>Максимальное количество дополнительных Ethernet</b>	2
<b>Интерфейсные модули TCP/IP</b>	Да
<b>Поддержка резервирования интерфейса Ethernet TCP/IP</b>	4
<b>Максимальное количество сетей PROFIBUS-DP (с использованием ведущих модулей PROFIBUS-DP)</b>	Да
<b>Поддержка резервирования сети PROFIBUS-DP</b>	Нет
<b>Поддержка резервирования (полукластеры)</b>	Да
<b>Протокол событийно-ориентированной отчетности (SOE)</b>	Нет - -
<b>Максимальный размер очереди событий</b>	
<b>Веб-страницы пользователей (Webvisu)</b>	Да
<b>Брандмауэр</b>	Да
<b>Докер</b>	В процессе




NX3008	
Диагностика одним касанием (OTD)	Да
Электронная бирка на дисплее (ETD)	Да
<b>Стандарты и сертификации</b> <b>IEC 61131-2</b> <b>IEC 61131-3</b> <b>DNV</b>  	Да
	Да
	В процессе Тип утверждения DNV – DNV-CG-0339 (TAA000013D)
	Да CE – 2014/35/EU (LVD) и 2014/30/EU (EMC)
	Да RoHS – 2011/65/EU
	Да UL/cUL Listed - UL 61010-1 (file E473496)
	В процессе EAC – CU TR 004/2011 (LVD) и CU TR 020/2011 (EMC)

Таблица 5: Общие особенности

**Примечания:**

Часы реального времени (RTC): Время хранения, время, в течение которого часы реального времени будут продолжать обновлять дату и время после отключения питания ЦП, составляет 15 дней для работы при температуре 25 °C. При максимальной температуре продукта время выдержки сокращается до 10 дней.

Максимальное количество модулей ввода/вывода на шине: Максимальное количество модулей ввода/вывода относится к сумме всех модулей на локальной шине и модулей расширения.

## 2.2.2. Память

	NX3008
Память адресуемых входных переменных (%I)	64 Кбайт
Память адресуемых выходных переменных (%Q)	64 Кбайт
Память переменных прямого представления (%M)	32 Кбайт
Память символьных переменных	12 Мбайт
Максимальный объем памяти, настраиваемый как сохраняемый или постоянный	1 Мбайт
Полная избыточная память данных Память входных переменных прямого представления (%I) Память выходных переменных прямого представления (%Q) Память переменных прямого представления (%M) Память символьных переменных	- - - -
Общая память Память программы (ограничена 32 МБ) + Память исходного кода (резервная копия) + Память файлов Webvisu	256 Мбайт
Память пользовательских файлов	2 Гбайт

Таблица 6: Память

**Примечание:**

Память адресуемых входных переменных (%I): Область, в которой хранятся адресуемые входные переменные. Адресуемые переменные означают, что к переменным можно обращаться напрямую, используя желаемый адрес. Например: %IW0, %IW100. Адресуемые входные переменные можно использовать для сопоставления цифровых или аналоговых входных точек. Для справки, 8 цифровых входов могут быть представлены на один байт, а одна точка аналогового входа может быть представлена на два байта.

Общая память адресуемых выходных переменных (%Q): Область, в которой хранятся адресуемые выходные переменные. Адресные переменные означают, что к переменным можно получить прямой доступ, используя желаемый адрес. Например: %QB0, %QW100. Адресные выходные переменные можно использовать для отображения цифровых или аналоговых выходных точек. Для справки, 8 цифровых выходов могут быть представлены на один байт, а одна точка аналогового выхода может быть представлена на два байта. Адресуемые выходные переменные могут быть определены как сохраняемые, постоянные или избыточные переменные, но общий размер не изменяется из-за конфигурации.

Память адресуемых переменных (%M): Область, в которой хранятся переменные адресуемых маркеров. Адресуемые переменные означают, что к переменным можно обращаться напрямую, используя желаемый адрес. Например: %MB0, %MW100.

Память символьных переменных: Область, в которой размещаются символьные переменные. Символические переменные — это переменные IEC, созданные в ROU и GVL во время разработки приложения, которые не адресуются непосредственно в памяти. Символьные переменные могут быть определены как сохраняемые или постоянные, и в этом случае будут использоваться области памяти сохраняемых символьных переменных или память постоянных символьных переменных соответственно. Система ПЛК размещает переменные в этой области, поэтому пространство, доступное для размещения переменных, созданных пользователем, меньше, чем указано в Таблице. Занятость системных переменных зависит от характеристик проекта (количество модулей, драйверов и т. д.), поэтому рекомендуется соблюдать свободное место в сообщениях компиляции инструмента MasterTool IEC XE.

Постоянная или сохраняемая память символьных переменных: Область, в которой размещаются сохраняемые символьные переменные. Сохраняемые данные сохраняют свои соответствующие значения даже после циклов выключения и включения ЦП. Постоянные данные сохраняют свои соответствующие значения даже после загрузки

**ВНИМАНИЕ**

Объявление и использование символических постоянных переменных должно выполняться исключительно через объект Persistent Variables, который может быть включен в проект через древовидное представление в Application -> Add Object -> Persistent Variables. Его не следует использовать для выражения VAR PERSISTENT в объявлении переменных поля программных модулей.

нового приложения в ЦП.

Полный список случаев, когда символические постоянные переменные сохраняют свои значения и когда значение теряется, можно найти в Таблице ниже. Помимо размера персистентной области, объявленного в Таблице выше, эти 44 байта зарезервированы для хранения информации о персистентных переменных (недоступных для

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

использования).

В таблице ниже показано поведение сохраняемых и постоянных переменных для различных ситуаций, в которых «-» означает, что значение потеряно, а «X» означает, что значение сохранено.

Команда/операция	VAR	VAR RETAIN	VAR PERSISTENT
Цикл питания	-	X	X
Теплый сброс	-	X	X
Сбросить холод	-	-	X
Сбросить происхождение	-	-	-
Извлеките ЦП со встроенным блоком питания из стойки при включенном питании.	-	X	X
Извлеките источник питания или ЦП без встроенного источника питания из стойки при включенном питании.	-	-	-
Скачать	-	-	X
Онлайн изменение	X	X	X
Очистить все	-	-	X
Процесс сброса (IEC 60870-5-104)	-	X	X

Таблица 7: Поведение переменных после события

В случае команды «Очистить все», если приложение было изменено таким образом, что постоянные переменные были удалены, вставлены в начало списка или иным образом имели измененный тип, значение этих переменных теряется (при запросе инструмента MasterTool для загрузки). Поэтому рекомендуется, чтобы изменения постоянных переменных GVL включали только добавление новых переменных в список.

Память программы: Память программы — это максимальный размер, который можно использовать для хранения пользовательского приложения. Эта область используется совместно с памятью исходного кода, являясь общей площадью (память программы + память исходного кода + память компиляции).

Память исходного кода (резервная копия): Эта область памяти используется в качестве резервной копии проекта. Если пользователь хочет импортировать проект, MasterTool IEC XE получит необходимую информацию в этой области. Необходимо следить за актуальностью проекта, сохраненного в качестве резервной копии, чтобы избежать потери важной информации. Эта область используется совместно с памятью исходного кода, являясь общей площадью (память программы + память исходного кода).

Память пользовательских файлов: эта область памяти предлагает пользователю другой способ хранения таких файлов, как doc, pdf, изображения и другие файлы. Эта функция позволяет записывать данные, как на карту памяти. Для получения дополнительной информации см. раздел «Память пользовательских файлов».

### 2.2.3. Протоколы

	NX3008	Интерфейс
Связь с программным обеспечением	Да	NET 1 / NET 2 / NET 3 / USB
Открытый протокол	Да	COM 1 / USB
Ведущее устройство MODBUS RTU	Да	COM 1
Ведомое устройство MODBUS RTU	Да	COM 1
MODBUS-TCP-клиент	Да	NET 1 / NET 2 / NET 3
MODBUS TCP-сервер	Да	NET 1 / NET 2 / NET 3
MODBUS RTU через TCP-клиент	Да	NET 1 / NET 2 / NET 3
MODBUS RTU через TCP-сервер	Да	NET 1 / NET 2 / NET 3
Мастер CANopen	Да	CAN
Ведомое устройство CANopen	Нет	-
CAN низкий уровень	Да	CAN
CAE J-1939	Да	CAN
OPC-DA-сервер	Да	NET 1 / NET 2 / NET 3
OPC-UA-сервер	Да	NET 1 / NET 2 / NET 3

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Мастер EtherCAT	Да	NET 1 / NET 2 / NET 3
SNMP-агент	Нет	-
SOE (событийные данные)	Нет	-

	NX3008	Интерфейс
Сервер МЭК 60870-5-104	В процессе	-
Сканер EtherNet/IP	Да	NET 1 / NET 2 / NET 3
EtherNet/IP-адаптер	Да	NET 1 / NET 2 / NET 3
MQTT-клиент	Да	NET 1 / NET 2 / NET 3 / USB
SNTP-клиент (для синхронизации часов)	Да	NET 1 / NET 2 / NET 3 / USB
PROFINET-контроллер	Да	NET 1 / NET 2 / NET 3
PROFINET-устройство	Нет	-
OpenVPN-клиент	Да	NET 1 / NET 2 / NET 3
OpenVPN-сервер	Да	NET 1 / NET 2 / NET 3
FTP	В процессе	-
ППМ	В процессе	-

Таблица 8: Протоколы

### Примечание:

**USB:** Необходимо использовать последовательный, WiFi или модемный адаптер.

Связь с программным обеспечением для программирования: Для связи с ЦП через интерфейс, отличный от NET 1, необходимо добавить шлюз с IP-адресом данного интерфейса.

### 2.2.4. Серийный интерфейс

#### 2.2.4.1. COM 1

	COM 1
Соединитель	Клеммная колодка, D+ и D- с экраном
Физический интерфейс	РС-485
Направление связи	полудуплекс
RS-485 макс. трансиверы	32
Прекращение	Да (необязательный через параметр)
Поперечное сечение	0.5 мм <sup>2</sup>
Скорость передачи данных	2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с
Изоляция	
Логика для последовательного порта	1000 В перем.тока / 1 минута
<b>Последовательный порт для защиты заземления</b>	1000 В перем.тока / 1 минута

Таблица 9: Характеристики последовательного интерфейса COM 1

### Примечание:

**RS-485 максимальное количество приемопередатчиков:** это максимальное количество интерфейсов RS-485, которые можно использовать на одной шине.

### 2.2.5. CAN-интерфейс

	CAN
Соединитель	Клеммная колодка, H и L с экраном
Физический интерфейс	CAN-шина
Поддерживаемые стандарты	CAN 2.0A 2.0B (11-битные и 29-битные идентификаторы)
Макс. количество узлов	64
Прекращение	Да (Настраиваемый)
Поперечное сечение	0.5 мм <sup>2</sup>
Скорость передачи данных	10, 20, 50, 100, 125, 250, 500, 1000 кбит/с

<b>Изоляция</b> <b>Логика для CAN</b> <b>CAN к защитному</b> <b>заземлению</b> ⚡	1000 В перем.тока / 1 минута 1000 В перем.тока / 1 минута
---	--

Таблица 10: Особенности интерфейса CAN

**2.2.6. USB интерфейс**

	<b>USB</b>
<b>Соединитель</b>	USB Женский
<b>Физический интерфейс</b>	USB версии 2.0
<b>Скорость передачи данных</b>	1,5 Мбит/с (низкая скорость), 12 Мбит/с (полная скорость) и 480 Мбит/с (высокая скорость)
<b>Максимальный ток</b>	500 мА
<b>Поддерживаемые устройства</b>	Запоминающее устройство USB RS-232 Последовательный преобразователь USB 3G/4G модем USB WiFi адаптер
<b>Изоляция</b> <b>Логика на USB</b> <b>USB для защитного</b> <b>заземления</b> ⚡	Неизолированный 1000 В переменного тока / 1 минута

Таблица 11: Особенности интерфейса USB

**Примечания:**

Последовательный преобразователь USB RS-232: см. список поддерживаемых устройств в разделе Преобразователь USB в RS-232.

USB-модем 3G/4G: см. список поддерживаемых устройств в разделе «Модемные устройства».

USB-адаптер WiFi: см. список поддерживаемых устройств в разделе «Адаптеры WiFi».

**2.2.7. Ethernet-интерфейсы****2.2.7.1. NET 1**

	<b>NET 1</b>
<b>Соединитель</b>	Экранированная розетка RJ45
<b>Авто кроссовер</b>	Да
<b>Максимальная длина кабеля</b>	100 м
<b>Тип кабеля</b>	UTP или ScTP, категория 5
<b>Скорость передачи данных</b>	10/100/1000 Мбит/с
<b>Физический слой</b>	10BASE-Te/100BASE-TX/1000BASE-T
<b>Канальный уровень</b>	LLC (управление логической связью)
<b>Сетевой уровень</b>	IP (Интернет-протокол)
<b>Транспортный уровень</b>	TCP (протокол управления передачей) UDP (протокол пользовательских дейтаграмм)
<b>Диагностики</b>	Светодиод – зеленый 1000 Мбит/с (соединение/активность) Светодиод – желтый 100 Мбит/с (соединение/активность) Светодиоды – зеленый и желтый 10 Мбит/с (связь/активность)

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

<b>Изоляция</b>	1500 В перем.тока / 1 минута
<b>Ethernet-интерфейс к логике</b>	1500 В перем.тока / 1 минута
<b>Интерфейс Ethernet к интерфейсу Ethernet</b>	1500 В перем.тока / 1 минута
<b>Интерфейс Ethernet для защитного заземления</b>	1500 В перем.тока / 1 минута

Таблица 12: Особенности интерфейса Ethernet NET 1

### 2.2.7.2. NET 2

	NET 2
Соединитель	Экранированная розетка RJ45
Авто кроссовер	Да
Максимальная длина кабеля	100 м
Тип кабеля	UTP или ScTP, категория 5
Скорость передачи данных	10/100/1000 Мбит/с
Физический слой	10BASE-Te/100BASE-TX/1000BASE-T
Канальный уровень	LLC (управление логической связью)
Сетевой уровень	IP (Интернет-протокол)
Транспортный уровень	TCP (протокол управления передачей) UDP (протокол пользовательских дейтаграмм)
Диагностики	Светодиод – зеленый 1000 Мбит/с (соединение/активность) Светодиод – желтый 100 Мбит/с (соединение/активность) Светодиоды – зеленый и желтый 10 Мбит/с (связь/активность)

	NET 2
<b>Изоляция</b>	1500 В перем.тока / 1 минута
<b>Ethernet-интерфейс к логике</b>	1500 В перем.тока / 1 минута
<b>Интерфейс Ethernet к интерфейсу Ethernet</b>	1500 В перем.тока / 1 минута
<b>Интерфейс Ethernet для защитного заземления</b>	1500 В перем.тока / 1 минута

Таблица 13: Особенности интерфейса Ethernet NET 2

### 2.2.7.3. NET 3

	NET 3
Соединитель	Экранированная розетка RJ45
Авто кроссовер	Да
Максимальная длина кабеля	100 м
Тип кабеля	UTP или ScTP, категория 5
Скорость передачи данных	10/100/1000 Мбит/с
Физический слой	10BASE-Te/100BASE-TX/1000BASE-T
Канальный уровень	LLC (управление логической связью)
Сетевой уровень	IP (Интернет-протокол)
Транспортный уровень	TCP (протокол управления передачей) UDP (протокол пользовательских дейтаграмм)

## 2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ

Диагностики	Светодиод – зеленый 1000 Мбит/с (соединение/активность) Светодиод – желтый 100 Мбит/с (соединение/активность) Светодиоды – зеленый и желтый 10 Мбит/с (связь/активность)
Изоляция Ethernet-интерфейс к логике Интерфейс Ethernet к интерфейсу Ethernet Интерфейс Ethernet для защитного заземления	1500 В перем.тока / 1 минута 1500 В перем.тока / 1 минута  1500 В перем.тока / 1 минута

Таблица 14: Особенности интерфейса Ethernet NET 3

### 2.2.8. Интерфейс карты памяти

Карта памяти может использоваться для хранения различных данных, таких как журналы пользователей, веб-страницы, проектная документация и исходные файлы.

	Карта памяти
Максимальная емкость	32 Гбайт
Минимальная емкость	2 Гбайт
Тип	MicroSD
Файловая система	FAT32
Извлеките карту безопасно	Да, через специальное меню для этой функции.

Таблица 15: Особенности интерфейса карты памяти

#### Примечания:

**Максимальная емкость:** Емкость карты памяти должна быть меньше или равна этому пределу для правильной работы ЦП Nexто, в противном случае ЦП Nexто может не обнаружить карту памяти или даже вызвать проблемы во время передачи данных.

**Минимальная емкость:** Емкость карты памяти должна быть больше или равна этому пределу для правильной работы ЦП Nexто, в противном случае ЦП Nexто может не обнаружить карту памяти или даже вызвать проблемы во время передачи данных.

**Файловая система:** рекомендуется форматировать карту памяти с помощью процессора Nexто, иначе это может привести к потере производительности в интерфейсе карты памяти.

### 2.2.9. Источник питания

Номинальное входное напряжение	24 В пост.тока
Максимальная выходная мощность	15 Вт
Максимальный выходной ток	3 А
Входное напряжение	от 18 до 30 В пост.тока
Максимальный входной ток (пусковой)	15 А
Максимальный входной ток	1,5 А
Прерывание максимального входного напряжения	10 мс при 24 В пост.тока
Изоляция	1000 В перем.тока / 1 минута 1000 В перем.тока / 1 минута
Вход в логику	0.5 мм <sup>2</sup>
Вход на защитное заземление	Да
Поперечное сечение	Да
Защита от инверсии полярности	Нет
Внутренний предохранитель	Нет

Таблица 16: Особенности источника питания

**2.2.10. Характеристики окружающей среды**

Потребляемый ток на блоке питания	-
рассеяние	9 Вт
Рабочая Температура	от -20 до 60 °C
Температура хранения	от -40 до 75 °C
Относительная влажность	5% до 96%, без конденсации
Защитное покрытие	Да
Уровень IP	IP 20
Размеры модуля (Ш x В x Г)	36,00 x 114,63 x 115,30 мм
Размеры упаковки (Ш x В x Г)	44,00 x 122,00 x 147,00 мм
Масса	330 г
Вес с упаковкой	380 г

Таблица 17: Характеристики окружающей среды

**Примечание:**

**Конформное покрытие электронных схем:** покрытие электронных схем защищает внутренние части продукта от влаги, пыли и других агрессивных элементов электронных схем.

## 2.3. Совместимость с другими продуктами

Для разработки приложения для ЦП серии Nexto необходимо проверить версию MasterTool IEC XE. Следующая таблица показывает минимальную требуемую версию (где были представлены контроллеры) и соответствующую версию прошивки на тот момент:

ЦП Nexto Series	MasterTool IEC XE	Версия прошивки
NX3008	3.40 или выше	1.10.0.0 или выше

Таблица 18: Совместимость с другими продуктами

Кроме того, в план разработки MasterTool IEC XE могут быть включены некоторые функции (например, специальные функциональные блоки и т. д.), для которых может потребоваться минимальная версия микропрограммы. Во время загрузки приложения MasterTool IEC XE проверяет версию прошивки, установленную на контроллере, и, если она не соответствует минимальным требованиям, выводит сообщение с запросом на обновление. Последнюю версию прошивки можно загрузить с веб-сайта Altus, и она полностью совместима с предыдущими приложениями.

## 2.4. Производительность

Производительность процессоров серии Nexto зависит от:

- Время приложения пользователя
- Интервал применения
- Время операционной системы
- Количество модулей (данные процесса, ввод/вывод, среди прочего)

### 2.4.1. Время применения

Время выполнения приложения Nexto CPUs зависит от следующих переменных:

- Время чтения ввода (локальное и удаленное)
- Время выполнения задач
- Время записи вывода (локальное и удаленное)

Важно подчеркнуть, что на время выполнения «Основной задачи» будет непосредственно влиять системная задача «Конфигурация», задача высокого приоритета, периодически выполняемая системой. Задача «Конфигурация» может прерывать «Основную задачу», а при использовании коммуникационных модулей, таких как модуль Ethernet NX5000, например, добавление времени к «Основной задаче» может составлять до 25% среднего времени выполнения.

### 2.4.2. Время выполнения инструкций

В таблице ниже представлено необходимое время выполнения различных инструкций.

Инструкция	Язык	Тип переменной	Время (мкс)
1000 Контакты	LD	BOOL	2,1
1000 Отделы	LD, ST	INT	9,2
		REAL	17,0
1000 Мультипликации	LD, ST	INT	6,4
		REAL	8,2
1000 Суммы	LD, ST	INT	4,4
		REAL	8,2

Таблица 19: Время обучения

### 2.4.3. Время инициализации

Процессоры серии Nexто имеют время инициализации 30 с, а начальный экран с логотипом NEXТО (заставка) появляется через 2 с после включения питания.

### 2.4.4. Интервал времени

Интервал времени основной задачи ЦП может быть установлен от 5 до 750 мс.

### 2.5. Физические размеры

Размеры в мм.

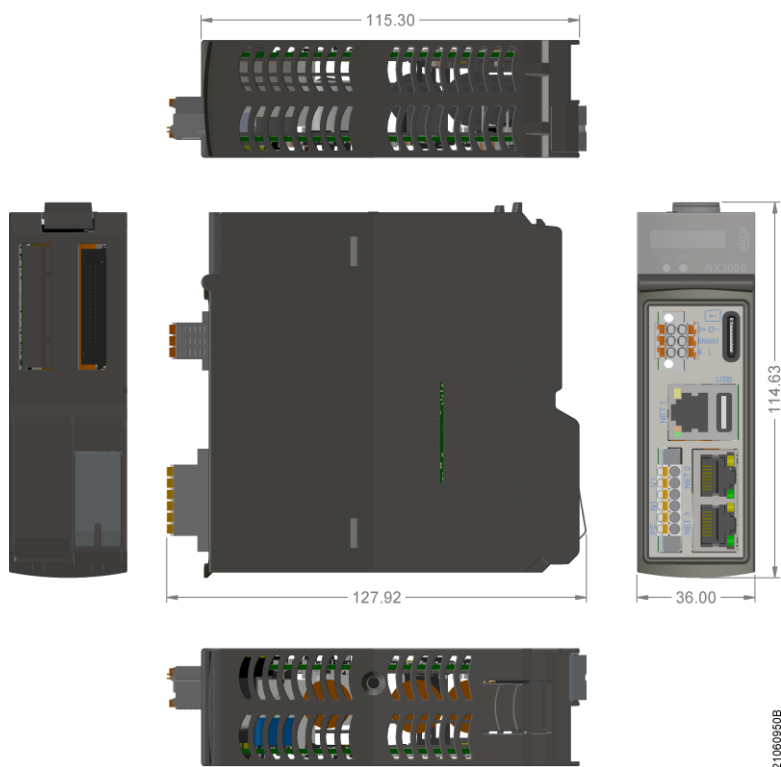


Рисунок 4: Размеры в мм

210609508

## 2.6. Данные о покупке

### 2.6.1. Включенные пункты

В комплект поставки входят следующие элементы:

- Модуль NX3008.
- 6-контактный разъем с креплением
- 2x3 разъем с креплением

**2.6.2. Код продукта**

Для покупки продукта необходимо использовать следующий код:

Код	Описание
<b>NX3008</b>	ЦП, 3 порта Ethernet, 1 порт USB, 1 последовательный порт, 1 порт CAN, интерфейс карты памяти, поддержка удаленного расширения стойки, встроенный блок питания и поддержка пользовательских веб-страниц

Таблица 20: Код продукта

**2.7. Связанные продукты**

Следующие продукты должны быть приобретены отдельно, когда это необходимо:

Код	Описание
<b>MT8500</b>	MasterTool IEC XE
<b>AL-2600</b>	Ответвление сети RS-485 и терминатор
<b>AL-2306</b>	Кабель RS-485 для сети MODBUS или CAN
<b>NX9101</b>	Карта памяти microSD на 32 ГБ с адаптерами miniSD и SD
<b>NX9202</b>	Кабель RJ45-RJ45 2 м
<b>NX9205</b>	Кабель RJ45-RJ45 5 м
<b>NX9210</b>	Кабель RJ45-RJ45 10 м
<b>NX9000</b>	Стойка объединительной платы на 8 слотов
<b>NX9001</b>	Стойка объединительной платы на 12 слотов
<b>NX9002</b>	Стойка объединительной платы на 16 слотов
<b>NX9003</b>	Стойка объединительной платы на 24 слота
<b>NX9010</b>	Стойка объединительной платы на 8 слотов (без горячей замены)
<b>NX9020</b>	Основание с 2 слотами для панельной сборки

Таблица 21: Связанные продукты

**Примечания:**

**MT8500:** MasterTool IEC XE доступен в четырех различных версиях: LITE, BASIC, PROFESSIONAL и ADVANCED. Дополнительные сведения см. в Руководстве пользователя MasterTool IEC XE — MU299609.

**AL-2600:** Этот модуль используется для разветвления и завершения сетей RS-422/485. Для каждого сетевого узла требуется AL-2600. AL-2600, который находится на концах сети, должен быть сконфигурирован с оконечной нагрузкой, за исключением случаев, когда есть устройство с активной внутренней оконечной нагрузкой, остальные должны быть сконфигурированы без оконечной нагрузки.

**AL-2306:** Две экранированные витые пары без разъемов, используемые для сетей на основе RS-485 или CAN.  
**NX9202/NX9205/NX9210:** Кабели, используемые для связи Ethernet и для соединения модулей расширения шины.  
**NX9020:** основание с 2 слотами для панельной сборки.

## 3. Установка

В этой главе представлены необходимые процедуры для физической установки ЦП серии Nexto, а также меры предосторожности, которые следует соблюдать при другой установке внутри панели, где установлен ЦП.

### ВНИМАНИЕ

Если оборудование используется способом, не указанным в данном руководстве, защита, обеспечиваемая оборудованием, может быть нарушена.

### 3.1. Механическая установка

Этот продукт должен быть вставлен в позицию 0 стойки объединительной панели. Для него требуются две последовательные позиции, это означает, что он использует позиции 0 и 1 стойки.

Механическая сборка этого модуля описана в Руководстве пользователя серии Nexto — MU214600.

### 3.2. Электрическая установка

### ОПАСНО

При выполнении любой установки в электрическом щите убедитесь, что основной источник питания ВЫКЛЮЧЕН.

На рисунке ниже показана электрическая схема ЦП NX3008, установленного в стойку объединительной платы серии Nexto. Изображенное размещение разъемов является просто иллюстративным.

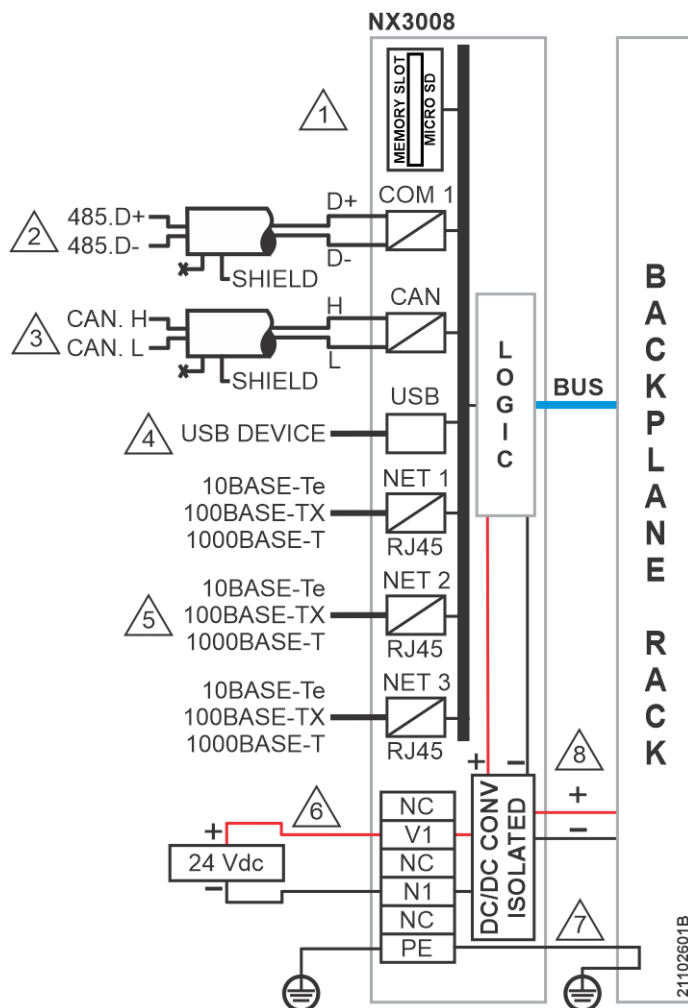


Рисунок 5: Электрическая схема процессора NX3008

**Примечания диаграммы:**

△1 Интерфейс карты MicroSD.

△2 Интерфейс RS-485.

△3 Интерфейс CAN.

△4 Интерфейс USB 2.0.

△5 Стандартный интерфейс Ethernet 10BASE-T<sub>e</sub>/100BASE-TX/1000BASE-T.

△6 Источник питания подключается к клеммам V1 и N1. Используйте кабель сечением 0,5 мм<sup>2</sup>.

△7 ЦП заземлен через стойки объединительной платы серии Nexto. Рекомендуется усилить заземление, подключаемое к клемме PE. Используйте кабель сечением 0,5 мм<sup>2</sup>.

△8 ЦПУ питает другие модули через соединение со стойкой объединительной платы.

⊕ Клемма функционального заземления.

### 3.3. Сетевое соединение Ethernet

Изолированный коммуникационный интерфейс NET 1, NET 2 и NET 3 позволяет подключаться к сети Ethernet, однако интерфейс NET 1 лучше всего подходит для связи с MasterTool IEC XE.

Сетевое соединение Ethernet использует кабели витой пары (10Base-Te/100Base-TX/1000Base-T), а определение скорости выполняется автоматически процессором Nexto. Этот кабель должен иметь один конец, подключенный к интерфейсу, который, вероятно, будет использоваться, а другой — к концентратору, коммутатору, микрокомпьютеру или другой точке сети Ethernet.

#### 3.3.1. IP Адрес

Интерфейс NET 1 Ethernet используется для связи Ethernet и для конфигурации ЦП со следующими параметрами конфигурации по умолчанию:

	NET 1
IP Адрес	192.168.15.1
Маска подсети	255.255.255.0
Адрес шлюза	192.168.15.253

Таблица 22: Конфигурация параметров по умолчанию для интерфейса Ethernet NET 1

Параметры IP-адреса и маски подсети можно увидеть на графическом дисплее ЦП через меню параметров, как описано в разделе «Информационное и конфигурационное меню ЦП».

Первоначально интерфейс NET 1 должен быть подключен к сети ПК с той же маской подсети для связи с MasterTool IEC XE, где можно изменить сетевые параметры. Дополнительные сведения о конфигурации и изменении параметров см. в разделе «Конфигурация интерфейсов Ethernet».

Интерфейс Ethernet NET 2 используется только для связи Ethernet и поставляется со следующими параметрами конфигурации по умолчанию:

	NET 2
IP Адрес	192.168.16.1
Маска подсети	255.255.255.0
Адрес шлюза	192.168.16.253

Таблица 23: Конфигурация параметров по умолчанию для интерфейса Ethernet NET 2

Параметры IP-адреса и маски подсети можно увидеть на графическом дисплее ЦП через меню параметров, как описано в разделе «Информационное и конфигурационное меню ЦП».

Сетевые параметры интерфейса NET 2 можно изменить с помощью MasterTool IEC XE. Дополнительные сведения о конфигурации и изменении параметров см. в разделе «Конфигурация интерфейсов Ethernet».

	NET 3
IP Адрес	192.168.17.1
Маска подсети	255.255.255.0
Адрес шлюза	192.168.17.253

Таблица 24: Конфигурация параметров по умолчанию для интерфейса Ethernet NET 3

Параметры IP-адреса и маски подсети можно увидеть на графическом дисплее ЦП через меню параметров, как описано в разделе «Информационное и конфигурационное меню ЦП».

Сетевые параметры интерфейса NET 3 можно изменить с помощью MasterTool IEC XE. Дополнительные сведения о конфигурации и изменении параметров см. в разделе «Конфигурация интерфейсов Ethernet».

#### 3.3.2. Бесплатный ARP

Интерфейс NETx Ethernet оперативно отправляет пакеты типа ARP в широковещательном режиме, сообщая свой IP- и MAC-адрес для всех устройств, подключенных к сети. Эти пакеты отправляются во время загрузки нового приложения программным обеспечением MasterTool IEC XE и при запуске ЦП, когда приложение переходит в режим выполнения.

Пять команд ARP запускаются в течение начального интервала в 200 мс, удваивая интервал для каждой новой запускаемой команды, что в сумме составляет 3 с. Пример: первый триггер происходит в момент времени 0, второй — в 200 мс, третий — в 600 мс и так далее до пятого триггера в момент времени 3 с.

#### 3.3.3. Установка сетевого кабеля

Порты Ethernet ЦП серии Nexto, обозначенные на панели NET, имеют стандартную распиновку, аналогичную

### 3. УСТАНОВКА

используемой в ПК. Тип разъема, тип кабеля, физический уровень, а также другие сведения о ЦП и сетевом устройстве Ethernet определяются в интерфейсах Ethernet.

В приведенной ниже таблице представлен гнездовой разъем ЦП RJ-45 Nexto с идентификацией и описанием допустимой распиновки для физических уровней 10BASE-TE и 100BASE-TX.

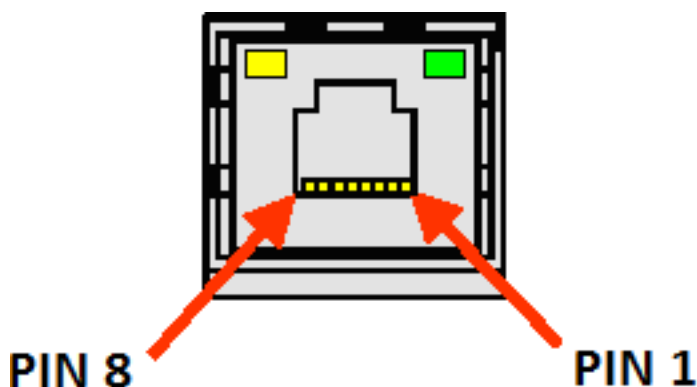


Рисунок 6: RJ45 Женский разъем

ПИН	Сигнал	Описание
1	TXD +	Передача данных, положительный
2	TXD -	Передача данных, отрицательный
3	RXD +	Прием данных, положительный
4	NU	Не используется
5	NU	Не используется
6	RXD -	Прием данных, отрицательный
7	NU	Не используется
8	NU	Не используется

Таблица 25: Распиновка гнездового разъема RJ45 — 10BASE-TE и 100BASE-TX

В приведенной ниже таблице показан гнездовой разъем RJ45 процессоров Nexto с идентификацией и описанием допустимой распиновки для физического уровня 1GBASE-T.

ПИН	Сигнал	Описание
1	TxRx A +	Двунаправленный, положительный
2	TxRx A -	Двунаправленный, отрицательный
3	TxRx B +	Двунаправленный, положительный
4	TxRx C +	Двунаправленный, положительный
5	TxRx C -	Двунаправленный, отрицательный
6	TxRx B -	Двунаправленный, отрицательный
7	TxRx D +	Двунаправленный, положительный
8	TxRx D -	Двунаправленный, отрицательный

Таблица 26: Распиновка гнездового разъема RJ45 — 1GBASE-T

Интерфейс может быть подключен к коммуникационной сети через концентратор или коммутатор, либо напрямую от коммуникационного оборудования. В этом последнем случае, благодаря функции автоматического кроссовера ЦП Nexто, нет необходимости в перекрестном сетевом кабеле, который используется для двухточечного соединения двух ПК через порт Ethernet.

Важно подчеркнуть, что под сетевым кабелем понимается пара штекерных разъемов RJ45, соединенных кабелем UTP или ScTP категории 5, будь то прямое или перекрестное соединение. Он используется для связи двух устройств через порт Ethernet.

Эти кабели, как правило, имеют замок соединения, который гарантирует идеальное соединение между гнездовым разъемом интерфейса и штекерным разъемом кабеля. В момент установки штекерный разъем должен быть вставлен в гнездовой разъем модуля до щелчка, обеспечивающего блокировку. Чтобы отсоединить кабель от модуля, необходимо использовать рычаг блокировки, чтобы отпереть один от другого.

### 3.4. Последовательное сетевое соединение RS-485 и CAN

Как показано на схеме электрического монтажа, интерфейсы RS-485 и CAN используют два сигнала связи и заземление. Рекомендуемый кабель AL-2306, использующий одну из двух пар и экран. Если контроллер расположен на одном конце сети, необходимо включить внутреннюю оконечную нагрузку (см. разделы конфигурации «Конфигурация интерфейса CAN (локальная сеть контроллера)» и «Дополнительные конфигурации для CAN и RS-485 соответственно»).

### 3.5. Установка карты памяти

В этом разделе показано, как вставить карту памяти в модели ЦП серии Nexто. Для получения дополнительной информации см. раздел «Карта памяти».

Первоначально необходимо позаботиться о правильном положении карты памяти. Один его угол отличается от трех других, и этот угол следует использовать в качестве ориентира для правильной вставки карты. Поэтому карту памяти необходимо вставлять согласно изображению на лицевой части процессора или так, как показано на рисунке ниже.

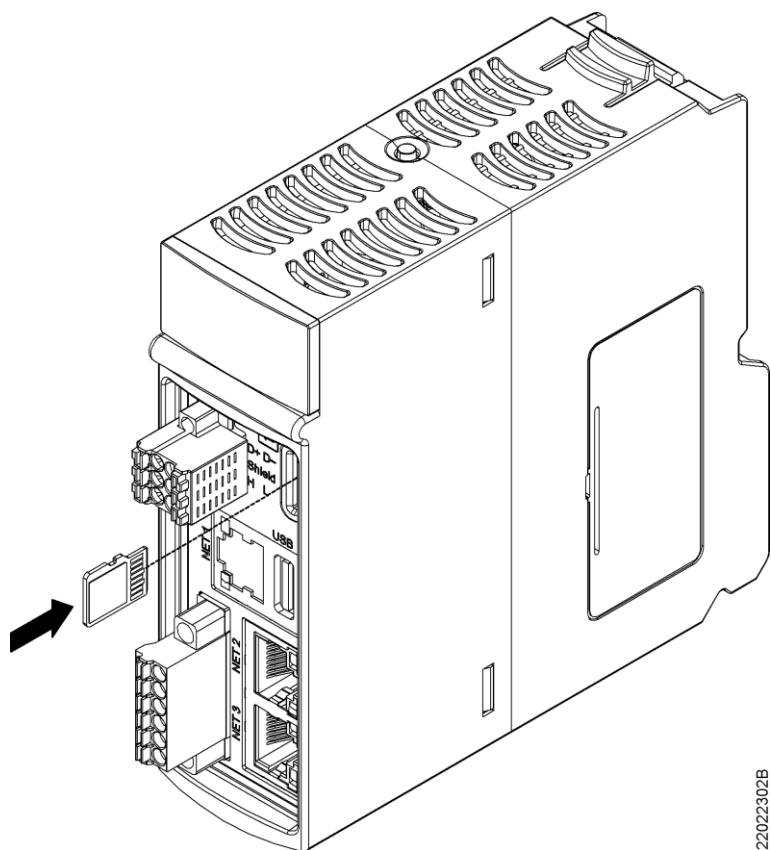


Рисунок 7: Установка карты памяти

Если карта установлена правильно, на графическом дисплее процессора появится символ. Для безопасного извлечения карты необходимо нажать клавишу MS, затем произойдет небольшая задержка, и символ карты исчезнет с графического дисплея. Теперь карта готова к снятию. Для этого карту необходимо прижать к процессору до щелчка, затем отпустить и вынуть из отсека, как показано на рисунке ниже. В этот момент карта будет свободной.

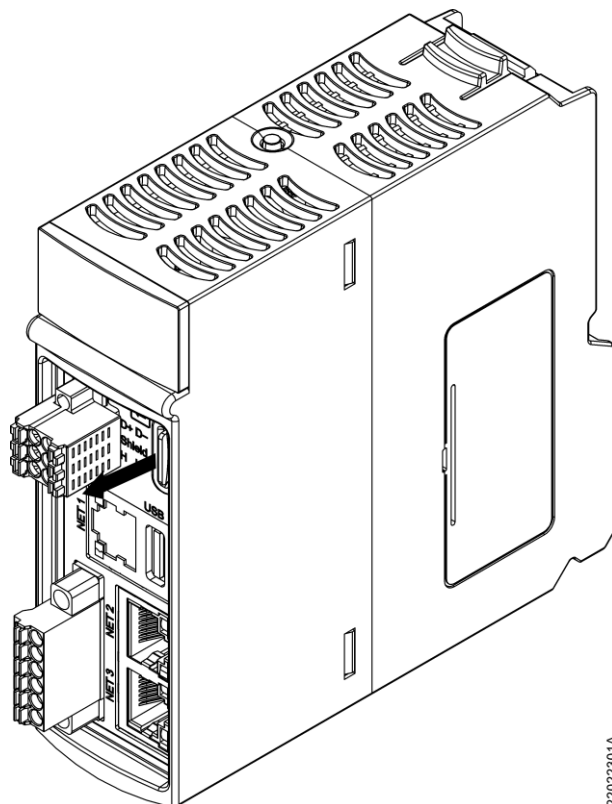


Рисунок 8: Извлечение карты памяти из процессора

## 3.6. Установка архитектуры

### 3.6.1. Установка модуля в стойку основной объединительной платы

Серия Nexto имеет эксклюзивный метод подключения и отключения модулей на шине, не требующий больших усилий со стороны оператора и гарантирующий целостность соединения. Для получения дополнительной информации о фиксации продуктов серии Nexto см. Руководство пользователя серии Nexto — MU214600.

## 3.7. Установка программатора

Для выполнения установки программного обеспечения для разработки MasterTool IEC XE необходимо иметь дистрибутив CD-ROM или загрузить установочный файл с сайта <https://www.altus.com.br/en/>. Для получения дополнительной информации о пошаговой установке обратитесь к руководству пользователя MasterTool IEC XE MT8500 — MU299609.

## 4. Первоначальное программирование

Основная цель этой главы — помочь в программировании и настройке ЦП серии Nexто, позволяя пользователю сделать первые шаги перед началом программирования устройства.

ЦП серии Nexто использует стандарт IEC 61131-3 для языков программирования, а именно: IL, ST, LD, SFC и FBD, а также дополнительный язык CFC. Эти языки можно разделить на текстовые и графические. IL и ST являются текстовыми языками и похожи на ассемблер и С соответственно. LD, SFC, FBD и CFC являются графическими языками. LD использует представление блока реле, и оно похоже на схемы реле. SFC использует представление диаграммы последовательности, что позволяет легко увидеть последовательность событий. FBD и CFC используют группу функциональных блоков, что позволяет четко видеть функции, выполняемые каждым действием.

Программирование осуществляется через интерфейс разработки MasterTool IEC XE (IDE). MasterTool IEC XE позволяет использовать шесть языков в одном проекте, поэтому пользователь может применять лучшие функции, предлагаемые каждым языком, что приводит к более эффективной разработке приложений, упрощению документирования и обслуживания в будущем.

Дополнительную информацию о программировании см. в Руководстве пользователя MasterTool IEC XE — MU299609, Руководстве по программированию MasterTool IEC XE — MP399609 или стандарте IEC 61131-3.

### 4.1. Организация памяти и доступ

В отличие от других устройств серии Nexто (которые основаны на ЦП с прямым порядком байтов), ЦП NX3008 основан на ЦП ARM, в котором используется традиционная организация памяти с прямым порядком байтов (такая же, как в процессорах x86 и Intel). При таком типе организации памяти наименее значимый байт сохраняется первым и всегда будет иметь наименьший адрес (например, %QB0 всегда будет менее значащим, чем %QB1, как показано в Таблице ниже, где для строки CPUNEXTO буква O — байт 0, а буква C — байт 7).

Кроме того, доступ к памяти должен выполняться осторожно, поскольку переменные с большим количеством бит (WORD, DWORD, LONG) используют в качестве индекса самый старший байт, другими словами, %QD4 всегда будет иметь самый старший байт %QB4. Поэтому нет необходимости выполнять вычисления, чтобы определить, какие DWORD соответствуют определенным байтам. В приведенной ниже таблице показана организация с прямым порядком байтов и с прямым порядком байтов.

MSB ← Формат хранения и передачи двоичных данных, при котором сначала передаётся младший бит → LSB								
BYTE	%QB7	%QB6	%QB5	%QB4	%QB3	%QB2	%QB1	%QB0
	C	P	U	N	E	X	T	O
WORD	%QW3		%QW2		%QW1		%QW0	
	CP		UN		EX		TO	
DWORD	%QD1				%QD0			
	CPUN				EXTO			
LWORD	%QL0							
	CPUNEXTO							
MSB ← Формат для хранения и передачи двоичных данных, при которой старший передаётся сначала → LSB								
BYTE	%QB0	%QB1	%QB2	%QB3	%QB4	%QB5	%QB6	%QB7
	C	P	U	N	E	X	T	O
WORD	%QW0		%QW2		%QW4		%QW6	
	CP		UN		EX		TO	
DWORD	%QD0				%QD4			
	CPUN				EXTO			
LWORD	%QL0							
	CPUNEXTO							

Таблица 27: Организация памяти и пример доступа

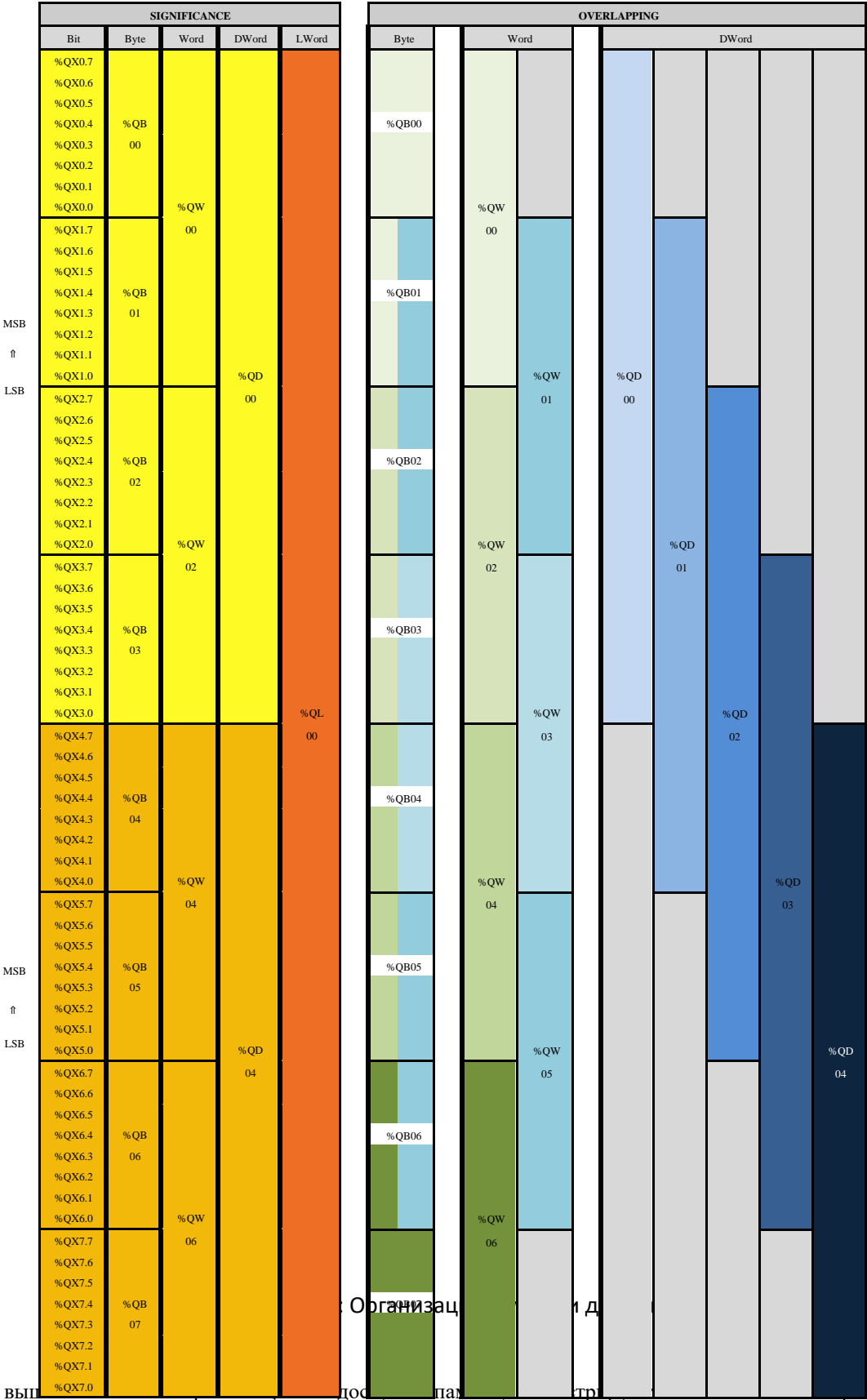


Таблица вычисления значений переменных для различных типов данных. Таблица показывает, как значения переменных для различных типов данных (Byte, Word, DWord, LWord) вычисляются из значений переменных для других типов переменных, включая перекрытие.

## 4.2. Профили проектов

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Профиль проекта в MasterTool IEC XE состоит из шаблона приложения вместе с группой правил проверки, которые определяют разработку приложения, снижая сложность программирования. Приложения могут создаваться по следующим профилям:

Одинарная Базовая Нормальная Экспертная Пользовательская

Профиль машины

Профиль проекта выбирается в мастере создания проекта. Каждый профиль проекта определяет шаблон стандартных имен для задач и программ, которые предварительно создаются в соответствии с выбранным профилем проекта. Кроме того, во время компиляции проекта (генерации кода) MasterTool IEC XE проверяет все правила, определенные выбранным профилем.

В следующих разделах подробно описаны характеристики каждого профиля, которые следуют постепенному наклону сложности. Основываясь на этих определениях, рекомендуется, чтобы пользователь всегда использовал самый простой профиль, отвечающий потребностям его приложения, и переходил на более сложный профиль только тогда, когда соответствующие правила больше препятствуют развитию, чем дидактические упрощения. Важно отметить, что инструмент программирования позволяет изменить профиль из существующего проекта (см. раздел обновления проекта в руководстве пользователя MasterTool IEC XE — MU299609), но разработчик должен внести любые необходимые корректировки, чтобы проект стал совместимым с правилами нового выбранного профиля.

##### ВНИМАНИЕ

Когда используется более одной задачи, доступ к вводу-выводу может осуществляться только в контексте MainTask. В случае, если параметр «Включить обновление ввода-вывода для каждой задачи» не может быть использован, он присутствует в MasterTool IEC XE версии 2.01.

#### 4.2.1. Единый профиль проекта

В профиле единого проекта у приложения есть только одна пользовательская задача, MainTask. Эта задача отвечает за выполнение одной программной единицы типа «Программа», называемой MainPrg. Эта отдельная программа может вызывать другой модуль программирования типа «Программа», «Функция» или «Функциональный блок», но весь код будет выполняться исключительно основной задачей.

В этом профиле MainTask будет циклического типа (Циклический) с фиксированным приоритетом 13 (тринадцать) и запускает исключительно программу MainPrg в непрерывном цикле. MainTask уже полностью определена, и разработчику необходимо создать программу MainPrg, используя любой из языков стандарта IEC 61131-3. Не всегда возможно преобразовать программу на другой язык, но всегда можно создать новую программу, построенную на другом языке, с тем же именем и заменить ее. Стандартный вариант MasterTool IEC XE заключается в использовании стандартного проекта MasterTool, связанного с одним профилем, который также включает в себя MainPrg, созданный на языке, выбранном во время создания проекта.

Этот тип приложений никогда не должен учитывать такие проблемы, как непротиворечивость данных, совместное использование ресурсов или механизмы взаимного исключения.

Задание	POU	Приоритетность	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 ms	-

Таблица 29: Задача с одним профилем

#### 4.2.2. Базовый профиль проекта

В базовом профиле проекта у приложения есть одна пользовательская задача непрерывного типа, называемая MainTask, которая выполняет программу в непрерывном цикле (без определения времени цикла) с фиксированным приоритетом 13 (тринадцать). Эта задача отвечает за выполнение одного программного модуля программирования, называемого MainPrg. Важно отметить, что время цикла может варьироваться в зависимости от количества используемых коммуникационных задач, так как в этом режиме основная задача прерывается коммуникационными задачами.

Этот профиль также позволяет включать две событийные задачи с более высоким приоритетом, которые могут прервать (вытеснить) MainTask в любой момент: задача с именем ExternInterruptTask00 является событийной задачей типа Внешний с приоритетом, фиксированным в 02 (два); задача с именем TimeInterruptTask00 является событийной задачей типа Циклический с фиксированным приоритетом 01 (один).

Базовая модель шаблона проекта включает в себя три задачи, уже полностью определенные, как представлено в Таблице ниже. Разработчику нужно только создать сопутствующие программы.

Задание	POU	Приоритетность	Тип	Интервал	Событие
MainTask	MainPrg	13	Непрерывный	-	-
ExternInterruptTask00	ExternInterruptPrg00	02	Внешний	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 мс	-

Таблица 30: Основные задачи профиля

#### 4.2.3. Обычный профиль

В обычном профиле проекта у приложения есть одна пользовательская задача циклического типа, называемая MainTask. Эта задача отвечает за выполнение одного программного модуля программирования, называемого MainPrg. Эта программа и эта задача аналогичны единственной задаче и единственной программе Единого профиля, но здесь приложение может интегрировать дополнительные пользовательские задачи. Эти другие задачи, называемые ЦиклическийTask00 и ЦиклическийTask01, каждая из которых отвечает за монопольное выполнение своей соответствующей программы ЦиклическийPrg<nn>. Задачи ЦиклическийTask<nn> всегда относятся к циклическому типу и имеют фиксированный приоритет 13 (тринадцать), такой же приоритет, как и MainTask. Эти два типа образуют группу, называемую базовыми задачами, связанные с которыми программы могут вызывать другие программные модули типа «Программа», «Функция» и «Функциональный блок».

Кроме того, в этот профиль могут быть включены событийные задачи с более высоким приоритетом, чем базовые задачи, которые в любой момент могут прервать (прервать) выполнение этих задач.

Задача ExternInterruptTask00 представляет собой событийную задачу типа Внешний, выполнение которой инициируется некоторым внешним событием, таким как изменение управляющего сигнала на последовательном порту или изменение цифрового входа на шине NEXTO. Этот приоритет задачи зафиксирован в 02 (два), отвечая исключительно за выполнение программы ExternInterruptPrg00. Задача TimeInterruptTask00 является событийной задачей типа Циклический с фиксированным приоритетом 01 (один), отвечающей за выполнение исключительно программы TimeInterruptPrg00.

В обычной модели проекта есть пять задач, и их POU уже полностью определены, как показано в Таблице ниже. Разработчику нужно только реализовать содержимое программы, выбрав в мастере любой из языков стандарта IEC 61131-3. Интервал задач и триггерные события могут быть настроены разработчиком, а ненужные задачи могут быть исключены.

Задание	POU	Приоритетность	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 ms	-
ЦиклическийTask00	ЦиклическийPrg00	13	Циклический	200 ms	-
ЦиклическийTask01	ЦиклическийPrg01	13	Циклический	500 ms	-
ExternInterruptTask00	ExternInterruptPrg00	02	Внешний	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 ms	-

Таблица 31: Обычные задачи профиля

#### 4.2.4. Экспертные задачи профиля

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Профиль экспертного проекта включает в себя те же основные задачи, ЦиклическийTask<nn>, ExternInterruptTask00 и TimeInterruptTask00 с теми же приоритетами (13, 02 и 01 соответственно), но является расширением по сравнению с предыдущими из-за принятия нескольких событийных задач. То есть приложение может включать различные задачи ExternInterruptTask<nn> или TimeInterruptTask<nn>, которые выполняют программы ExternInterruptPrg<nn> и TimeInterruptPrg<nn>. Приоритеты дополнительных событийных задач можно свободно выбирать от 08 до 12. В этом профиле, помимо стандартных программ, каждая задача может выполнять дополнительные программы.

В этом профиле проекта приложение также может включать пользовательскую задачу FreeTask типа Freewheeling с приоритетом 31, отвечающую за выполнение программы FreePrg. Поскольку эта задача имеет низкий приоритет, она может быть прервана всеми остальными, чтобы выполнять коды, которые могут быть заблокированы.

Уже есть восемь полностью определенных задач, как показано в Таблице ниже, а также связанные с ними программы на выбранном языке. Интервалы и триггерные события любой задачи, а также приоритеты событийных задач могут быть установлены пользователем.

При разработке приложения с использованием профиля проекта «Эксперт» необходимо уделить особое внимание масштабированию событийных задач. Если существует обмен информацией и ресурсами между этими задачами или между ними и базовыми задачами, настоятельно рекомендуется принять стратегии для обеспечения согласованности данных.

Задание	POU	Приоритетность	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
ЦиклическийTask00	ЦиклическийPrg00	13	Циклический	200 мс	-
ЦиклическийTask01	ЦиклическийPrg01	13	Циклический	500 мс	-
ExternInterruptTask00	ExternInterruptPrg00	02	Внешний	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 мс	-
ExternInterruptTask01	ExternInterruptPrg01	11	Внешний	-	IO_EVT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Циклический	30 мс	-
FreeTask	FreePrg	31	Непрерывный	-	-

Таблица 32: Экспертные задачи профиля

##### 4.2.5. Пользовательский профиль проекта

Пользовательский профиль проекта позволяет разработчику изучить весь потенциал системы выполнения, реализованной в центральных процессорах серии Nexto. Никакая функциональность не отключена; никакие приоритеты, ассоциации задач и программ или номенклатуры не навязываются. Единственным исключением является MainTask, которая всегда должна существовать с этим именем в этом профиле.

Помимо задач в реальном времени с приоритетом от 00 до 15, которые планируются по приоритету, в этом профиле также можно определить задачи с более низким приоритетом в диапазоне от 16 до 31. В этом диапазоне используется полностью честный планировщик (разделение времени), что необходимо для запуска кодов, которые можно заблокировать (например, использование сокетов).

Разработчик может частично следовать или не следовать организации, определенной в других профилях проекта, в соответствии с характеристиками приложения. С другой стороны, пользовательская модель, связанная с этим профилем, не требует предварительного определения таких элементов, как задача, программа или параметр, предоставляя разработчику возможность создать все элементы, из которых состоит приложение. Однако пользователь может генерировать те же элементы, что и для профиля «Эксперт».

0

Задание	POU	Приоритетность	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
ЦиклическийTask00	ЦиклическийPrg00	13	Циклический	200 мс	-
ЦиклическийTask01	ЦиклическийPrg01	13	Циклический	500 мс	-
ExternInterruptTask00	ExternInterruptPrg00	02	Внешний	-	IO_EVT_0
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	20 мс	-
ExternInterruptTask01	ExternInterruptPrg01	11	Внешний	-	IO_EVT_1
TimeInterruptTask01	TimeInterruptPrg01	09	Циклический	30 мс	-
FreeTask	FreePrg	31	Непрерывный	-	-

Таблица 33: Пользовательские задачи профиля

**4.2.6. Профиль машины**

В профиль машины по умолчанию у приложения есть пользовательская задача типа Циклический с именем MainTask. Эта задача отвечает за реализацию одного POU типа Program, называемого MainPrg. Эта программа может вызывать другие модули программирования типа «Программа», «Функция» или «Функциональный блок», но любой пользовательский код будет выполняться исключительно MainTask.

Этот профиль характеризуется более короткими интервалами в MainTask, что позволяет быстрее выполнять пользовательский код. Эта оптимизация возможна, потому что MainTask также выполняет обработку шины. Таким образом, в отличие от других профилей, машинный профиль не требует переключения контекста для обработки шины, что сокращает общее время обработки.

Этот профиль может дополнительно включать задачу прерывания, называемую TimeInterruptTask00, с более высоким приоритетом, чем основная задача, и, следовательно, может прервать ее выполнение в любое время.

Задание	POU	Приоритетность	Тип	Интервал	Событие
MainTask	MainPrg	13	Циклический	20 мс	-
TimeInterruptTask00	TimeInterruptPrg00	01	Циклический	4 мс	-

Таблица 34: Задачи профиля машины

Кроме того, этот профиль поддерживает включение дополнительных задач, связанных с внешними прерываниями.

**4.2.7. Основная таблица**

		Профили проектов					
		единичный	машинный	основной	обыкновенный	экспертный	Пользовательский
Всего задач		01	04	[01.03]	[01.32]	[01.32]	[01.32]
Задачи по программе		01		01	01	<n>	<n>
Основная задача	Тип приоритетного количества	Циклический	Циклический	Продолжающийся	Циклический	Циклический	Циклический
		13	13	13	13	13	13
		01	01	01	01	01	01
Задача прерывания по времени	Тип приоритетного количества		Циклический	Циклический	Циклический	Циклический	Циклический
			01	01	01	01 or [08.12]	01 or [08.12]
			[00.01]	[00.01]	[00.01]	[00.31]	[00.31]
Задача внешнего прерывания	Тип приоритетного количества		Внешний	Внешний	Внешний	Внешний	Внешний
			02	02	02	02 or [08.12]	02 or [08.12]
			[00.01]	[00.01]	[00.01]	[00.31]	[00.31]
Циклическая задача	Тип приоритетного количества				Циклический	Циклический	Циклический
					13	13	13
					[00.31]	[00.31]	[00.31]
Бесплатная задача	Тип приоритетного количества					Продолжающийся	Продолжающийся
						31	31
						[00.01]	[00.01]
Событийная	Тип приоритетного						Event
							<n>

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

задача	ого количества						[00.31]
--------	-------------------	--	--	--	--	--	---------

Таблица 35: Основной профиль и задания в таблице

##### ВНИМАНИЕ

Предлагаемые имена ROU, связанные с задачами, не состоят. Их можно изменить, если они также изменены в конфигурациях задач.

#### 4.2.8. Максимальное количество заданий

Максимальное количество задач, которые может создать пользователь, определено только для пользовательского профиля, единственного, у которого есть это разрешение. У остальных уже созданы и настроены задачи. Однако создаваемые задачи должны использовать следующие префиксы в соответствии с типом каждой из задач: ЦиклическийTaskxx, TimeInterruptTaskxx, ExternInterruptTaskxx, где xx представляет номер создаваемой задачи.

Таблица ниже описывает максимальное количество задач ИЕС на ЦП и профиль проекта, где экземпляры протокола также считаются коммуникационными задачами ЦП.

Задача конфигурации (задача WHSB)	Тип задания	NX3008					
		S	B	N	E	P	M
Задача конфигурации (задача WHSB)	Циклический	1	1	1	1	1	0
Пользовательские задачи	Циклический Иницируется событием Иницируется внешним событием Произвольное движение Иницируется состоянием	1	1	31	31	31	2
		0	0	0	0	31	0
		0	1	0	30	31	0
		0	1	0	1	31	0
		0	0	0	0	31	0
NET — экземпляры клиента или сервера	Циклический	16					
COM(n) — главные или подчиненные экземпляры	Циклический	1					
ИТОГО		32					

Таблица 36: Максимальное количество задач IEC NX3008

**Примечания:**

**Расшифровка условных обозначений профиля:** Буквы S, B, N, E, C и M соответствуют профилям Single, Basic, Normal, Expert, Custom и Machine соответственно.

Значения: Число, определенное для каждого типа задачи, представляет максимально допустимые значения.

Задача WHSB: WHSB — это системная задача, которую необходимо учитывать, чтобы не превысить общую стоимость.

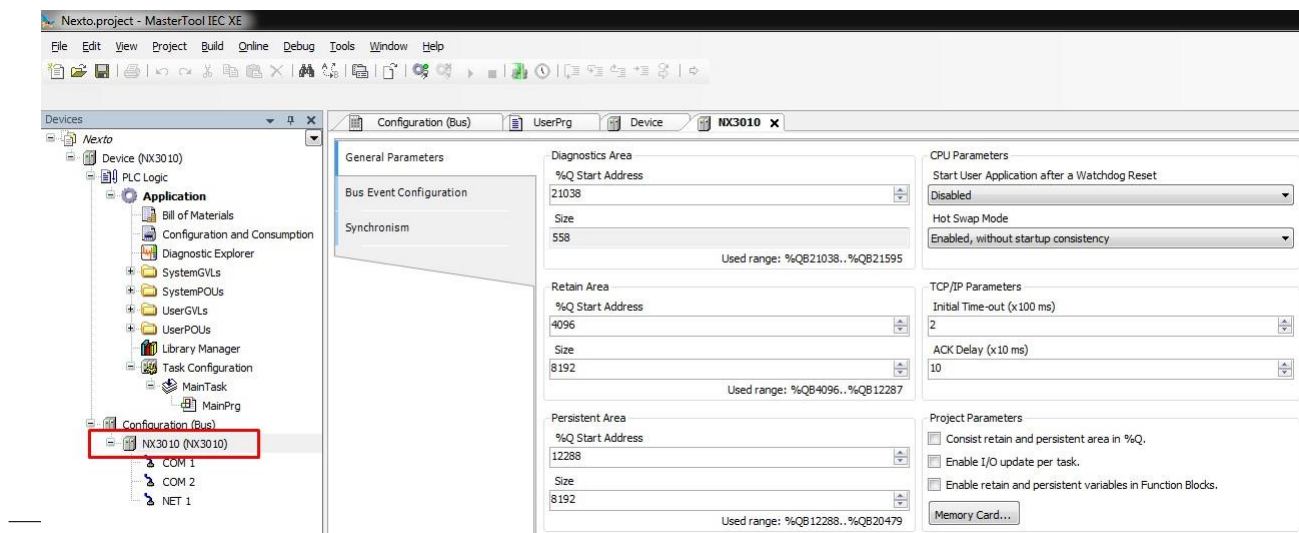
NETs — Экземпляры клиента или сервера: максимальное заданное значение учитывает все системные интерфейсы Ethernet, включая модули расширения, когда они применяются. Например. Экземпляры протокола MODBUS.

COM (n) — главные или подчиненные экземпляры: «n» представляет номер последовательного интерфейса. Даже с модулями расширения значение Таблицы будет максимальным для каждого интерфейса. Например. Экземпляры протокола MODBUS.

Итого: общее значение представляет собой не сумму всех задач профиля, а максимальное значение, допустимое для каждого ЦП. Таким образом, пользователь может создавать несколько типов задач, при этом установленные цифры для каждой и общая стоимость не превышаются.

### 4.3. Конфигурация процессора

Конфигурация ЦП Nexto находится в дереве устройств, как показано на рисунке ниже, и доступ к ней можно получить, дважды щелкнув соответствующий объект. На этой вкладке можно настроить область диагностики, область сохраняемой и постоянной памяти и режим горячей замены, среди других параметров, как описано в разделе «Конфигурация ЦП».



Кроме того, двойным щелчком по значку CPU NET 1 можно настроить интерфейс Ethernet, который будет использоваться для связи между контроллером и программным обеспечением MasterTool IEC XE.

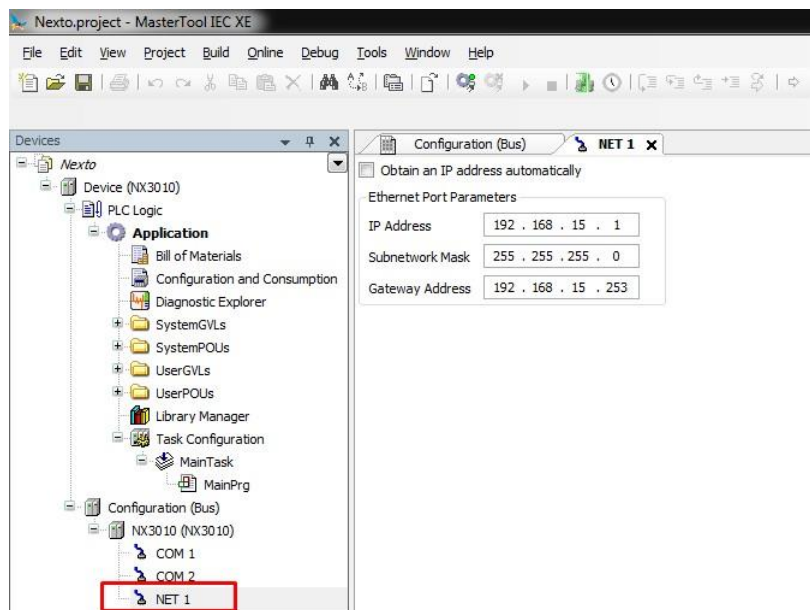


Рисунок 10: Настройка коммуникационного порта ЦП

Конфигурация, заданная на этой вкладке, будет применена к устройству только при отправке приложения на устройство (загрузке), что описано далее в разделах Поиск устройства и Логин.

#### 4.4. Библиотеки

Есть несколько ресурсов инструментов программирования, которые доступны через библиотеки. Следовательно, эти библиотеки должны быть вставлены в проект, чтобы его использование стало возможным. Процедуру установки и дополнительную информацию о доступных библиотеках можно найти в Руководстве по программированию MasterTool — MP399609.

#### 4.5. Вставка экземпляра протокола

ЦП серии Nexto, как описано в разделе «Протоколы», предлагают несколько протоколов связи. За исключением связи OPC DA и OPC UA, которые имеют другую процедуру настройки, вставку протокола можно выполнить, просто щелкнув правой кнопкой мыши нужный интерфейс связи, выбрав добавление устройства и, наконец, выполнив настройку, как показано на Раздел конфигурации протоколов. Ниже представлены примеры.

##### 4.5.1. MODBUS RTU

Первым шагом для настройки MODBUS RTU в ведомом режиме является включение экземпляра в нужный COM (в данном случае COM 1), щелкнув правой кнопкой на COM и выбрав «Добавить устройство...», как показано на рисунке ниже:

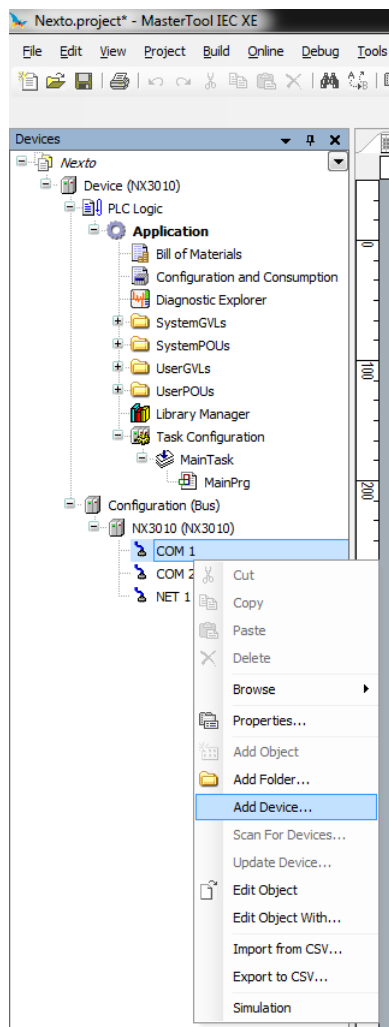


Рисунок 11: Добавление экземпляра

После этого на экране появятся доступные для пользователя протоколы. Определите режим конфигурации протокола, выбрав MODBUS Symbol RTU Slave для настройки символического отображения или MODBUS RTU Slave для прямой адресации (%Q) и нажмите «Добавить устройство», как показано на рисунке ниже.

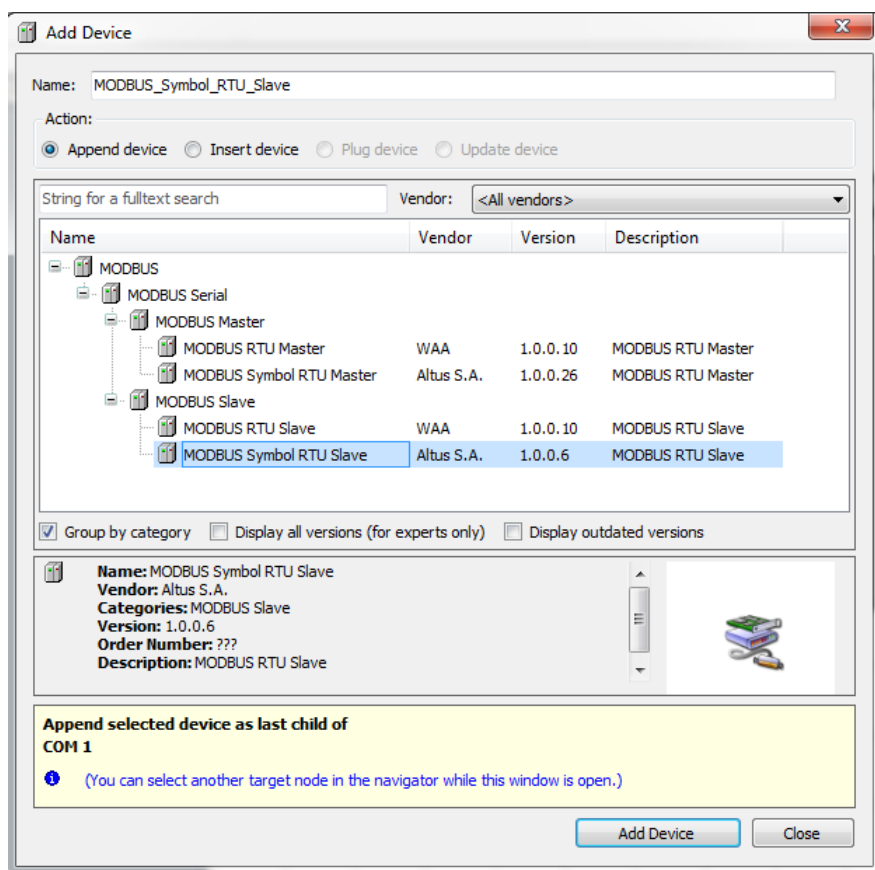


Рисунок 12: Выбор протокола

#### 4.5.2. MODBUS Ethernet

Первым шагом для настройки сети MODBUS Ethernet (клиент в данном примере) является включение экземпляра в желаемую NET (в данном случае NET 1, поскольку ЦП NX3010 имеет только один интерфейс Ethernet). Щелкните правой кнопкой мыши по NET и выберите «Добавить устройство...», как показано на рисунке ниже.

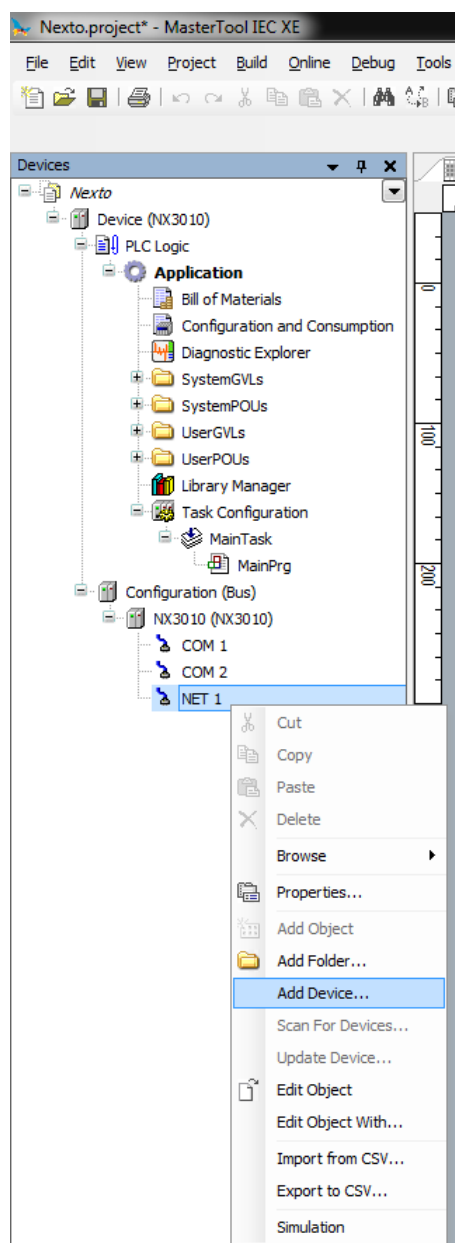


Рисунок 13: Добавление экземпляра

После этого на экране появятся доступные для пользователя протоколы. В этом меню определяется режим конфигурации протокола. Выбор опции «Клиент символа MODBUS» для настройки символьного отображения или «Клиент MODBUS» для прямой адресации (%Q). Затем нажмите «Добавить устройство», как показано на рисунке ниже.

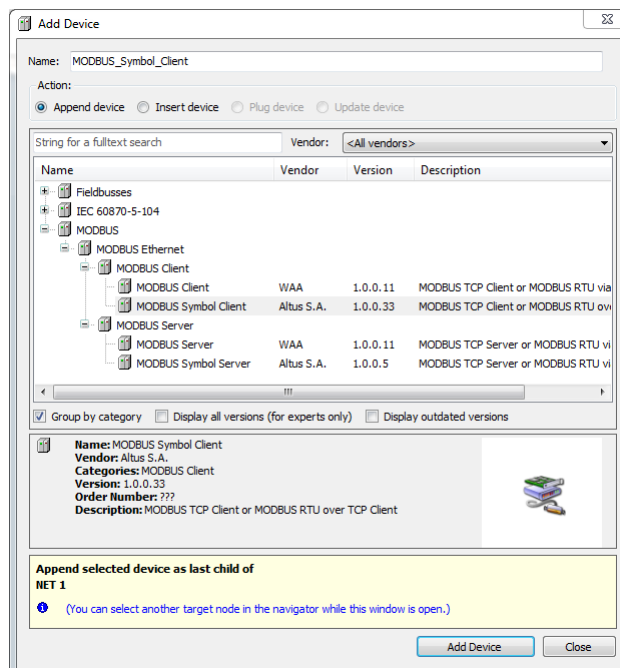


Рисунок 14: Выбор протокола

## 4.6. Поиск устройства

Чтобы установить связь между ЦП и MasterTool IEC XE, сначала необходимо найти и выбрать нужное устройство. Конфигурация этой связи находится на объекте Устройство в дереве устройств, на вкладке Параметры связи. На этой вкладке после выбора шлюза и нажатия кнопки «Сканировать сеть» программное обеспечение MasterTool IEC XE выполняет поиск устройств и показывает ЦП, найденные в сети интерфейса Ethernet станции, на которой работает инструмент.

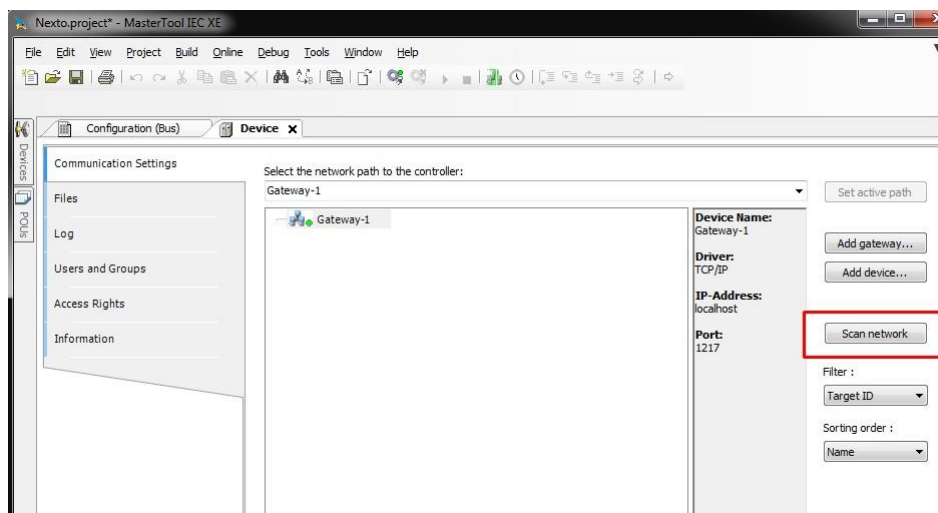


Рисунок 15: Поиск процессора

Если ранее настроенный шлюз отсутствует, его можно включить с помощью кнопки Добавить шлюз, используя IP-адрес по умолчанию localhost для использования шлюза, резидентного на станции, или изменив IP-адрес для использования внутреннего шлюза устройства.

Далее необходимо выбрать нужный контроллер, нажав «Установить активный путь». Это действие выбирает контроллер и сообщает конфигурационному программному обеспечению, какой контроллер следует использовать для связи и отправки проекта.

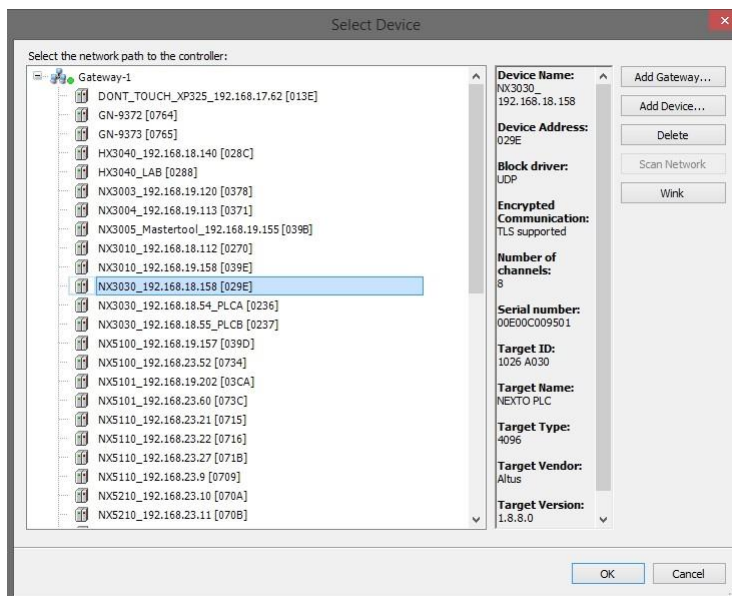


Рисунок 16: Выбор процессора

Кроме того, пользователь может изменить имя отображаемого устройства по умолчанию. Для этого необходимо щелкнуть правой кнопкой мыши по нужному устройству и выбрать «Изменить имя устройства». После изменения имени устройство ни при каких обстоятельствах не вернется к имени по умолчанию.

Если конфигурация Ethernet подключаемого ЦП находится в сети, отличной от интерфейса Ethernet станции, программное обеспечение MasterTool IEC XE не сможет найти устройство. В этом случае рекомендуется использовать команду «Простое подключение», расположенную в меню «Онлайн».

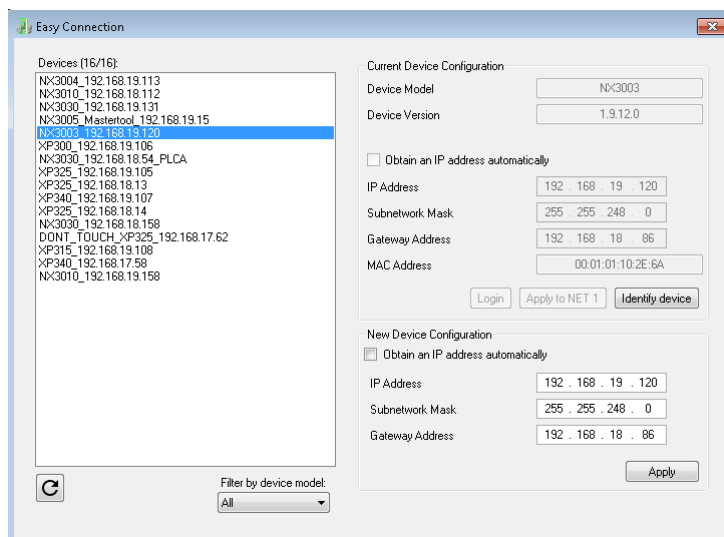


Рисунок 17: Легкое соединение

Эта команда выполняет связь на уровне MAC с интерфейсом NET 1 устройства, позволяя навсегда изменить конфигурацию интерфейса Ethernet ЦП, независимо от IP-конфигурации станции и от конфигурации, ранее настроенной на устройстве. Итак, с помощью этой команды можно изменить конфигурацию устройства, чтобы поместить его в ту же сеть интерфейса Ethernet станции, на которой работает MasterTool IEC XE, что позволяет найти и выбрать устройство для связи. Полное описание команды Easy Connection можно найти в Руководстве пользователя MasterTool IEC XE, код MU299609.

## 4.7. Логин

После компиляции приложения и исправления ошибок, которые могут быть обнаружены, пришло время отправить проект на ЦП. Для этого просто щелкните команду «Войти», расположенную в онлайн-меню MasterTool IEC XE, как показано на следующем рисунке. Эта операция может занять несколько секунд, в зависимости от размера сгенерированного файла.

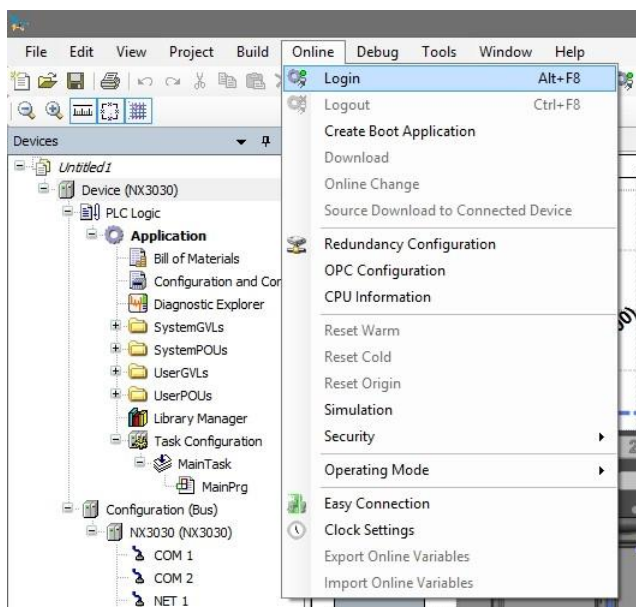


Рисунок 18: Отправка проекта в ЦП

После выполнения команды могут появиться некоторые сообщения пользовательского интерфейса, которые представлены из-за различий между старым проектом и новым проектом, который был отправлен, или просто из-за изменения какой-либо переменной.

Если конфигурация Ethernet проекта отличается от устройства, связь может быть прервана в конце процесса загрузки, когда на устройстве применяется новая конфигурация. Таким образом, будет представлено следующее предупреждающее сообщение, предлагающее пользователю продолжить эту операцию или нет.

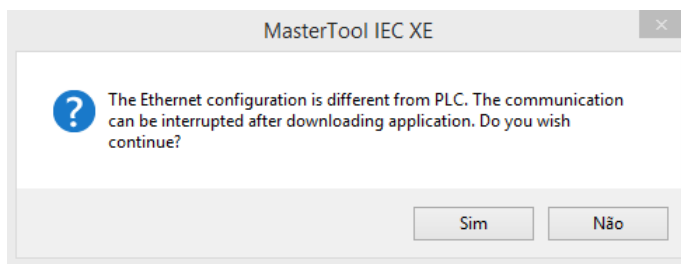


Рисунок 19: Предупреждение о конфигурации IP

Если на ЦП нет приложения, будет представлено следующее сообщение.



Рисунок 20: Нет приложения на устройстве

Если на ЦП уже есть приложение, в зависимости от различий между проектами будут представлены следующие варианты:

Вход с изменением онлайн: выполнить вход и отправить новый проект, не останавливая текущее приложение ЦП (см. пункт «Режим запуска»), обновляя изменения при выполнении нового цикла.

Вход с загрузкой: выполните вход и отправьте новый проект с остановленным ЦП (см. Режим остановки). Когда приложение инициализируется, обновление уже будет выполнено.

Вход без каких-либо изменений: выполняет вход без отправки нового проекта.

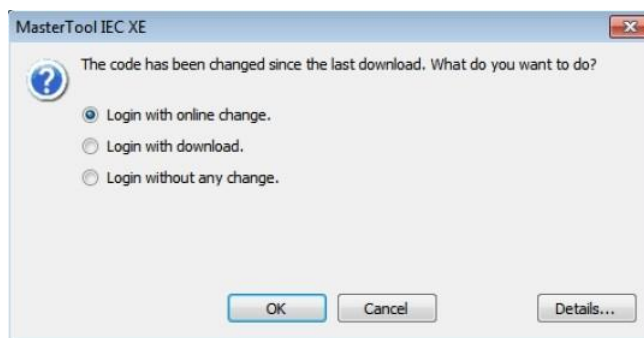


Рисунок 21: Обновление проекта на процессоре

**ВНИМАНИЕ**

В онлайн-изменениях не разрешается связывать отображение символических переменных из списка глобальных переменных (GVL) и использовать эти переменные в другом списке глобальных переменных (GVL).

Если новое приложение содержит изменения в конфигурации, онлайн-изменение будет невозможно. В этом случае MasterTool IEC XE запрашивает, должен ли вход в систему выполняться как загрузка (остановка приложения) или операция должна быть отменена.

PS.: Кнопка Подробнее... показывает изменения, внесенные в приложение.

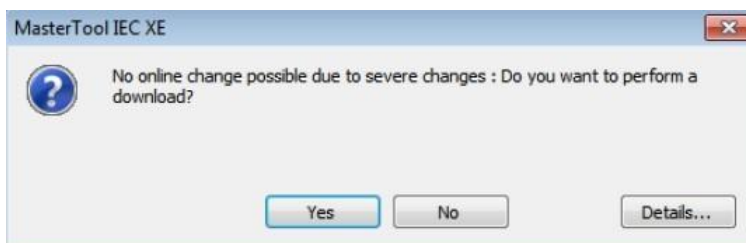


Рисунок 22: Изменение конфигурации

Наконец, в конце этого процесса MasterTool IEC XE предлагает возможность перенести (загрузить) исходный код во внутреннюю память устройства, как показано на следующем рисунке:

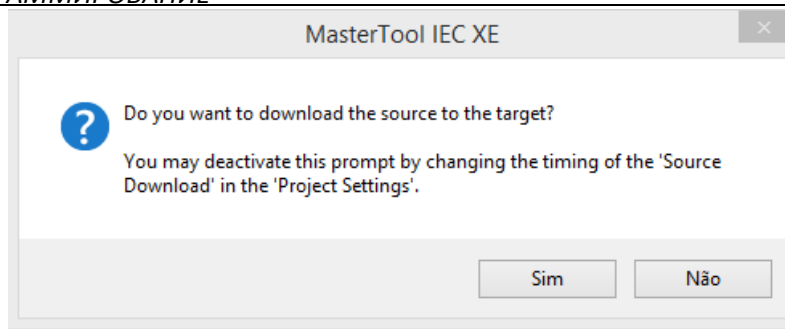


Рисунок 23: Загрузка исходного кода

Передача исходного кода имеет основополагающее значение для обеспечения будущего восстановления проекта и внесения изменений в приложение, загруженное в устройство.

## 4.8. Режим работы

Сразу после того, как проект будет отправлен в ЦП, приложение не будет немедленно выполнено (за исключением случая онлайн-изменения). Для этого должна быть выполнена команда Start. Таким образом, пользователь может контролировать выполнение приложения, отправленного в ЦП, что позволяет предварительно настроить начальные значения, которые будут использоваться ЦП в первом цикле выполнения.

Чтобы выполнить эту команду, просто перейдите в меню «Отладка» и выберите параметр «Запуск», как показано на рисунке ниже.

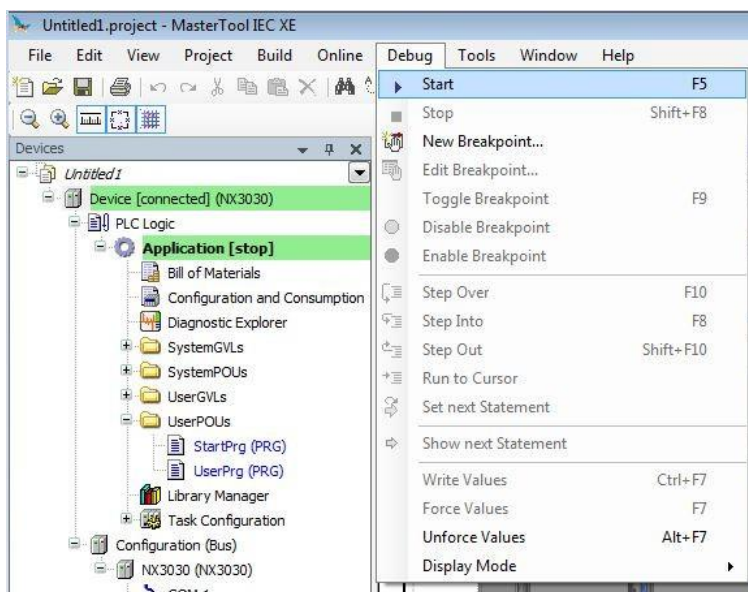


Рисунок 24: Запуск приложения

На рисунке ниже показано выполнение приложения. Если выбрана вкладка POU, созданные переменные перечислены в окне мониторинга, в котором значения могут быть визуализированы и принудительно заданы пользователем.

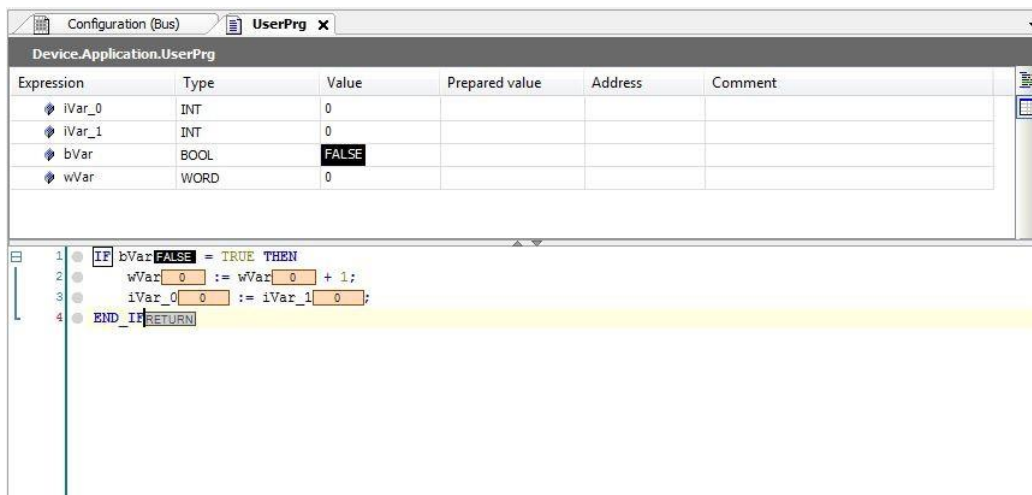


Рисунок 25: Запуск программы

Если в ЦП уже сохранено внутреннее загрузочное приложение, оно автоматически переходит в режим работы при включении устройства без необходимости онлайн-команды через MasterTool IEC XE.

## 4.9. Режим остановки

Чтобы остановить выполнение приложения, пользователь должен выполнить команду Stop, доступную в меню Debug, как показано на рисунке ниже.

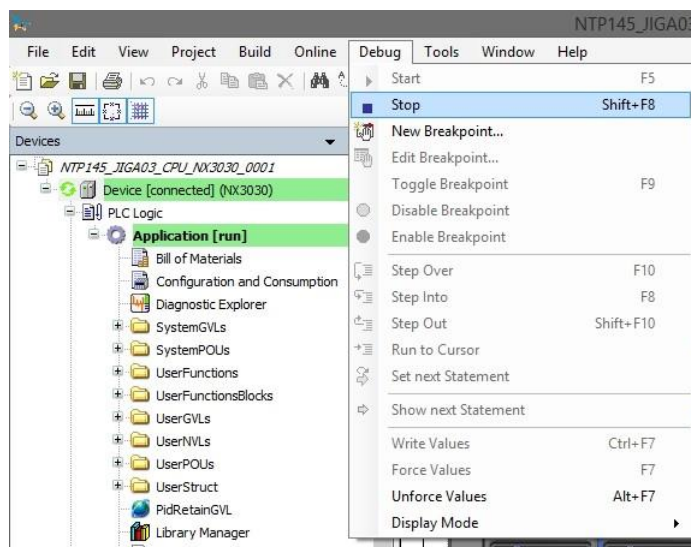


Рисунок 26: Остановка приложения

Если ЦП инициализируется без сохраненного приложения, он автоматически переходит в режим остановки, как это происходит при возникновении программного исключения.

## 4.10. Запись и принудительное использование переменных

После входа в ПЛК пользователь может записать или принудительно ввести значения в переменную проекта. Команда записи (CTRL + F7) записывает значение в переменную, и это значение может быть перезаписано инструкциями, выполняемыми в приложении.

Кроме того, команда принудительной записи (F7) записывает значение в переменную, не позволяя изменить это значение до тех пор, пока не будут освобождены принудительные переменные.

Важно отметить, что при использовании ведомого устройства MODBUS RTU и сервера MODBUS Ethernet и не выбранном параметре «Только чтение» из настроенных отношений команда принудительной записи (F7) должна выполняться над доступными переменными в мониторинге. окно, так как команда записи (CTRL + F7) оставляет переменные для перезаписи при выполнении новых считываний.

##### ВНИМАНИЕ

Когда ЦП с принудительными переменными и обесточен, переменные потеряют принудительное значение при следующей инициализации.

Предел форсирования для ЦП Nexto составляет 128 переменных, независимо от

##### ВНИМАНИЕ

Форсирование переменных может быть выполнено в онлайн-режиме.

Диагностические переменные не могут быть принудительно введены, только записаны, потому что диагностика предоставляется ЦП и будет перезаписана им.

### 4.11. Выход

Для завершения онлайн-связи с ЦП необходимо выполнить команду Logout (выход), расположенную в меню Online, как показано на рисунке ниже.

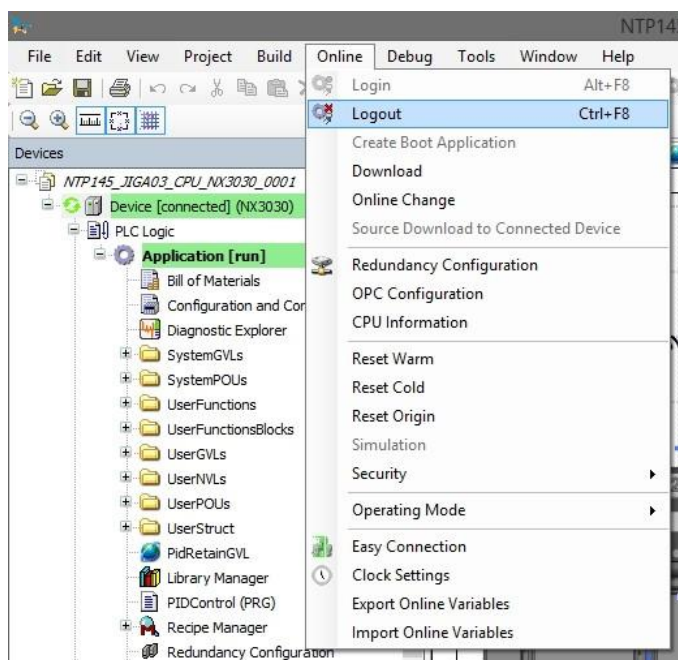


Рисунок 27: Завершение онлайн-связи с ЦП

### 4.12. Загрузка проекта

ЦП серии Nexto способны хранить исходный код приложения во внутренней памяти устройства, что позволяет в будущем извлекать (загружать) весь проект и изменять приложение.

Чтобы восстановить проект, ранее хранившийся во внутренней памяти ЦП, необходимо выполнить команду, расположенную в меню «Файл», как показано на следующем рисунке.

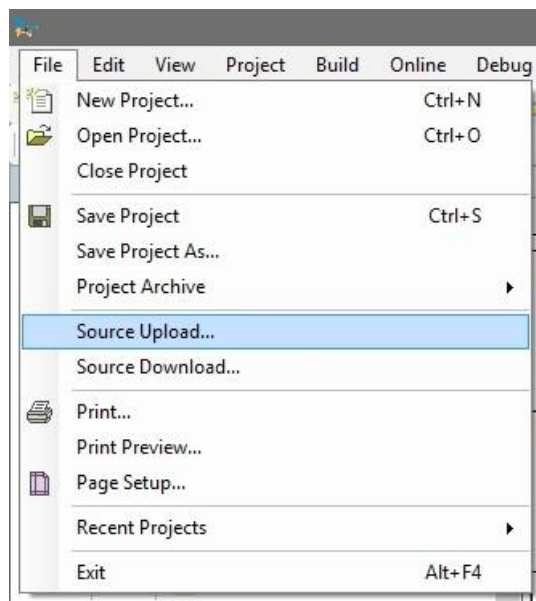


Рисунок 28: Вариант загрузки проекта

Далее просто выберите нужный процессор и нажмите ОК, как показано на рисунке ниже.

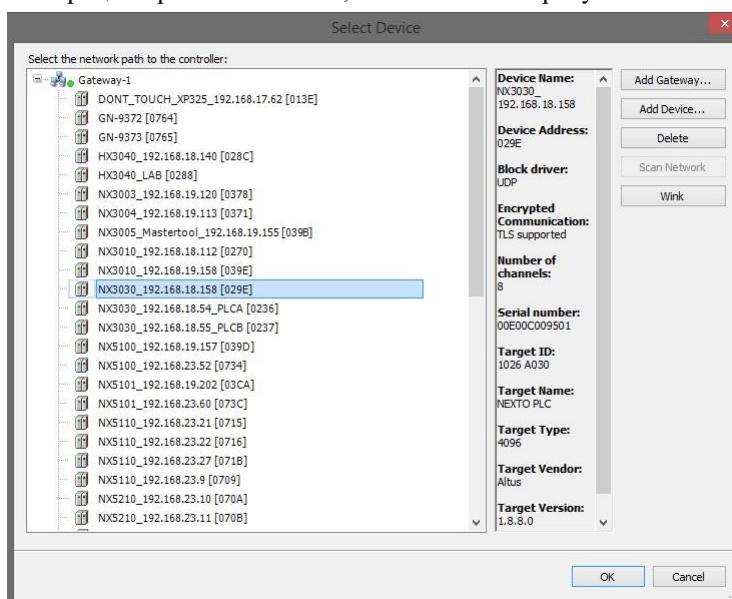


Рисунок 29: Выбор процессора

Чтобы убедиться, что проект, загруженный в ЦП, идентичен и может быть доступен на других рабочих станциях, см. главу

*Метод загрузки проектов/входа в систему без различий между проектами в руководстве пользователя MasterTool IEC XE MT8500 — MU299609.*

#### ВНИМАНИЕ

Размер области памяти для хранения проекта в ЦП Nexto определяется в разделе Память.

##### ВНИМАНИЕ

Загрузка восстанавливает последний проект, сохраненный в контроллере, как описано в предыдущих параграфах. В случае, если было загружено только приложение, без передачи его исходного кода, восстановление с помощью процедуры Upload будет невозможно.

### 4.13. Рабочие состояния процессора

#### 4.13.1. Запуск

Когда ЦП находится в режиме Run, выполняются все прикладные задачи.

#### 4.13.2. Остановка

Когда ЦП находится в режиме остановки, все прикладные задачи останавливаются. Значения переменных в задачах сохраняются с текущим значением, а точки вывода переходят в безопасное состояние.

Когда ЦП переходит в режим остановки из-за загрузки приложения, переменные в задачах приложения будут потеряны, за исключением типа постоянных переменных.

#### 4.13.3. Точка останова

Когда в задаче достигается отладочная отметка, она прерывается. Все активные задачи в приложении не будут прерываться, продолжая свое выполнение. С помощью этой функции можно шаг за шагом пройти и исследовать ход программы в онлайн-режиме в соответствии с позициями прерываний, включенных через редактор.

Дополнительные сведения об использовании точек останова см. в Руководстве по использованию MasterTool IEC XE — MU299609.

#### 4.13.4. Исключение

Когда процессор находится в состоянии исключения, это указывает на то, что в одной из активных задач приложения произошла какая-то неправильная операция. Задача, вызвавшая Exception, будет приостановлена, а остальные задачи перейдут в режим Stop. Выводить задачи из этого состояния и снова запускать их можно только после нового состояния запуска ЦП. Таким образом, только с теплым сбросом, холодным сбросом, сбросом источника или перезапуском ЦП приложение снова переводится в режим выполнения.

#### 4.13.5. Сброс настройки тепла

Эта команда переводит ЦП в режим остановки и инициализирует все переменные задач приложения, кроме постоянных и сохраняемых переменных типа. Переменные, инициализированные определенным значением, примут именно это значение, остальные переменные примут стандартное значение инициализации (ноль).

#### 4.13.6. Сброс настройки холода

Эта команда переводит ЦП в режим остановки и инициализирует все переменные задач приложения, кроме переменных постоянного типа. Переменные, инициализированные определенным значением, примут именно это значение, остальные переменные примут стандартное значение инициализации (ноль).

#### 4.13.7. Сбросить происхождение

Эта команда удаляет все переменные задачи приложения, включая переменные постоянного типа, удаляет ЦП приложения и переводит ЦП в режим остановки.

Примечания:

Сброс: при выполнении сброса точки останова, определенные в приложении, отключаются.

Команда: Для выполнения команд Reset Warm, Cold или Origin необходимо, чтобы MasterTool IEC XE находился в режиме онлайн с ЦП.

### 4.14. Программы (POU) и списки глобальных переменных (GVL)

Проект, созданный MasterTool IEC XE, содержит набор программных модулей (POU) и списки глобальных переменных, которые призваны облегчить программирование и использование контроллера. В следующих разделах описаны основные элементы, которые являются частью этой стандартной структуры проекта.

#### 4.14.1. Программа MainPrg

MainTask связан с одним уникальным программным модулем программного типа с именем MainPrg. Программа MainPrg создается автоматически и не может быть изменена пользователем.

#### 4. НАЧАЛЬНОЕ ПРОГРАММИРОВАНИЕ

---

Программный код MainPrg следующий, на языке ST:

```
(*Главный POU, связанный с MainTask, который вызывает StartPrg, UserPrg/ActivePrg и NonSkippedPrg.
```

```
Этот POU заблокирован для редактирования.*)
```

```
PROGRAM MainPrg
VAR

    isFirstCycle : BOOL := TRUE;
END_VAR

SpecialVariablesPrg();
IF isFirstCycle THEN

    StartPrg();
    isFirstCycle := FALSE;
```

MainPrg вызывает два других POU программного типа с именами StartPrg и UserPrg. Хотя UserPrg вызывается всегда,

StartPrg вызывается только один раз при запуске приложения ПЛК.

В отличие от программы MainPrg, которую нельзя модифицировать, пользователь может изменить программы StartPrg и UserPrg. Изначально при создании проекта из мастера эти две программы создаются пустыми, но пользователь может вставлять в них код.

##### 4.14.2. Программа StartPrg

В этом POU пользователь может создавать логику, циклы, стартовые переменные и т. д., которые будут выполняться только один раз в первом цикле ПЛК, прежде чем выполнить UserPrg POU в первый раз. И не вызывается снова во время выполнения проекта.

В случае, если пользователь загружает новое приложение, или если ПЛК выключается, а также в условиях Reset Origin, Reset Cold и Reset Warm, этот POU будет выполняться снова.

##### 4.14.3. Программа UserPrg

В этом POU пользователь должен создать главное приложение, отвечающее за собственное управление процессом. Этот POU вызывается основным POU (MainPrg).

Пользователь также может создавать дополнительные POU (программы, функции или функциональные блоки) и вызывать их или создавать их экземпляры внутри UserPrg POU до конца своей программной инструкции. Также можно вызывать функции и экземпляры функциональных блоков, определенные в библиотеках.

##### 4.14.4. Система GVL\_Diagnostics

System\_Diagnostics GVL содержит диагностические переменные ЦП, коммуникационного интерфейса (Ethernet и PROFIBUS) и всех коммуникационных драйверов. Этот GVL не является ediТаблица, и переменные объявляются автоматически с типом, указанным устройством, которому он принадлежит, когда он добавляется в проект.

**ВНИМАНИЕ**

В System\_Diagnostics GVL также объявлены диагностические переменные прямого представления MODBUS Client/Master Requests.

Некоторые устройства, такие как коммуникационный драйвер MODBUS Symbol, не имеют своих диагностических данных, размещенных в переменных %Q с помощью директивы AT. То же самое происходит с новейшими коммуникационными драйверами, такими как Server IEC 60870-5-104.

На следующем рисунке показан пример представления этого GVL в онлайн-режиме.

Device.Application.System_Diagnostics				
Expression	Type	Value	Address	Con
✱ DG_IEC_60870_5_104_Server	T_DIAG_IEC104_SERVER_1			DG_I
✱ DG_MODBUS_Symbol_Client	T_DIAG_MODBUS_ETH_CLIENT_1			DG_I
✱ tDiag	T_DIAG_MODBUS_DIAGNOSTICS_CLIENT			
✱ byDiag_1_reserved	BYTE	0		Rese
✱ tCommand	T_DIAG_MODBUS_COMMANDS			
✱ byDiag_3_reserved	BYTE	0		Rese
✱ tStat	T_DIAG_MODBUS_ETH_CLIENT_STATS			
✱ wTXRequests	WORD	1589		Coun
✱ wRXNormalResponses	WORD	1589		Coun
✱ wRXExceptionResponses	WORD	0		Coun
✱ wRXIllegalResponses	WORD	0		Coun
✱ wDiag_12_reserved	WORD	0		Rese
✱ wDiag_14_reserved	WORD	0		Rese
✱ wDiag_16_reserved	WORD	0		Rese
✱ wDiag_18_reserved	WORD	0		Rese
✱ DG_MODBUS_Symbol_Client_NX5000	T_DIAG_MODBUS_ETH_CLIENT_1			DG_I
✱ DG_MODBUS_Symbol_RTU_Master	T_DIAG_MODBUS_RTU_MASTER_1			DG_I
✱ DG_MODBUS_Symbol_Server_NX5000	T_DIAG_MODBUS_ETH_SERVER_1			DG_I
✱ DG_NX3030	T_DIAG_NX3030_1		%QB66229	DG_I
✱ tSummarized	T_DIAG_SUMMARIZED_1			
✱ tDetailed	T_DIAG_DETAILED_1			
✱ DG_NX5001	T_DIAG_NX5001_1		%QB66922	DG_I
✱ DG_MODBUS_Client	T_DIAG_MODBUS_ETH_CLIENT_1		%QB67191	DG_I
✱ DG_MBUS_Direct_1_Mapping_000	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67211	DG_I
✱ byStatus	T_DIAG_MODBUS_ETH_MAPPING_STAT...			
✱ bCommIdle	BIT	FALSE		Comr
✱ bCommExecuting	BIT	FALSE		Comr
✱ bCommPostponed	BIT	TRUE		Comr
✱ bCommDisabled	BIT	FALSE		Comr
✱ bCommOk	BIT	TRUE		Previ
✱ bCommError	BIT	FALSE		Previ
✱ bCommAborted	BIT	FALSE		Previ
✱ bDiag_7_reserved	BIT	FALSE		Rese
✱ eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR		Last
✱ eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION		Last
✱ byDiag_3_reserved	BYTE	0		reser
✱ wCommCounter	WORD	397		Coun
✱ wCommErrorCounter	WORD	0		Coun
✱ DG_MBUS_Direct_1_Mapping_001	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67219	DG_I
✱ DG_MBUS_Direct_1_Mapping_003	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67235	DG_I
✱ DG_MBUS_Direct_1_Mapping_002	T_DIAG_MODBUS_ETH_MAPPING_1		%QB67243	DG_I
✱ DG_NX5000	T_DIAG_NX5000_1		%QB67251	DG_I

Рисунок 30: System\_Diagnostics GVL в онлайн-режиме

#### 4.14.5. Отключение GVL

Disables GVL содержит MODBUS Master/Client путем символического сопоставления переменных отключения запроса. Это не обязательно, но не рекомендуется использовать автоматическую генерацию этих переменных, которая осуществляется нажатием на кнопку Generate Disabling Variables на вкладке заявки на устройство. Эти переменные объявлены как тип BOOL и имеют следующую структуру:

Запрос отключения объявления переменных:

```
[Device Name]_DISABLE_[Requisition Number] : BOOL;
```

Где:

**Имя устройства:** Имя, которое отображается в дереве для устройства MODBUS.

**Номер заявки:** Номер заявки, указанный в таблице заявки на устройство MODBUS в порядке сверху вниз, начиная с 0001.

Пример:

Device.Application.Disable

```
VAR_GLOBAL
```

```
MODBUS_Device_DISABLE_0001 : BOOL;
MODBUS_Device_DISABLE_0002 : BOOL;
MODBUS_Device_DISABLE_0003 : BOOL;
MODBUS_Device_1_DISABLE_0001 : BOOL;
MODBUS_Device_1_DISABLE_0002 : BOOL;
```

s

Автоматическая генерация с помощью кнопки Generate Disabling Variables только создает переменные, но не удаляет их автоматически. Таким образом, в случае удаления какой-либо связи соответствующая отключающая переменная должна быть удалена вручную.

Disables GVL является ediТаблица, поэтому переменные отключения заявки могут быть созданы вручную без необходимости следования модели, созданной автоматическим объявлением, и могут использоваться в обоих направлениях одновременно, но всегда должны иметь тип BOOL. И необходимо позаботиться о том, чтобы не удалять и не изменять автоматически объявленные переменные, поскольку они могут использоваться для некоторых устройств MODBUS. Если переменная будет удалена или изменена, во время компиляции проекта будет сгенерирована ошибка. Чтобы исправить автоматически объявленное имя переменной, необходимо следовать модели, приведенной выше, в соответствии с устройством и заявкой, к которой они относятся.

На следующем рисунке показан пример представления этого GVL в онлайн-режиме. Если значения переменных TRUE, это означает, что заявка, к которой принадлежат переменные, отключена, и наоборот, когда значение переменной равно FALSE.









Device.Application.Disables			
Expression	Type	Value	Prepared
 MODBUS_Slave_1_DISABLE_0001	BOOL	FALSE	
 MODBUS_Slave_1_DISABLE_0002	BOOL	TRUE	
 MODBUS_Slave_1_DISABLE_0003	BOOL	FALSE	
 MODBUS_Slave_1_DISABLE_0004	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0001	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0002	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0003	BOOL	FALSE	
 MODBUS_Server_1_DISABLE_0004	BOOL	TRUE	

Рисунок 31: Отключить GVL в онлайн-режиме

#### 4.14.6. GVL IOQualities

IOQualities GVL содержит переменные качества модулей ввода-вывода, объявленных на шине ЦП. Этот GVL не является ediТаблица, и переменные автоматически объявляются как массивы типов LibDataTypes.QUALITY, а размеры соответствуют количеству ввода-вывода модуля, которому он принадлежит, когда тот добавляется в проект.

Пример: Device.Application.IOQualities

## VAR\_GLOBAL

```

QUALITY_NX1001: ARRAY[0..15] OF LibDataTypes.QUALITY;
QUALITY_NX2020: ARRAY[0..15] OF LibDataTypes.QUALITY;
QUALITY_NX6000: ARRAY[0..7] OF LibDataTypes.QUALITY;
QUALITY_NX6100: ARRAY[0..3] OF LibDataTypes.QUALITY;

```

Когда приложение находится в режиме RUN, можно просмотреть значения переменных качества модулей ввода-вывода, которые были добавлены в проект через IOQualities GVL.

## 4.14.7. Module\_Diagnostics GVL

Module\_Diagnostics GVL содержит диагностические переменные модулей ввода-вывода, используемых в проекте, за исключением ЦП и коммуникационных драйверов. Этот GVL не является ediТаблица, и переменные автоматически объявляются с типом, указанным модулем, к которому он принадлежит, когда тот добавляется в проект.

На следующем рисунке показан пример представления этого GVL в онлайн-режиме.

Device.Application.Module_Diagnostics				
Expression	Type	Value	Address	Comment
[-] DG_NX1001	T_DIAG_NX1001_1		%QB67008	DG_NX1001 diagnostics variable
[-] tGeneral	T_DIAG_GENERAL_NX1001_1			
[-] bReserved_8	BIT	FALSE		Reserved
[-] bReserved_9	BIT	FALSE		Reserved
[-] bReserved_10	BIT	FALSE		Reserved
[-] bReserved_11	BIT	FALSE		Reserved
[-] bReserved_12	BIT	FALSE		Reserved
[-] bReserved_13	BIT	FALSE		Reserved
[-] bReserved_14	BIT	FALSE		Reserved
[-] bReserved_15	BIT	FALSE		Reserved
[-] bActiveDiagnostics	BIT	FALSE		Module has active diagnostics
[-] bFatalError	BIT	FALSE		Module has fatal error
[-] bConfigMismatch	BIT	FALSE		Module has parameterization error
[-] bWatchdogError	BIT	FALSE		Module has watchdog expired
[-] bOTDSwitchError	BIT	FALSE		Module one touch diag switch error
[-] bReserved_5	BIT	FALSE		Reserved
[-] bReserved_6	BIT	FALSE		Reserved
[-] bReserved_7	BIT	FALSE		Reserved
[+] DG_NX1005	T_DIAG_NX1005_1		%QB67010	DG_NX1005 diagnostics variable
[+] DG_NX2001	T_DIAG_NX2001_1		%QB67014	DG_NX2001 diagnostics variable
[+] DG_NX2020	T_DIAG_NX2020_1		%QB67018	DG_NX2020 diagnostics variable
[+] DG_NX6000	T_DIAG_NX6000_1		%QB67022	DG_NX6000 diagnostics variable
[-] DG_NX6100	T_DIAG_NX6100_1		%QB67040	DG_NX6100 diagnostics variable
[-] tGeneral	T_DIAG_GENERAL_NX6100_1			
[-] bActiveDiagnosticsOutput00	BIT	FALSE		Output 00 with diagnostics
[-] bActiveDiagnosticsOutput01	BIT	FALSE		Output 01 with diagnostics
[-] bActiveDiagnosticsOutput02	BIT	FALSE		Output 02 with diagnostics
[-] bActiveDiagnosticsOutput03	BIT	FALSE		Output 03 with diagnostics
[-] bReserved_12	BIT	FALSE		Reserved
[-] bReserved_13	BIT	FALSE		Reserved
[-] bReserved_14	BIT	FALSE		Reserved
[-] bReserved_15	BIT	FALSE		Reserved
[-] bActiveDiagnostics	BIT	FALSE		Module has active diagnostics
[-] bFatalError	BIT	FALSE		Module has fatal error
[-] bConfigMismatch	BIT	FALSE		Module has parameterization error
[-] bWatchdogError	BIT	FALSE		Module has watchdog expired
[-] bOTDSwitchError	BIT	FALSE		Module one touch diag switch error
[-] bCalibrationError	BIT	FALSE		Module has calibration error
[-] bNoExternalSupply	BIT	FALSE		External power s...y is below the ...
[-] bReserved_07	BIT	FALSE		Reserved
[-] tDetailed	T_DIAG_DETAILED_NX6100_1			
[-] tAnalogOutput_00	T_DIAG_ANALOG_OUTPUT			
[-] tAnalogOutput_01	T_DIAG_ANALOG_OUTPUT			
[-] tAnalogOutput_02	T_DIAG_ANALOG_OUTPUT			

Рисунок 32: Module\_Diagnostics GVL в онлайн режиме

#### 4.14.8. Качества GVL

Качественный GVL содержит переменную качества внутренних переменных MODBUS Master/Client символического отображения. Не обязательно, но рекомендуется использовать автоматическую генерацию этих переменных, что делается нажатием на кнопку Generate Quality Variables на вкладке сопоставления устройств. Эти переменные объявлены как тип LibDataTypes.QUALITY и имеют следующую структуру:

Объявление переменной сопоставления качества:

```
[Device Name]_QUALITY_[Mapping Number]: LibDataTypes.QUALITY;
```

Где:

**Наименование устройства:** Имя, которое отображается в дереве устройства.

**Номер сопоставления:** номер сопоставления, который был объявлен в таблице сопоставления устройства, в порядке возрастания и убывания, начиная с 0001.

#### ВНИМАНИЕ

Невозможно связать переменные качества с прямым представлением отображений ведущих/клиентских драйверов MODBUS. Поэтому рекомендуется использовать символьное отображение драйверов MODBUS.

Пример: Device.Application.Qualities

VAR\_GLOBAL

```
MODBUS_Device_QUALITY_0001: LibDataTypes.QUALITY;
MODBUS_Device_QUALITY_0002: LibDataTypes.QUALITY;
MODBUS_Device_QUALITY_0003: LibDataTypes.QUALITY;
```

Качественный GVL является едтаблицей, поэтому переменные качества сопоставления могут быть созданы вручную без необходимости следовать модели автоматического объявления и могут использоваться в обоих направлениях одновременно. Но всегда должен иметь тип LibDataTypes.QUALITY и следить за тем, чтобы не удалять и не изменять автоматически объявленные переменные, потому что они могут использоваться каким-то устройством. Если переменная будет удалена или изменена, во время компиляции проекта будет сгенерирована ошибка. Чтобы исправить автоматически объявленное имя переменной, необходимо следовать модели, приведенной выше, в соответствии с устройством и заявкой, к которой они относятся.

Для устройств связи MODBUS переменные качества ведут себя так, как показано в Таблице 45. На следующем рисунке показан пример представления этого GVL в режиме Online.

#### ВНИМАНИЕ

Если переменная символического отображения клиент/ведущий драйвер MODBUS отображается в драйвере сервера IEC 60870-5-104, необходимо, чтобы переменные качества отображения MODBUS были созданы для генерации действительных событий качества для таких точек сервера IEC 60870-5-104. Наоборот, не будут генерироваться события «плохого» качества для клиентов сервера IEC 60870-5-104 в ситуациях, когда ведущий/клиент MODBUS не может обмениваться данными со своими подчиненными/серверами, например...

Device.Application.Qualities				
Expression	Type	Value	Address	Comment
MODBUS_Slave_1_QUALITY_0001	LibDataTypes.QUALITY			
VALIDITY	QUALITY_VALIDITY	VALIDITY_GOOD		Quality validity
FLAGS	QUALITY_FLAGS			Quality flags
FLAG_OUT_OF_RANGE	BIT	FALSE		Bit 8
FLAG_INACCURATE	BIT	FALSE		Bit 9
FLAG_OLD_DATA	BIT	FALSE		Bit 10
FLAG_FAILURE	BIT	FALSE		Bit 11
FLAG_OPERATOR_BLOCKED	BIT	FALSE		Bit 12
FLAG_TEST	BIT	FALSE		Bit 13
FLAG_RESERVED_0	BIT	FALSE		Bit 14
FLAG_RESERVED_1	BIT	FALSE		Bit 15
FLAG_RESTART	BIT	FALSE		Bit 0
FLAG_COMM_FAIL	BIT	FALSE		Bit 1
FLAG_REMOTE_SUBSTITU...	BIT	FALSE		Bit 2
FLAG_LOCAL_SUBSTITUTED	BIT	FALSE		Bit 3
FLAG_FILTER	BIT	FALSE		Bit 4
FLAG_OVERFLOW	BIT	FALSE		Bit 5
FLAG_REFERENCE_ERROR	BIT	FALSE		Bit 6
FLAG_INCONSISTENT	BIT	FALSE		Bit 7
MODBUS_Slave_1_QUALITY_0002	LibDataTypes.QUALITY			
MODBUS_Slave_1_QUALITY_0003	LibDataTypes.QUALITY			
MODBUS_Slave_1_QUALITY_0004	LibDataTypes.QUALITY			
MODBUS_Server_1_QUALITY_0001	LibDataTypes.QUALITY			
MODBUS_Server_1_QUALITY_0002	LibDataTypes.QUALITY			
MODBUS_Server_1_QUALITY_0003	LibDataTypes.QUALITY			
VALIDITY	QUALITY_VALIDITY	VALIDITY_QUESTIONABLE		Quality validity
FLAGS	QUALITY_FLAGS			Quality flags
FLAG_OUT_OF_RANGE	BIT	FALSE		Bit 8
FLAG_INACCURATE	BIT	FALSE		Bit 9
FLAG_OLD_DATA	BIT	TRUE		Bit 10
FLAG_FAILURE	BIT	FALSE		Bit 11
FLAG_OPERATOR_BLOCKED	BIT	FALSE		Bit 12
FLAG_TEST	BIT	FALSE		Bit 13
FLAG_RESERVED_0	BIT	FALSE		Bit 14
FLAG_RESERVED_1	BIT	FALSE		Bit 15
FLAG_RESTART	BIT	FALSE		Bit 0
FLAG_COMM_FAIL	BIT	TRUE		Bit 1
FLAG_REMOTE_SUBSTITU...	BIT	FALSE		Bit 2
FLAG_LOCAL_SUBSTITUTED	BIT	FALSE		Bit 3
FLAG_FILTER	BIT	FALSE		Bit 4
FLAG_OVERFLOW	BIT	FALSE		Bit 5
FLAG_REFERENCE_ERROR	BIT	FALSE		Bit 6
FLAG_INCONSISTENT	BIT	FALSE		Bit 7
MODBUS_Server_1_QUALITY_0004	LibDataTypes.QUALITY			

Рисунок 33: Качества ГВЛ в онлайн-режиме

#### 4.14.9. GVL ReqDiagnostics

ReqDiagnostics GVL содержит диагностические переменные заявки символьного отображения MODBUS Master/Client. Не обязательно, но рекомендуется использовать автоматическую генерацию этих переменных, что делается нажатием кнопки Generate Diagnostics Variables на вкладке запросов устройств. Эти объявления переменных имеют следующую структуру:

Объявление диагностической переменной заявки:

```
[Device Name]_REQDG_[Requisition Number]: [Variable Type];
```

Где:

**Наименование устройства:** Имя, которое отображается в дереве устройства.

**Номер сопоставления:** номер сопоставления, который был объявлен в таблице сопоставления устройства, в порядке возрастания и убывания, начиная с 0001.

**Тип переменной:** NXMODBUS\_DIAGNOSTIC\_STRUCTS.T\_DIAG\_MODBUS\_RTU\_MAPPING\_1 для главного устройства MODBUS и NXMODBUS\_DIAGNOSTIC\_STRUCTS.T\_DIAG\_MODBUS\_ETH\_MAPPING\_1 для клиента MODBUS.

ВНИМАНИЕ

Переменные диагностики заявки прямого сопоставления MODBUS Master/Client объявляются в System Diagnostics GVL.

Пример:  
Device.Application.ReqDiagnostics

```
VAR_GLOBAL

MODBUS_Device_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_RTU_MAPPING_1;
MODBUS_Device_REQDG_0002 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_RTU_MAPPING_1;
MODBUS_Device_REQDG_0003 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_RTU_MAPPING_1;
MODBUS_Device_1_REQDG_0001 : NXMODBUS_DIAGNOSTIC_STRUCTS.
                                T_DIAG_MODBUS_ETH_MAPPING_1;
```

ReqDiagnostics GVL является дополняемой, поэтому диагностические переменные заявок могут быть созданы вручную без необходимости следовать модели, созданной автоматическим объявлением. Оба способа могут использоваться одновременно, но переменные всегда должны иметь тип, относящийся к устройству. И позаботьтесь о том, чтобы не удалять и не изменять автоматически объявленные переменные, потому что они могут использоваться каким-то устройством. Если переменная удалена или изменена, во время компиляции проекта будет сгенерирована ошибка. Чтобы исправить автоматически объявленное имя переменной, необходимо следовать модели, приведенной выше, в соответствии с устройством и заявкой, к которой они относятся.

На следующем рисунке показан пример представления этого GVL в онлайн-режиме.

Device.Application.ReqDiagnostics		
Expression	Type	Value
MODBUS_Slave_1_REQDG_0001	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
byStatus	T_DIAG_MODBUS_RTU_MAPPING_STATUS	
eLastErrorCode	MASTER_ERROR_CODE	NO_ERROR
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	969
wCommErrorCounter	WORD	0
MODBUS_Slave_1_REQDG_0002	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Slave_1_REQDG_0003	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Slave_1_REQDG_0004	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Server_1_REQDG_0001	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Server_1_REQDG_0002	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
MODBUS_Server_1_REQDG_0003	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	
byStatus	T_DIAG_MODBUS_ETH_MAPPING_STATUS	
eLastErrorCode	MASTER_ERROR_CODE	ERR_CONNECTION_TIMEOUT
eLastExceptionCode	MODBUS_EXCEPTION	NO_EXCEPTION
byDiag_3_reserved	BYTE	0
wCommCounter	WORD	116
wCommErrorCounter	WORD	49
MODBUS_Server_1_REQDG_0004	NXMODBUS_DIAGNOSTIC_STRUCTS.T_DIAG_MODBUS...	

Рисунок 34: ReqDiagnostics GVL в интерактивном режиме

## 5. Конфигурация

ЦП серии Nexто настроены и запрограммированы с помощью программного обеспечения MasterTool IEC XE. Выполненная конфигурация определяет поведение и режимы использования периферийных устройств, а также специальные функции ЦП. Программа представляет собой Приложение, разработанное пользователем.

### 5.1. Конфигурация ЦП

#### 5.1.1. Общие параметры

Приведенные ниже параметры являются частью конфигурации ЦП, включенной в приложение. Каждый элемент должен быть должным образом проверен для правильного выполнения проекта.

Помимо этих параметров, можно изменить имя каждого модуля, вставленного в приложение, нажав правую кнопку на модуле. В пункте «Свойства» на листе «Общие» измените имя, которое ограничено 24 символами.

Настройки	Описание	Стандарт	Варианты
Адрес начала %Q	<b>Зона диагностики (%Q)</b>		
	Начальный адрес диагностики UCP (%Q)	Автоматически выделяется при создании проекта.	0 до 64682
Размер	Размер области диагностики в байтах	1253	Невозможно изменить размер области диагностики процессора.
Адрес начала %Q	<b>Удерживающая зона (%Q)</b>		
	Начальный адрес сохраняемой памяти данных (%Q)	4096	0 до 65535
Размер	Сохранить размер памяти данных в байтах	2048	0 до 65535
Адрес начала %Q	<b>Постоянная область (%Q)</b>		
	Начальный адрес сохраняемой памяти данных (%Q)	12288	0 до 65535
Размер	Сохранить размер памяти данных в байтах	2048	0 до 65535
Запустить пользовательское приложение после сброса Таймером контрольной системы	<b>Параметры процессора</b>		
	При включении запускает пользовательское приложение после сброса аппаратного таймера контрольной системы или перезапуска Runtime, но с сохранением диагностической индикации через светодиод WD и через переменные.	Выключено	Включено выключено

Настройки	Описание	Стандарт	Варианты
Горячая замена	Режим горячей замены модуля	Enabled, without startup consistency (may vary according to CPU model)	<ul style="list-style-type: none"> <li>- Отключено, только для заявленных модулей</li> <li>- Отключено (с согласованностью запуска)</li> <li>- Отключено, без согласованности при запуске</li> <li>- Включено, с согласованностью запуска только для объявленных модулей</li> <li>- Включено, с согласованностью запуска</li> <li>- Включено, без согласованности при запуске</li> </ul>
Включить обновление ввода-вывода для каждой задачи	Параметры проекта		
	Настройка обновления входов и выходов в задачах, в которых они используются.	Не отмечено	<ul style="list-style-type: none"> <li>- Отмечено: входы и выходы обновляются задачами, в которых они используются.</li> <li>- Не отмечено: входы и выходы обновляются только MainTask.</li> </ul>
Включить внутреннее завершение	Параметры порта CAN		
	Если включено, используйте внутреннюю терминацию на интерфейсе CAN.	Не отмечено	<ul style="list-style-type: none"> <li>- Отмечено: внутренняя терминация на интерфейсе CAN включена.</li> <li>- Не отмечено: внутренняя терминация на интерфейсе CAN отключена.</li> </ul>

Таблица 37: Настройки процессора

**ВНИМАНИЕ**

Когда начальный адрес или размер сохраняемой или постоянной памяти данных изменяется в пользовательском приложении, память перераспределяется, что очищает область сохраняемых и постоянных переменных. Так что пользователю приходится быть осторожным, чтобы не потерять сохраненные данные в памяти.

**ВНИМАНИЕ**

В ситуациях, когда область символической постоянной памяти изменяется, программатор MasterTool IEC XE отображает сообщение, чтобы выбрать поведение для этой области после загрузки модифицированной программы. Выбор такого поведения не влияет на постоянную область прямого представления, которая всегда чиста.

**ВНИМАНИЕ**

Параметр «Включить обновление ввода-вывода для каждой задачи» не поддерживается ведущими устройствами полевой шины, такими как модуль NX5001. Эта функция применима только для модулей ввода и вывода, присутствующих на локальной шине контроллера (основная стойка и стойки расширения).

**ВНИМАНИЕ**

Даже когда точка ввода-вывода используется в других задачах, с отмеченным параметром Включить обновление ввода-вывода для каждой задачи, она будет продолжать обновляться и в основной задаче, за исключением случаев, когда все точки модуля используются в какой-либо другой задаче, в этом случае они больше не будут обновляться на MainTask.

## 5. КОНФИГУРАЦИЯ

### 5.1.1.1. Горячая замена

ЦП серии Nexто имеют возможность смены модулей ввода/вывода в шине без необходимости отключения системы и без потери информации. Эта функция известна как «горячая замена».

При горячей замене соответствующее поведение системы изменяется в соответствии с таблицей конфигурации,

#### ВНИМАНИЕ

ЦП серии Nexто не гарантируют сохранность постоянных и сохраняемых переменных в случае, если источник питания или даже ЦП извлекаются из стойки объединительной платы, находящейся под напряжением.

определенной пользователем, которая представляет следующие параметры:

Отключено, только для объявленных модулей Отключено (с согласованностью запуска) Отключено, без согласованности запуска

Включено, с согласованностью запуска только для объявленных модулей Включено, с согласованностью запуска

Включено, без согласованности при запуске

Таким образом, пользователь может выбрать поведение, которое система должна принять в нештатных ситуациях на шине и когда ЦП находится в режиме работы. В приведенной ниже Таблице представлены возможные нештатные ситуации на шине.

Ситуация	Возможные причины
Несовместимая конфигурация	- Какой-то модуль, подключенный к шине, отличается от той модели, которая заявлена в конфигурации.
отсутствующий модуль	- Модуль снят с шины. - Какой-то неисправный модуль не отвечает процессору - Некоторые позиции автобуса работают со сбоями.

Таблица 38: Нештатные ситуации с автобусами

Для получения дополнительной информации о диагностике, соответствующей описанным выше ситуациям, см. Диагностика через переменные.

Если модуль присутствует в определенной позиции, в которой не должно быть согласно конфигурации модулей, этот модуль считается недеklarированным. Параметры горячей замены отключены, только для объявленных модулей и включены, с согласованностью запуска только для объявленных модулей не учитывают модули, находящиеся в этом состоянии.

#### 5.1.1.1.1. Горячая замена отключена, только для заявленных модулей

В этой конфигурации ЦП немедленно переходит в режим остановки при возникновении нештатной ситуации на шине (как описано в Таблице 38). Светодиод DG начинает мигать 4 раза (согласно Таблице 39). В этом случае, чтобы заставить ЦП вернуться в нормальное состояние Run, в дополнение к отмене того, что вызвало нештатную ситуацию, необходимо выполнить Reset Warm или Reset Cold. Если выполняется сброс источника, необходимо будет выполнить загрузку, чтобы ЦП мог вернуться в нормальное состояние (Выполнение). Команды Reset Warm, Reset Cold и Reset Origin можно выполнить с помощью MasterTool IEC XE в онлайн-меню.

ЦП останется в нормальном режиме работы, даже если на шине будет обнаружен модуль, не объявленный.

#### 5.1.1.1.2. Горячая замена отключена

Эта настройка не допускает возникновения нештатных ситуаций в шине (как показано в Таблице 38) модулей, в том числе необъявленных и присутствующих на шине. Процессор переходит в режим Stop, и светодиод DG начинает мигать 4 раза (как в Таблице 39). В этих случаях, чтобы вернуть ЦП в нормальный режим работы, в дополнение к устранению причин, вызвавших нештатную ситуацию, необходимо выполнить теплый сброс или холодный сброс. Если сброс исходной точки выполнен, вам необходимо загрузить проект, чтобы ЦП мог вернуться в обычный режим работы. Команды Reset Warm, Reset Cold и Reset Origin можно выполнить с помощью MasterTool IEC XE в онлайн-меню.

#### 5.1.1.1.3. Горячая замена отключена, без согласованности при запуске

Позволяет системе запускаться даже тогда, когда какой-либо модуль находится в нештатной ситуации на шине (как показано в Таблице 38). О нештатных ситуациях сообщается посредством диагностики.

Любая модификация шины приведет к тому, что ЦП перейдет в режим остановки, а светодиод DG начнет мигать 4 раза (как в Таблице 39). Чтобы в этих случаях ЦП вернулся в нормальное состояние Run, необходимо выполнить теплый сброс или холодный сброс. Если выполняется сброс источника, необходимо будет загрузить ЦП, чтобы ЦП

## 5. КОНФИГУРАЦИЯ

мог вернуться в нормальное состояние выполнения. Команды Reset Warm, Reset Cold и Reset Origin можно выполнить с помощью MasterTool IEC XE в онлайн-меню.

### 5.1.1.1.4. Включена горячая замена, с согласованностью запуска только для заявленных модулей

«Запуск» — это интервал между подачей питания на ЦП (или командой сброса, или загрузкой приложения) до первого перехода ЦП в рабочий режим после включения. Эта конфигурация проверяет, не возникла ли какая-либо нештатная ситуация на шине (как описано в Таблице 38) во время запуска. В положительном случае процессор переходит в режим остановки и светодиод DG начинает мигать 4 раза (согласно Таблице 39). После этого, чтобы перевести ЦП в режим Run, а также для устранения причин, вызвавших нештатную ситуацию, необходимо выполнить команду Reset Warm или Reset Cold, которую можно выполнить с помощью MasterTool IEC XE (меню Online). Если выполняется сброс источника, необходимо будет выполнить загрузку, чтобы ЦП мог вернуться в нормальное состояние (Выполнение).

После запуска, если в каком-либо модуле возникнет какая-либо ситуация, описанная в предыдущей таблице, система будет продолжать работать в обычном режиме и сигнализировать о проблеме с помощью диагностики.

Если для заявленных модулей нет других отклонений от нормы, ЦП перейдет в нормальное состояние (Run), даже если на шине присутствует необъявленный модуль.

#### ВНИМАНИЕ

В этой конфигурации, когда происходит сбой питания (даже временный), выполняется команда «Теплый сброс», «Холодный сброс» или загрузка нового приложения, и если какой-либо модуль находится в ненормальной ситуации на шине, ЦП переходит в режим останова и отключается. светодиод DG начнет мигать 4 раза (согласно таблице 39). Это считается стартовой ситуацией.

Это наиболее рекомендуемый вариант, поскольку гарантирует целостность системы при ее инициализации и позволяет изменять модули при работающей системе.

### 5.1.1.1.5. Горячая замена включена с согласованностью запуска

Эта настройка проверяет, не было ли каких-либо нештатных ситуаций в шине (как показано в Таблице 38) во время запуска, даже если на шине нет объявленных и присутствующих модулей; если да, то CPU переходит в режим Stop и светодиод DG начинает мигать 4 раза (как показано в Таблице 39). В этих случаях, чтобы вернуть ЦП в нормальный режим работы, в дополнение к устранению причин, вызвавших нештатную ситуацию, необходимо выполнить теплый сброс или холодный сброс. Если сброс исходной точки выполнен, вам необходимо загрузить проект, чтобы ЦП мог вернуться в обычный режим работы. Команды Reset Warm, Reset Cold и Reset Origin можно выполнить с помощью MasterTool IEC XE в онлайн-меню.

### 5.1.1.1.6. Горячая замена включена без согласованности при запуске

Позволяет системе начать работу, даже если модуль находится в нештатной ситуации на шине (как описано в Таблице 38). О нештатных ситуациях сообщается посредством диагностики во время и после запуска.

#### ВНИМАНИЕ

Этот вариант рекомендуется на этапе внедрения системы, поскольку он позволяет загружать новые приложения и отключать питание без наличия всех настроенных модулей.

### 5.1.1.1.7. Как сделать горячую замену

#### ПРЕДУПРЕЖДЕНИЕ

Перед выполнением «горячей замены» важно снять всю возможную статическую энергию, накопленную в теле. Для этого коснитесь (голыми руками) любой заземленной металлической поверхности, прежде чем брать за модули. Такая процедура гарантирует, что пределы статической энергии модуля не будут

**ВНИМАНИЕ**

Рекомендуется контролировать диагностику горячей замены в системе управления приложениями, разработанной пользователем, чтобы гарантировать, что значение, возвращаемое модулем, проверяется перед использованием.

Процедура горячей замены описана ниже:

Отсоедините модуль от стойки объединительной платы, используя предохранительный замок. Снимите модуль, сильно потянув.

Вставьте новый модуль в стойку объединительной платы.

Убедитесь, что замок безопасности полностью подключен. При необходимости сильнее прижмите модуль к стойке объединительной платы.

В случае выходных модулей удобно отсоединять точки в процессе замены, чтобы уменьшить образование дуги в разъеме модуля. Это необходимо сделать, отключив питание или форсировав точки вывода с помощью программных средств. Если нагрузка небольшая, отключение не требуется.

Важно отметить, что в случаях, когда ЦП переходит в режим остановки и светодиод DG начинает мигать 4 раза (согласно Таблице 39, из-за какой-либо нештатной ситуации на шине (как описано в Таблице 38, выходные модули имеют свои точки работы в соответствии с конфигурацией модуля, когда ЦП переключается из режима работы в режим останова. В случае запуска приложения, когда ЦП переходит в режим останова, не перейдя в режим выполнения, выходные модули переводят свои точки в безопасный режим отказа, другими словами, поворачивают выключен (0 В пост. тока).

Что касается входных модулей, если один модуль удаляется из находящейся под напряжением стойки объединительной платы, состояние логической точки останется в последнем значении. В случае удаления разъема состояние логической точки будет переведено в безопасное состояние, что означает нулевой или высокий импеданс.

Ниже в Таблице 39 представлены состояния шины и рабочее состояние светодиода Nexto CPU DG.

Дополнительные сведения о состояниях светодиодов диагностики см. в разделе «Диагностика с помощью светодиодов».

Условие	Включено, с согласованностью запуска	Включено, с согласованностью запуска только для объявленных модулей	Включено, без согласованности при запуске	Выключено	Отключено, только для объявленных модулей	Отключено, без согласованности при запуске
Недекларированный модуль	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: остановлено
Недекларированный модуль (условие запуска)	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено
Отсутствует модуль	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: остановлено
Отсутствует модуль (условие запуска)	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено

Условие	Включено, с согласованностью запуска	Включено, с согласованностью запуска только для объявленных модулей	Включено, без согласованности при запуске	Выключено	Отключено, только для объявленных модулей	Отключено, без согласованности при запуске
Несовместимая конфигурация	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 2х Приложение: запущено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено
Несовместимая конфигурация (условие запуска)	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено или LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 2х Приложение: запущено
Дублированный адрес слота	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено
Нерабочий модуль	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено	LED DG: мигает 4х Приложение: остановлено

Таблица 39: Горячая замена и взаимосвязь условий

**Примечание:**

**Включено, без согласованности запуска:** Когда этот режим горячей замены включен, в обычных ситуациях, когда при запуске системы появляется несовместимый модуль, приложение переходит из состояния «Стоп» в состояние «Выполнить». Однако, если этот модуль обозначен как NX5000 или NX5001, а в этом месте находится другой модуль, приложение останется в состоянии «Останов».

**5.1.1.2. Сохраняемые и постоянные области памяти**

ЦП Nexto позволяет использовать символьные переменные и выходные переменные прямого представления в качестве сохраняемых или постоянных переменных.

Выходные переменные прямого представления, которые будут сохраняемыми или постоянными, должны быть объявлены в общих параметрах ЦП, как описано в разделе «Конфигурация ЦП». Этим выходным переменным прямого представления также можно присвоить символические имена с помощью директивы AT, плюс использование ключевого слова RETAIN или PERSISTENT в ее объявлении. Например, наличие %QB4096 и %QB20480 в сохраняемой и постоянной памяти соответственно.

```

PROGRAM UserPrg
VAR RETAIN

byRetentiveVariable_01 AT %QB4096 : BYTE;
END_VAR

VAR PERSISTENT

byPersistentVariable_01 AT %QB20480 : BYTE;
END_VAR

```

## 5. КОНФИГУРАЦИЯ

В случае, если символические переменные, объявленные с помощью директивы AT, не находятся в соответствующей сохраняемой и/или постоянной памяти, могут быть представлены ошибки во время генерации кода в MasterTool, информирующие о наличии несохраняемых или непостоянных переменных, определенных в сохраняемой или постоянной памяти. постоянные области памяти.

Что касается символических переменных, которые будут сохраняемыми или постоянными, только сохраняемые переменные

```
VAR_GLOBAL RETAIN
```

```
    wGlobalSymbolicRetentiveVariable_01 : WORD;  
END_VAR
```

```
    wLocalSymbolicRetentiveVariable_01 : WORD;  
END_VAR
```

могут быть локальными или глобальными, поскольку постоянные символические переменные всегда должны быть глобальными. Для объявления сохраняемых символьных переменных необходимо использовать ключевое слово RETAIN. Например, для локальных переменных или, для глобальных переменных, объявленных в списке глобальных переменных:

С другой стороны, постоянные символические переменные должны быть объявлены в объекте Persistent Variables, добавляемом в приложение. Эти переменные будут глобальными и будут объявлены внутри объекта следующим

```
VAR_GLOBAL PERSISTENT RETAIN
```

```
    wGlobalSymbolicPersistentVariable_01 : WORD;  
END_VAR
```

образом:

```
VAR_GLOBAL PERSISTENT RETAIN
```

```
    wGlobalSymbolicPersistentVariable_01 : WORD;  
END_VAR
```

```
VAR_GLOBAL RETAIN
```

### 5.1.1.3. Параметры проекта

Параметры проекта ЦП связаны с конфигурацией обновления ввода/вывода в задаче, в которой они используются в задачах проекта.

Конфигурация	Описание	По умолчанию	Варианты
Включить обновление ввода-вывода для каждой задачи	Обновляет ввод и вывод в задачах, где они используются	Не отмечено	- Отмечено - Не отмечено
Скопируйте проект с процессора на карту памяти	Карта памяти		
	Скопируйте проект из внутренней памяти ЦП на карту памяти	Неполноценный	- Включено: Конфигурация включена - Отключено: Конфигурация отключена
Пароль для копирования проекта с процессора на карту памяти	Пароль для копирования проекта из внутренней памяти ЦП на карту памяти	-	Пароль 6 цифр (0 до 999999)

## 5. КОНФИГУРАЦИЯ

Скопируйте проект с карты памяти на ЦП	Скопируйте проект с карты памяти во внутреннюю память ЦП	Неполноценный	- Включено: Конфигурация включена - Отключено: Конфигурация отключена
Пароль для копирования проекта с карты памяти на ЦП	Пароль для копирования проекта с карты памяти во внутреннюю память процессора	-	Пароль 6 цифр (0 до 999999)

Таблица 40: Параметры проекта процессора

### ВНИМАНИЕ

После настройки возможностей копирования проекта и создания загрузочного приложения необходимо найти файл «Application.crc», чтобы конфигурации, касающиеся карты памяти, вступили в силу. Поиск можно выполнить в Select the Application.crc с помощью кнопки Find File..., как показано на рисунке 135.

#### 5.1.1.4. Параметры CAN Port

Параметры порта CAN связаны с интерфейсом CAN.

Конфигурация	Описание	По умолчанию	Варианты
Включить внутреннее завершение	Включает терминацию внутреннего порта CAN	Не отмечено	- отмечено - не отмечено

Таблица 41: Параметры CAN Port

#### 5.1.2. Конфигурация внешнего события

Внешнее событие — это функция, доступная в ЦП, которая позволяет цифровому входу, установленному пользователем, при активации запускать выполнение определенной задачи с кодом, определяемым пользователем. Таким образом, возможно через этот вход при срабатывании прервать выполнение основного приложения и запустить заданное приложение в задаче ExternInterruptTask00, которая имеет более высокий приоритет, чем другие задачи приложения. Поскольку входы и выходы обновляются в контексте задачи MainTask, задача «Внешнее событие» не имеет обновленных входных и выходных данных во время ее вызова. При необходимости используйте функции обновления ввода/вывода.

Также важно отметить, что во избежание генерации нескольких событий за очень короткий промежуток времени была ограничена обработка этого типа событий каждые 10 мс, т.е. если два и более события происходят в течение 10 мс после первого события, второе и последующие события отбрасываются. Это ограничение наложено для предотвращения внешнего события, которое генерируется неконтролируемым образом, не блокирует ЦП, так как задача имеет более высокий приоритет по сравнению с другими.

Для регистрации внешнего события необходимо вставить модуль цифрового ввода и выполнить описанные ниже настройки в ЦП с помощью программного обеспечения MT8500.

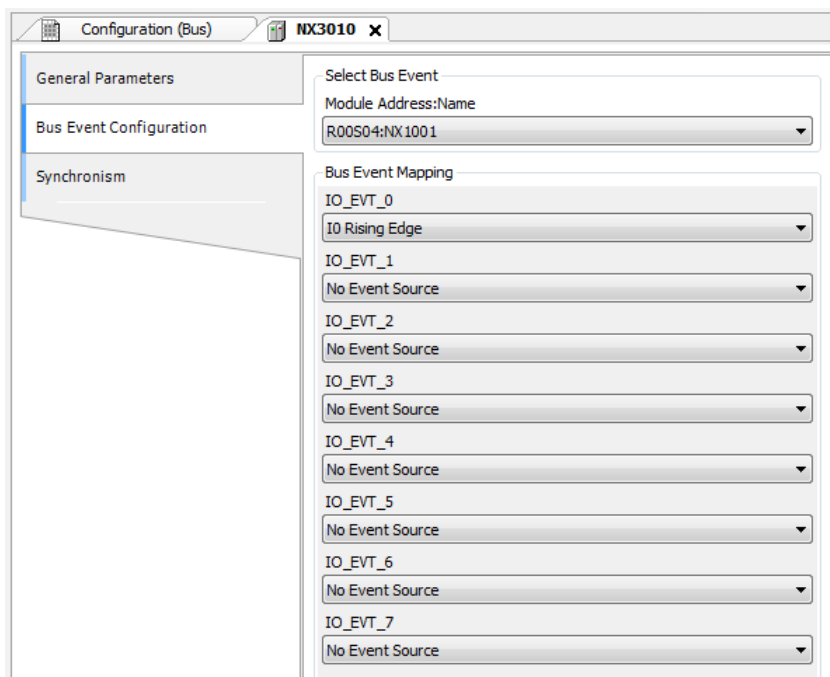


Рисунок 35: Экран конфигурации для внешнего события в ЦП

В закладке конфигурации Внешнее событие в настройках ЦП необходимо выбрать, какой модуль будет источником прерывания, в поле Адрес модуля: Имя. Затем необходимо выбрать, какой вход этого модуля будет отвечать за генерацию события (IO\_EVT\_0). В этом выборе могут быть выбраны параметры, описанные на рисунке ниже.

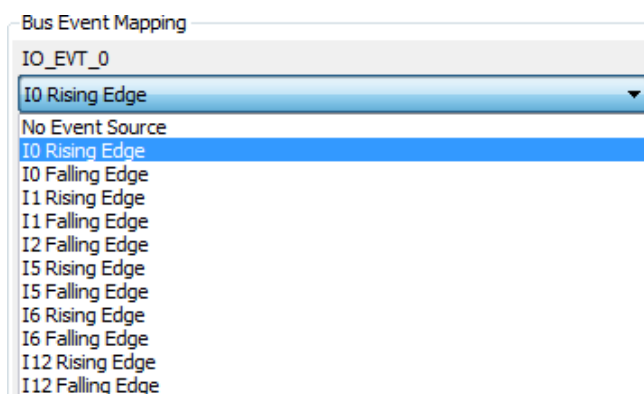


Рисунок 36: Опции внешнего источника событий модуля NX1001

Помимо настройки ЦП требуется конфигурация задачи, отвечающей за выполнение определяемых пользователем действий. В этом случае пользователь должен использовать профиль проекта, поддерживающий внешние события. Для получения дополнительной информации см. раздел «Профили проектов». На экране конфигурации ExternInterruptTask00 (Рисунок ниже) необходимо выбрать источник события в соответствующем поле. В этом случае следует выбрать IO\_EVT\_0, поскольку другие источники происхождения (от IO\_EVT\_1 до IO\_EVT\_7) недоступны. В последовательности следует проверить поле POU, если выбрано правильное POU, потому что оно будет использоваться пользователем для определения действий, которые должны выполняться при возникновении внешнего события.

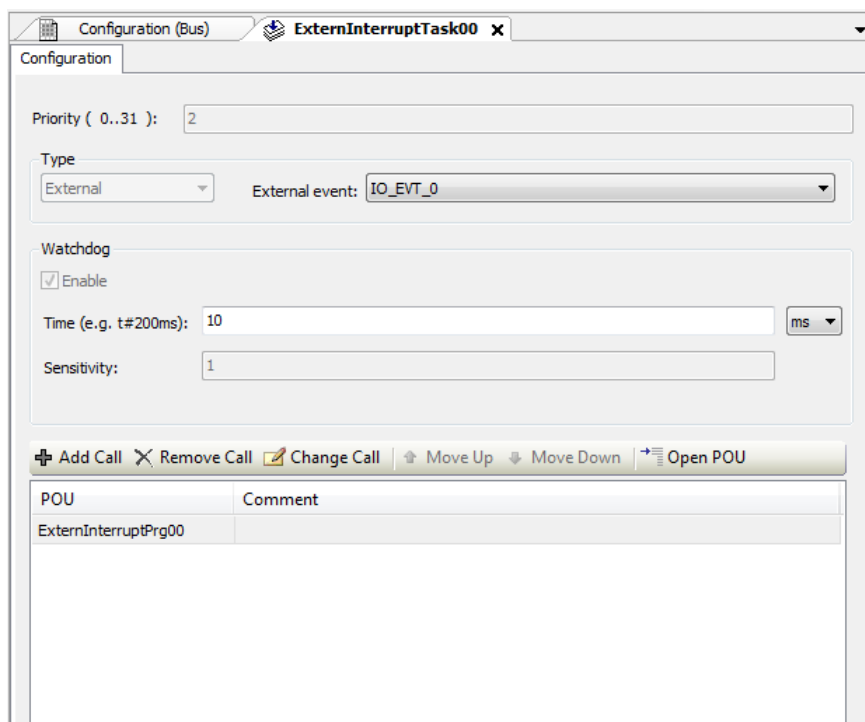


Рисунок 37: Экран конфигурации ExternInterruptTask00

### 5.1.3. Синхронизация времени

Для синхронизации времени процессоры серии Nexto используют SNTP (простой протокол сетевого времени) или синхронизацию через IEC 60870-5-104.

Чтобы использовать протоколы синхронизации времени, пользователь должен установить следующие параметры на вкладке «Синхронизм», доступ к которой осуществляется через ЦП, в дереве устройств:

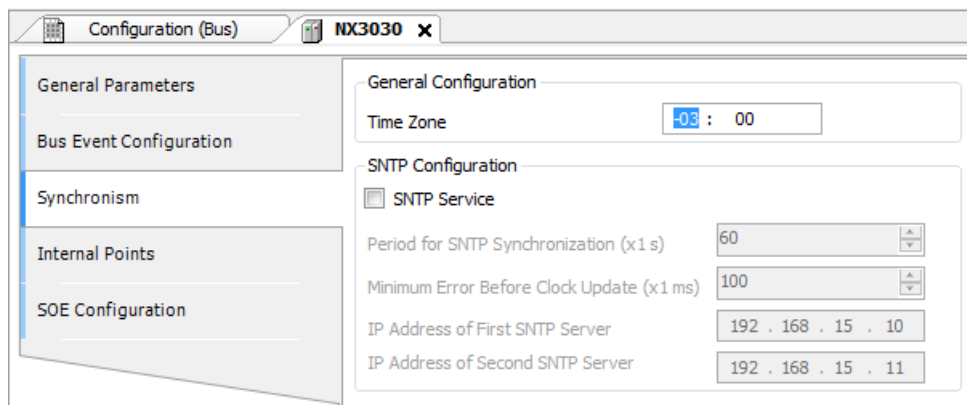


Рисунок 38: Конфигурация SNTP

Конфигурация	Описание	По умолчанию	Варианты
Часовой пояс (чч:мм)	Часовой пояс местоположения пользователя. Можно вставлять часы и минуты.	-3:00	12:59 до +13:59
SNTP-сервис	Включает службу SNTP.	Неполноценный	Неполноценный Включено
Период синхронизации SNTP (x1 с)	Временной интервал запросов на синхронизацию (секунды).	60	1 до 255
Минимальная ошибка перед обновлением часов (x1 мс)	Допустимое значение смещения между сервером и клиентом (миллисекунды).	100	1 до 65519
Адрес IP Первого SNTP	Адрес IP основного сервера SNTP.	192.168.15.10	1.0.0.1 до 223.255.255.254
Сервер	Адрес IP вторичного сервера SNTP.	192.168.15.11	1.0.0.1 до 223.255.255.254

Таблица 42: Конфигурация SNTP

**Примечания:**

**SNTP-сервер:** Можно определить предпочтительный адрес и еще один дополнительный адрес для доступа к серверу SNTP и, следовательно, для получения синхронности времени. Если оба поля пусты, служба SNTP останется отключенной.

**Часовой пояс:** конфигурация часового пояса используется для преобразования местного времени в UTC и наоборот. В то время как некоторые источники синхронизации используют местное время (протокол IEC 60870-5-104, функция SetDateAndTime), другие используют время UTC (SNTP). Время UTC обычно используется для отметки событий (DNP3, IEC 60870-5-104 и журнал устройств MasterTool), в то время как местное время используется другими функциями ЦП (функция GetDateAndTime, информация о дате и времени ОТД).

В проекте разрешено включать более одного источника синхронизации, однако устройство не поддерживает синхронизацию от более чем одного источника синхронизации во время работы. Поэтому неявно определен механизм приоритета. Синхронизация через SNTP более приоритетна, чем через протокол IEC 60870-5-104. Таким образом, когда оба источника включены и присутствует сервер SNTP, он будет отвечать за синхронизацию часов ЦП, и любая команда синхронизации из IEC 60870-5-104 будет отклонена.

**5.1.3.1. SNTP**

При включении ЦП будет вести себя как SNTP-клиент, то есть будет отправлять запросы на синхронизацию времени на SNT-P/NTP-сервер, который может находиться в локальной сети или в Интернете. SNTP-клиент работает с разрешением 1 мс, но с точностью до 100 мс. Точность синхронизации времени через SNTP зависит от конфигурации протокола (минимальная ошибка обновления часов) и характеристик сети Ethernet, в которой он находится, если и клиент, и сервер находятся в одной сети (локальная) или в разных сетях (удаленная). Обычно точность составляет порядка десятков миллисекунд.

ЦП отправляет Циклический запрос на синхронизацию в соответствии со временем, установленным в поле Период для синхронизации по SNTP. При первой попытке синхронизации, сразу после запуска службы, запрашивается первый сервер, указанный в IP-адресе первого сервера. В случае, если он не отвечает, запросы направляются на второй сервер, указанный во втором IP-адресе сервера, что обеспечивает резервирование серверов SNTP. В случае, если второй сервер также не отвечает, тот же самый процесс попытки синхронизации выполняется снова, но только по истечении периода синхронизации. Другими словами, в каждый период синхронизации ЦП пытается подключиться один раз к каждому серверу, он пытается подключиться ко второму серверу, если первый не отвечает. Время ожидания ответа от SNTP-сервера по умолчанию установлено в 5 с и не может быть изменено.

Если после синхронизации разница между текущим временем ЦП и временем, полученным сервером, превышает значение, установленное в параметре Минимальная ошибка перед обновлением часов, время ЦП обновляется. SNTP использует время в формате UTC (универсальное скоординированное время), поэтому параметр часового пояса должен быть установлен правильно, чтобы время, считанное SNTP, было правильно преобразовано в местное время.

Процесс выполнения клиента SNTP можно проиллюстрировать следующими шагами:

1. Попытка синхронизации через первый сервер. Если синхронизация прошла успешно, ЦП ожидает время новой синхронизации (Период синхронизации SNTP) и снова синхронизируется с этим сервером, используя его в качестве основного сервера. В случае сбоя (сервер не отвечает менее чем через 5 с) выполняется шаг 2.

2. Попытка синхронизации через второй сервер. Если синхронизация прошла успешно, ЦП ожидает время новой

## 5. КОНФИГУРАЦИЯ

синхронизации (Период синхронизации SNTP) и попытается синхронизироваться с этим сервером, используя первичный сервер. В случае сбоя (сервер не отвечает менее чем через 5 с) выжидается время, относящееся к периоду синхронизации, и снова выполняется шаг 1.

Поскольку время ожидания ответа сервера SNTP составляет 5 с, пользователь должен обратить внимание на значения периода синхронизации менее 10 с. В случае, если первичный сервер не отвечает, время синхронизации будет не менее 5 с (ожидание ответа первичного сервера и попытка синхронизации с вторичным сервером). В случае, если ни первичный сервер, ни вторичный не отвечают, время синхронизации будет минимум 10 с (ожидание ответа двух серверов и нового соединения с первой попытки сервера).

В зависимости от подсети сервера SNTP клиент будет использовать интерфейс Ethernet, который находится в соответствующей подсети, для отправки запросов на синхронизацию. Если в той же подсети, что и сервер, нет интерфейса conRisunokd, запрос может быть сделан любым интерфейсом, который может найти маршрут к серверу

### 5.1.3.2. Летнее время (DST)

Конфигурация перехода на летнее время должна выполняться косвенно через функцию SetTimeZone, которая изменяет

#### ВНИМАНИЕ

Служба SNTP зависит от пользовательского приложения только для своей конфигурации. Следовательно, эта служба будет выполняться, даже когда ЦП находится в режимах STOP или BREAKPOINT, если в ЦП есть приложение с ~~включенным и правильно настроенным клиентом SNTP~~

часовой пояс, применяемый к часам реального времени. В начале перехода на летнее время необходимо использовать функцию увеличения часового пояса на один час. В конце перехода на летнее время он используется для уменьшения его за один час.

Для получения дополнительной информации см. раздел Часы 24/7.

### 5.1.4. Внутренние точки

Точка связи — это хранилище в памяти ЦП в виде двух различных переменных. Один представляет значение точки (тип BOOL, BYTE, WORD и т. д.), а другой представляет ее качество (тип QUALITY). Внутренние точки — это точки, значение и качество которых вычисляются внутри пользовательского приложения, то есть они не имеют внешнего происхождения, как это бывает с точками, связанными с IED (коммуникационными драйверами типа Master/Client) или с

#### ВНИМАНИЕ

В отличие от того, что происходит с модулями ввода-вывода, объявленными на локальной шине, которые имеют свои собственные переменные качества, созданные MasterTool (качества ввода-вывода GVL) и автоматически обновляемые ЦП, модули ввода-вывода, объявленные на удаленных устройствах PROFIBUS, не имеют.

Пользователь несет ответственность за объявление переменных качества точки PROFIBUS, связь этих переменных качества с переменными значений на вкладке «Внутренние точки», а также за генерацию и обновление значения переменных качества из существующих Диагностика PROFIBUS: модули ввода-вывода PROFIBUS, головка PROFIBUS и мастер PROFIBUS.

локальными модулями ввода/вывода.

Функция этой вкладки конфигурации внутренних точек состоит в том, чтобы связать переменную, которая представляет значение точки, с той, которая представляет ее качество. Он должен использоваться для связи переменных значения и качества, созданных внутри программы ПЛК (как в GVL), которые обычно будут впоследствии отображаться на коммуникационный драйвер типа «Сервер» для связи с центром управления.

#### ВНИМАНИЕ

Если переменная-значение не имеет связанной переменной качества, она будет отображаться как постоянное хорошее качество по умолчанию (без существенных указаний), когда переменная-значение сообщается клиенту или центру управления.

Таким образом, эта вкладка не предназначена для создания или объявления внутренних точек. Для этого просто объявите переменные значения и/или качества в GVL и сопоставьте их с коммуникационным драйвером.

Конфигурация внутренних точек, показанная на рисунке ниже, соответствует параметрам, описанным в таблице ниже. В таблице Internal Points (внутренние точки) можно настроить до 5120 записей.

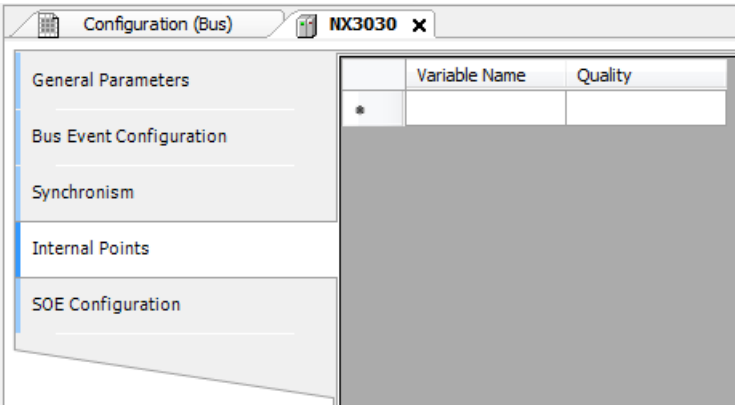
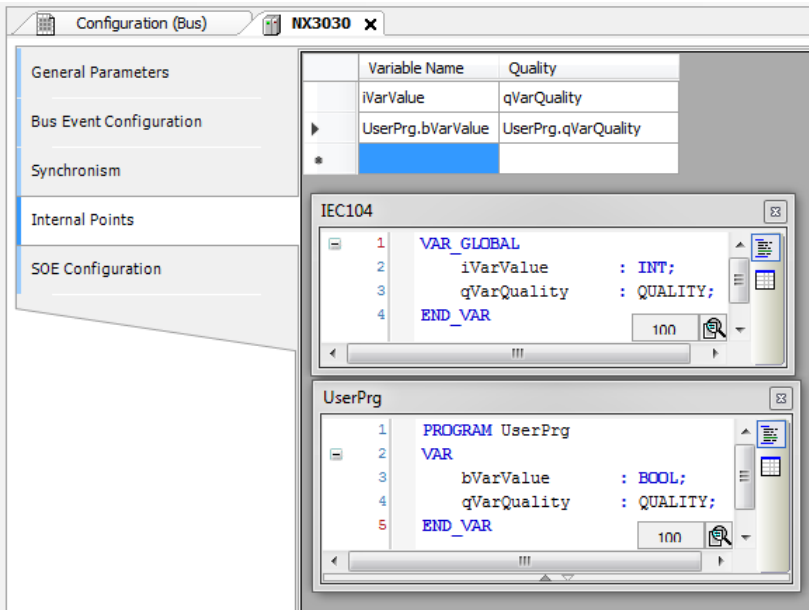


Рисунок 39: Экран конфигурации внутренних точек

Конфигурация	Описание	По умолчанию	Варианты
Имя переменной	Переменная символа, в которой хранится внутреннее значение точки	-	Accept variables of Тип BOOL, WORD, DWORD, LWORD, INT, DINT, LINT, UINT, UDINT, ULINT, REAL, LREAL or DBP. Переменная может быть простой, массивом или элементом массива и может быть частью структуры.
Качество	Переменная символа, в которой хранится внутреннее качество точки	-	Переменные типа QUALITY (LibRtuStandard), которые могут быть простыми, массивами или элементами массива и могут быть частью структуры.

Таблица 43: Конфигурация внутренних точек



На рисунке ниже показан пример конфигурации двух внутренних точек:

#### 5.1.4.1. Качественные конверсии

Качество внутренней точки — это информация об уровне доверия к значению, хранящемуся в этой точке. Качество может информировать, например, о том, что хранимое значение выходит за пределы допустимого диапазона или что оно допустимо, но имеет низкий уровень доверия.

Стандарты IEC 61850, DNP3 и IEC104 имеют собственные форматы представления информации о качестве точки. Серия Nexto, в свою очередь, имеет собственный формат качества (но очень похожий на IEC 61850), который называется Internal Quality. Этот формат определяется типом QUALITY (библиотека LibRtuStandard) и используется внутри для хранения качества, позволяя производить преобразование между протоколами без потери информации.

Когда выполняется сопоставление одной и той же точки связи между двумя водителями, преобразование качества выполняется автоматически в два этапа. Например: если точка связи отображается из драйвера клиента DNP3 в драйвер сервера IEC104, сначала качество будет преобразовано из формата DNP3 во внутренний формат (и сохранено внутри ЦП), после чего оно будет преобразовано из внутреннего формата к формату IEC104.

Следующие таблицы определяют преобразование собственных форматов протоколов во внутренний формат. Случай, когда необходимо проконсультироваться о преобразовании между протоколами, необходимо проанализировать в два этапа, просматривая каждую из таблиц во внутреннем формате и после их сопоставления

#### ВНИМАНИЕ

В случае внутренних точек, привязанных к коммуникационным драйверам, не рекомендуется изменять значение флагов качества, не имеющих соответствия в данном протоколе (т. е. флагов, не описанных в следующих таблицах). Это приведет к генерации событий, равных предыдущему (но с более поздней отметкой времени), и, таким образом, в зависимости от конфигурации, выбранной для режима передачи событий аналоговых входов, оно может перезаписать предыдущее событие, если это

##### 5.1.4.1.1. Внутреннее качество

Это структура КАЧЕСТВА. Таблица подробно показывает каждый из ее компонентов.

Бит	Наименование	Тип	Описание
0	FLAG_RESTART	BOOL	Флаг RESTART указывает на то, что данные поля не обновлялись с момента перезагрузки устройства.
1	FLAG_COMM_FAIL	BOOL	Указывает на сбой связи на пути между устройством источника данных и устройством отчетов.
2	FLAG_REMOTE_SUBSTITUTED	BOOL	Если TRUE, значения данных перезаписываются в удаленных устройствах связи.
3	FLAG_LOCAL_SUBSTITUTED	BOOL	Если TRUE, значение данных перезаписывается устройством, сгенерировавшим этот флаг. Такое поведение может возникать из-за работы в диагностике или временно из-за вмешательства человека.
4	FLAG_FILTER	BOOL	Флаг служит для сигнализации и предотвращения события перегрузки канала связи в виде колебаний (быстрых изменений) на цифровых входах.
5	FLAG_OVERFLOW	BOOL	Этот флаг должен указывать на проблему качества, т. е. на то, что значение атрибута, с которым связано качество, выходит за рамки представления.
6	FLAG_REFERENCE_ERROR	BOOL	Этот флаг должен указывать, что значение не может быть правильным из-за несоответствия калибровочного эталона.
7	FLAG_INCONSISTENT	BOOL	Этот флаг должен указывать, что функция оценки обнаружила несоответствие.

Бит	Наименование	Тип	Описание
-----	--------------	-----	----------

## 5. КОНФИГУРАЦИЯ

8	FLAG_OUT_OF_RANGE	BOOL	Этот флаг должен указывать на проблему качества, заключающуюся в том, что атрибут, с которым было связано качество, выходит за пределы предварительно определенных значений.
9	FLAG_INACCURATE	BOOL	Этот флаг должен указывать на то, что значение не соответствует заявленной точности источника.
10	FLAG_OLD_DATA	BOOL	Значение кажется устаревшим. В случае, если обновление не происходит в течение определенного периода времени.
11	FLAG_FAILURE	BOOL	Этот флаг должен указывать на то, что функция наблюдения обнаружила внутреннюю или внешнюю ошибку.
12	FLAG_OPERATOR_BLOCKED	BOOL	Обновление заблокировано оператором.
13	FLAG_TEST	BOOL	Это должен быть дополнительный идентификатор, который можно использовать для классификации значения как тестового значения, которое не будет использоваться в рабочих целях.
14-15	RESERVED	-	Зарезервировано
16-17	VALIDITY	QUALITY_VALIDITY	0 – Хорошо (доверительное значение, означает отсутствие нештатных условий) 1 – Недействительно (Значение не соответствует значению ИЭУ) 2 – Зарезервировано (Зарезервировано) 3 – сомнительно (текущее значение может отличаться от ИЭУ)

Таблица 44: Структура качества

### 5.1.4.1.2. Внутреннее качество MODBUS

Поскольку стандарт MODBUS не определяет типы качества для каждой точки, но для помощи в использовании диагностики связи каждой точки, MasterTool позволяет отображать переменные качества через внутреннюю собственную структуру для каждой точки MODBUS. Таблица ниже описывает типы качества, которые может принимать каждая точка MODBUS.

Итоговое качество	Итоговая ДЕЙСТВИТЕЛЬНОСТЬ	Описание
FLAG_RESTART	VALIDITY_INVALID	Начальное значение. Точка никогда не обновлялась.
-	VALIDITY_GOOD	Связь ОК. Пункт обновлен.
FLAG_COMM_FAIL AND FLAG_RESTART	VALIDITY_INVALID	Ошибка связи. Точка никогда не обновлялась.
FLAG_COMM_FAIL AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Произошла ошибка, но точка была обновлена и теперь имеет старое значение
FLAG_FAILURE AND FLAG_RESTART	VALIDITY_INVALID	Он получил ответ об исключении, и точка сохранена как начальное значение.
Итоговое качество	Итоговая ДЕЙСТВИТЕЛЬНОСТЬ	Описание
FLAG_FAILURE AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Он получил ответ об исключении, но точка имеет допустимое старое значение.
FLAG_RESTART AND FLAG_OLD_DATA	VALIDITY_QUESTIONABLE	Устройство остановлено. Точка имеет старое значение.

Таблица 45: Качество MODBUS

## 5.1.4.1.3. Модули качество ввода/вывода локальной шины

Чтобы помочь в использовании диагностики каждой точки ввода/вывода, MasterTool автоматически создает структуру качества для каждого модуля локальной шины, используемого в проекте ПЛК, посредством собственной внутренней структуры, доступной через структуру QUALITY, доступную в GVL IOQualities.

Таблица ниже описывает типы качества для каждой точки входа и выхода. Для получения дополнительной информации см. GVL IOQualities.

Диагностики	Итоговое качество	Итоговая ДЕЙСТВИТЕЛЬНОСТЬ	Описание
Все равно	FLAG_RESTART	VALIDITY_INVALID	Качество имеет это значение до того, как оно будет прочитано или записано в первый раз.
Не указано	-	VALIDITY_GOOD	Связь ОК. Пункт обновлен.
Не указано	FLAG_OLD_DATA AND FLAG_FAILURE	VALIDITY_QUESTIONABLE	Нерабочий модуль. Однако данные были прочитаны или записаны хотя бы один раз.
bOverRange OR bUnder-Range	FLAG_OUT_OF_RANGE	VALIDITY_INVALID	Значение выше или ниже допустимого диапазона входа модуля.
bInputNotEnable OR bOutputNotEnable	FLAG_OPERATOR_BLOCKED	VALIDITY_INVALID	Ввод/вывод не включен.
bOpenLoop	FLAG_FAILURE	VALIDITY_INVALID	Разомкнутый контур во входном модуле.
bFatalError	FLAG_FAILURE	VALIDITY_INVALID	Фатальный сбой оборудования.
bNoВнешнийSupply	FLAG_FAILURE	VALIDITY_INVALID	Внешний источник питания находится ниже рабочего минимального предела.
bShortCircuit OR bOutput-ShortCircuit	FLAG_FAILURE	VALIDITY_INVALID	Короткое замыкание на выходе.
bCalibrationError	FLAG_INACCURATE	VALIDITY_INVALID	Ошибка калибровки.
bColdJunctionSensorError	FLAG_INACCURATE	VALIDITY_INVALID	Ошибка датчика холодного спая.

Таблица 46: Качество модулей ввода/вывода

## 5.1.4.1.4. Качество модулей ввода/вывода PROFIBUS

В отличие от локальной шины, MasterTool не создает структуры качества модулей PROFIBUS автоматически, и ПЛК не обновляет такие структуры. Поэтому ответственность за создание и Циклическое обновление модулей PROFIBUS лежит на пользователе.

Чтобы помочь в разработке таких приложений, существуют следующие практические примеры на языке ST для основных модулей PROFIBUS (DI, DO, AI, AO), основанных на ведомых устройствах PROFIBUS Nexto Serie (NX5110). Пользователь должен чувствовать поощрение к любой необходимой адаптации и изменению, чтобы соответствовать его приложению.

Разработка подпрограммы обновления точек качества модулей ввода/вывода PROFIBUS должна начинаться с

**ВНИМАНИЕ**

Для правильного функционирования подпрограмм, представленных последовательно, необходимо включить Status в Diagnose в ведомых устройствах PROFIBUS

объявления и инициализации переменных качества, с GVL:

```
VAR_GLOBAL
```

```
QUALITY_PB_NX1005_I: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
```

```

QUALITY_PB_NX1005_O: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
                                FLAGS:= (FLAG_RESTART:= TRUE));
QUALITY_PB_NX6000: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
                                FLAGS:= (FLAG_RESTART:= TRUE));
QUALITY_PB_NX6100: LibDataTypes.QUALITY:= (VALIDITY:= VALIDITY_INVALID,
                                FLAGS:= (FLAG_RESTART:= TRUE));

```

## 5.1.4.1.5. Качество цифровых входов PROFIBUS

```

// Обновление качества цифрового входа PROFIBUS, модуль NX1005
// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99) ... IF
DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN
// Ожидает, что ведомое устройство PROFIBUS станет способным к обмену данными и диагностикой
// (Необходимо подождать, чтобы не допустить генерации некорректного качества) IF
DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
  DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND DG_NX5110.tPbusHeadA.wIdentNumber >
  0 THEN QUALITY_PB_NX1005_I.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Если на шине присутствует модуль (slot = 2) и
// если нет проблем с конфигурацией модулей (общих) и
// если в этом модуле (конкретном) нет проблем с конфигурацией и
// если модуль не идентифицировал фатальную ошибку ...
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 2)) = 0 AND
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND DG_NX1005_24_B
  пост.тока_8_DO_Trans_8_DI.tGeneral.bConfigMismatch = FALSE AND DG_NX1005_24_B
  пост.тока_8_DO_Trans_8_DI.tGeneral.bFatalError = FALSE THEN QUALITY_PB_NX1005_I.VALIDITY:=
  VALIDITY_GOOD; QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART:= FALSE;
  QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= FALSE;
  QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= FALSE;
ELSE
  QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_INVALID;
  QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= TRUE;
  // Если точка когда-либо обновлялась один раз ...
  IF NOT QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART THEN
    QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= TRUE;
  END_IF
END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS ... ELSE
QUALITY_PB_NX1005_I.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX1005_I.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX1005_I.FLAGS.FLAG_FAILURE:= FALSE;
// Если точка когда-либо обновлялась один раз ...
IF NOT QUALITY_PB_NX1005_I.FLAGS.FLAG_RESTART THEN
  QUALITY_PB_NX1005_I.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
END_IF

```

## 5.1.4.1.6. Качество цифрового вывода PROFIBUS

```

// Обновление качества цифрового выхода PROFIBUS, модуль NX1005

// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99)... IF
DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN
// Ожидает, что ведомое устройство PROFIBUS станет способным к обмену данными и диагностикой
// (Необходимо подождать, чтобы не допустить генерации некорректного качества) IF
DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
  DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND DG_NX5110.tPbusHeadA.wIdentNumber > 0
  THEN
QUALITY_PB_NX1005_O.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Если на шине присутствует модуль (slot = 2) и
// если нет проблем с конфигурацией модулей (общих) и
// если в этом модуле (конкретном) нет проблем с конфигурацией и
// если модуль не идентифицирует фатальную ошибку и
// если нет индикации короткого замыкания выходов и
// если нет индикации отсутствия внешнего источника питания ...
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 2)) = 0 AND
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND DG_NX1005_24_B
  пост.тока_8_DO_Trans_8_DI.tGeneral.bConfigMismatch = FALSE AND DG_NX1005_24_B
  пост.тока_8_DO_Trans_8_DI.tGeneral.bFatalError = FALSE AND DG_NX1005_24_B
  пост.тока_8_DO_Trans_8_DI.tDetailed.bOutputShortCircuit = FALSE AND DG_NX1005_24_B
  пост.тока_8_DO_Trans_8_DI.tDetailed.bNoВнешнийSupply = FALSE THEN
  QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_GOOD;
  QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART:= FALSE;
  QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= FALSE;
  QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= FALSE;
ELSE
  QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_INVALID;
  QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= TRUE;
  // Если точка когда-либо обновлялась один раз...
  IF NOT QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART THEN
    QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= TRUE;
  END_IF
END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS ... ELSE
QUALITY_PB_NX1005_O.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX1005_O.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX1005_O.FLAGS.FLAG_FAILURE:= FALSE;
// Если точка когда-либо обновлялась один раз ...
IF NOT QUALITY_PB_NX1005_O.FLAGS.FLAG_RESTART THEN
  QUALITY_PB_NX1005_O.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
END_IF

```

## 5.1.4.1.7. Качество аналоговых входов PROFIBUS

// Обновление качества аналогового входа PROFIBUS, модуль NX6000

```
// In communication success case with PROFIBUS slave (address = 99) ... IF
DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN
// Ожидает, что ведомое устройство PROFIBUS станет готовым к обмену данными и
диагностикой
// (Необходимо подождать, чтобы избежать генерации недопустимого качества) IF
DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
  DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND DG_NX5110.tPbusHeadA.wIdentNumber > 0
  THEN
QUALITY_PB_NX6000.FLAGS.FLAG_COMM_FAIL:= FALSE;
// Если на шине присутствует модуль (slot = 3) и
// если нет проблем с конфигурацией модулей (общих) и
// если в этом модуле (конкретном) нет проблем с конфигурацией и
// если модуль не идентифицирует фатальную ошибку и
// если нет индикации ошибки калибровки и
// если нет индикации ошибки превышения/недостатка диапазона и
// если нет индикации ошибки ввода в разомкнутом цикле ...
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 3)) = 0 AND
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bConfigMismatch = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bFatalError = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bCalibrationError = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOverRange = FALSE
  AND
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bUnderRange = FALSE AND
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop = FALSE THEN
  QUALITY_PB_NX6000.VALIDITY:= VALIDITY_GOOD; QUALITY_PB_NX6000.FLAGS.FLAG_RESTART:=
  FALSE; QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= FALSE;
  QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= FALSE;
ELSE
  // Условие для включения индикации неточности
  // (сначала проверьте, т.к. должна преобладать недопустимая валидность)
  IF DG_NX6000_8_AI_Voltage_Current.tGeneral.bCalibrationError = TRUE THEN
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= TRUE;
  ELSE
    QUALITY_PB_NX6000.FLAGS.FLAG_INACCURATE:= FALSE; END_IF
  // Условие для включения индикации выхода за пределы допустимого диапазона
  // (сначала проверьте, т.к. должна преобладать недопустимая валидность)
  IF DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOverRange =
  TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bUnderRange = TRUE THEN
    QUALITY_PB_NX6000.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= TRUE;
  ELSE
    QUALITY_PB_NX6000.FLAGS.FLAG_OUT_OF_RANGE:= FALSE;
```

```

END_IF
// Условие включения общей индикации отказа (приоритет)
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 3)) > 0 OR
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bConfigMismatch = TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tGeneral.bFatalError = TRUE OR
  DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop = TRUE
THEN
  QUALITY_PB_NX6000.VALIDITY:= VALIDITY_INVALID;
  QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= TRUE;
  // Если точка когда-либо обновлялась один раз... IF NOT
  QUALITY_PB_NX6000.FLAGS.FLAG_RESTART AND
  NOT DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop
  THEN
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= TRUE; END_IF
  ELSE
    QUALITY_PB_NX6000.FLAGS.FLAG_RESTART:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
    QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= FALSE;
  END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS... ELSE
QUALITY_PB_NX6000.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX6000.FLAGS.FLAG_COMM_FAIL:= TRUE;
QUALITY_PB_NX6000.FLAGS.FLAG_FAILURE:= FALSE;
// Если точка когда-либо обновлялась один раз... IF NOT
QUALITY_PB_NX6000.FLAGS.FLAG_RESTART AND
  NOT DG_NX6000_8_AI_Voltage_Current.tDetailed.tAnalogInput_00.bOpenLoop THEN
  QUALITY_PB_NX6000.FLAGS.FLAG_OLD_DATA:= TRUE;
END_IF
END_IF

```

```

// Обновление качества аналогового выхода PROFIBUS, модуль NX6100

// В случае успешной связи с ведомым устройством PROFIBUS (адрес = 99)...IF
DG_NX5001.tMstStatus.abvSlv_State.bSlave_99 = TRUE THEN

// Ожидает, что ведомое устройство PROFIBUS станет готовым к обмену данными и
диагностикой

// (Необходимо подождать, чтобы избежать генерации недопустимого качества)

IF DG_NX5110.tPbusHeadA.tStatus1.bStation_Non_Existent = FALSE AND
  DG_NX5110.tPbusHeadA.tStatus1.bStation_Not_Ready = FALSE AND
  DG_NX5110.tPbusHeadA.wIdentNumber > 0 THEN

QUALITY_PB_NX6100.FLAGS.FLAG_COMM_FAIL:= FALSE;

// Если на шине присутствует модуль (слот = 4) и // если нет проблем с конфигурацией модулей (общие) и
// если в этом модуле (конкретном) нет проблем с конфигурацией и// если нет идентификации фатальной
ошибки модулем и

```

#### 5.1.4.1.8. Качество аналоговых выходов PROFIBUS

```

// если нет индикации отсутствия внешнего источника питания и
// если нет индикации ошибки выхода в разомкнутом цикле и
// если нет индикации короткого замыкания выходов ...
IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 4)) = 0 AND
  DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = FALSE AND
  DG_NX6100_4_AO_Voltage_Current.tGeneral.bConfigMismatch = FALSE AND
  DG_NX6100_4_AO_Voltage_Current.tGeneral.bFatalError = FALSE AND
  DG_NX6100_4_AO_Voltage_Current.tGeneral.bCalibrationError = FALSE AND
  DG_NX6100_4_AO_Voltage_Current.tGeneral.bNoВнешнийSupply = FALSE AND
  DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop = FALSE
  AND
  DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bShortCircuit = FALSE THEN
  QUALITY_PB_NX6100.VALIDITY:= VALIDITY_GOOD; QUALITY_PB_NX6100.FLAGS.FLAG_RESTART:=
  FALSE; QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
  QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= FALSE;
  QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= FALSE;
ELSE
  // Условие для включения индикации неточности
  // (сначала проверьте, т.к. должна преобладать недопустимая валидность)
  IF DG_NX6100_4_AO_Voltage_Current.tGeneral.bCalibrationError = TRUE THEN
    QUALITY_PB_NX6100.VALIDITY:= VALIDITY_QUESTIONABLE;
    QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= TRUE;
  ELSE
    QUALITY_PB_NX6100.FLAGS.FLAG_INACCURATE:= FALSE;
  END_IF
  // Условие для включения индикации общего отказа (приоритет)
  IF (DG_NX5110.tPbusHeadA.dwModuleNotPresent AND SHL(1, 4)) > 0 OR
    DG_NX5110.tPbusHeadA.tSummarized.bConfigMismatch = TRUE OR
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bConfigMismatch = TRUE OR
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bFatalError = TRUE OR
    DG_NX6100_4_AO_Voltage_Current.tGeneral.bNoВнешнийSupply = TRUE OR
    DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop = TRUE
    OR
    DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bShortCircuit = TRUE THEN
    QUALITY_PB_NX6100.VALIDITY:= VALIDITY_INVALID;
    QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= TRUE;
    // Если точка когда-либо обновлялась один раз...
    IF NOT QUALITY_PB_NX6100.FLAGS.FLAG_RESTART AND NOT
      DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop
    THEN
      QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= TRUE;
    END_IF
  ELSE
    QUALITY_PB_NX6100.FLAGS.FLAG_RESTART:= FALSE;
    QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;
    QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= FALSE;
  END_IF
END_IF
END_IF
// При сбое связи PROFIBUS с ведомым устройством PROFIBUS... ELSE
QUALITY_PB_NX6100.VALIDITY:= VALIDITY_INVALID;
QUALITY_PB_NX6100.FLAGS.FLAG_COMM_FAIL:= TRUE;

```

```

QUALITY_PB_NX6100.FLAGS.FLAG_FAILURE:= FALSE;

// Если точка когда-либо обновлялась один раз...IF
NOT QUALITY_PB_NX6100.FLAGS.FLAG_RESTART AND

    NOT DG_NX6100_4_AO_Voltage_Current.tDetailed.tAnalogOutput_00.bOpenLoop THEN
QUALITY_PB_NX6100.FLAGS.FLAG_OLD_DATA:= TRUE;

END_IF
END_IF

```

## 5.2. Конфигурация последовательных интерфейсов

### 5.2.1. COM 1

Коммуникационный интерфейс COM 1 состоит из клемм D+ и D- для полудуплексного стандарта RS-485, обеспечивающего двухточечную или сетевую связь по открытым протоколам MODBUS RTU Slave или MODBUS RTU Master.

При использовании протокола MODBUS Master/Slave некоторые из этих параметров (например, последовательный режим, биты данных, порог приема) автоматически настраиваются MasterTool для правильной работы этого протокола.

Ниже следуют параметры, которые необходимо настроить для правильной работы приложения.

Конфигурация	Описание	По умолчанию	Варианты
Серийный тип	Конфигурация последовательного канала.	RS-485	RS-485
Скорость передачи данных	Конфигурация скорости порта последовательной связи.	115200	2400, 4800, 9600, 19200, 38400, 57600, 115200 bps
Паритет	Конфигурация четности последовательного порта.	Не указано	Нечетное число Четное число не указано
Биты данных	Устанавливает количество битов данных	8	7 и 8
Стоповые биты	в каждом символе последовательной связи.	1	1 и 2
Последовательный режим	Устанавливает стоповые биты последовательного порта.	Нормальный режим	- Расширенный режим: расширенный режим работы, который предоставляет информацию о полученном кадре данных (см. примечание к разделу COM 1). - Нормальный режим: нормальный режим работы последовательной связи.

Таблица 47: RS-485 Стандартные последовательные конфигурации

#### Примечания:

**Расширенный режим:** Этот режим работы последовательной связи предоставляет информацию о принятом кадре данных. Доступная информация следующая:

Один байт для полученных данных (RX\_CHAR : BYTE): Сохраните семь или восемь бит полученных данных, в зависимости от конфигурации последовательной связи.

Один байт для ошибок сигнала (RX\_ERROR : BYTE): Он имеет формат, описанный ниже:

- Бит 0: 0 — допустимы символы в битах с 0 по 7. 1 - символ в битах с 0 по 7 недействителен (или не может быть допустим) из-за проблем, указанных в битах с 10 по 15.

## 5. КОНФИГУРАЦИЯ

- Бит 2: не используется.
- Бит 3: ошибка прерывания UART. Последовательный ввод оставался в логическом 0 (пробел) в течение времени, превышающего символ (стартовый бит + биты данных + бит четности + стоповые биты).
- Бит 4: ошибка кадра UART. Логический 0 (пробел) был прочитан, когда ожидался первый стоповый бит, и он должен быть логической 1 (метка).
- Бит 5: ошибка четности UART. Считанный бит четности неверен в соответствии с вычисленным значением.
- Бит 6: ошибка переполнения UART. Данные были потеряны во время чтения FIFO UART. Новые персонажи были получены до того, как более поздние были удалены. Эта ошибка будет указана только в первом символе, прочитанном после индикации ошибки переполнения. Это означает, что некоторые старые данные были потеряны.
- Бит 7: Ошибка переполнения линии RX. Этот символ был записан, когда линия RX была завершена, перезаписывая непрочитанные символы.

Два байта для сигнала метки времени (RX\_TIMESTAMP : DWORD): указывает время молчания в интервале от 0 до 4 294 967 295 с использованием 1 мкс в качестве базы. Он насыщается в 4.294.967.295 мкс, если время молчания выше 4.294.967.295 единиц. RX\_TIMESTAMP символа измеряет время по ссылке, которая может быть любой из трех опций ниже:

- В большинстве случаев конец более позднего символа.
- Конфигурация последовательного порта.
- Окончание последовательной связи с использованием функционального блока SERIAL\_TX, другими словами, когда последний символ отправляется в линию.

Помимо измерения тишины между символами, RX\_TIMESTAMP также важен, поскольку он измеряет время паузы последнего символа в строке RX. Измерение тишины важно для правильной реализации протокола, например, MODBUS RTU. Этот протокол определяет межкадровый интервал длиной более 3,5 символов и межбайтовый интервал длиной менее 1,5 символа.

Биты данных: Конфигурация битов данных последовательных интерфейсов ограничивает поля Stop Bits и Communication Parity. Следовательно, количество стоповых битов и метод контроля четности будут варьироваться в зависимости от количества битов данных.

Биты данных	Стоповые биты	Паритет
7	1, 2	NO PARITY, ODD, EVEN
8	1, 2	NO PARITY, ODD, EVEN

Таблица 48: Конкретные конфигурации

### 5.2.1.1. Расширенные конфигурации

Расширенные конфигурации связаны с управлением последовательной связью, другими словами, когда необходимо использование более точного управления передачей и приемом данных.

Конфигурация	Описание	По умолчанию	Варианты
Порог приема UART	Количество байтов, которое должно быть получено, чтобы сгенерировать новое прерывание UART. Низкие значения делают TIMESTAMP более точным при использовании РАСШИРЕННОГО РЕЖИМА и минимизируют ошибки переполнения. Однако слишком низкие значения могут вызвать несколько прерываний, задерживающих работу ЦП.	8	1, 4, 8 и 14
Терминатор RS-485	Включите завершение RS-485 в COM1.	Включено	Включено Выключено

Таблица 49: Стандартные последовательные расширенные конфигурации RS-485

### 5.3. Конфигурация интерфейса ЛСК (Локальная сеть контроллера)

#### 5.3.1. ЛСК

Интерфейс ЛСК обеспечивает двухточечную или сетевую связь с другими устройствами, имеющими этот интерфейс, с использованием прикладного протокола CANopen.

Параметры интерфейса ЛСК, которые необходимо настроить для корректной работы приложения, описаны ниже:

Конфигурация	Описание	По умолчанию	Варианты
Сеть	Идентификационный номер интерфейса CAN	0	0 (фиксированный)
Скорость передачи данных	Скорость передачи по шине CAN (кбит/с). Другие устройства должны использовать ту же скорость передачи данных.	250	10, 20, 50, 100, 125, 250, 500, 1000

Таблица 50: Конфигурация ЛСК

Параметры, относящиеся к протоколу CANopen, описаны в разделе Конфигурация протоколов.

### 5.4. Доступ к веб-странице управления

Разработан для выполнения Конфигурации и диагностики доступа к некоторым функциям. Доступ к веб-странице защищен пользователем и паролем, при этом admin является значением По умолчанию для обоих полей.

На вкладке «Управление» есть другие ресурсы, такие как «Система», «Сеть», «USB-устройство», «Брандмауэр» и «OpenVPN». Ресурсы, доступные на этой вкладке, различаются в зависимости от функций, доступных для используемого контроллера, и доступ к ним возможен только после входа пользователя в систему, как показано на рисунке ниже.

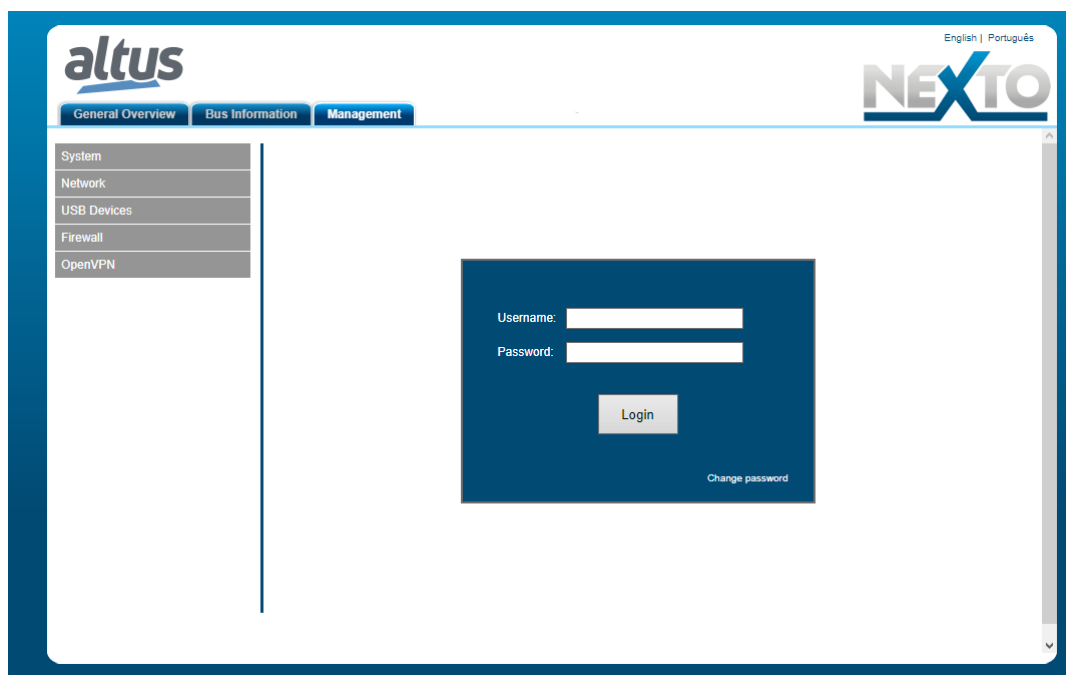


Рисунок 41: Доступ к веб-управлению

#### 5.4.1. Страница системы

Вкладка «Система» предназначена для внутреннего использования в Altus и ограничена пользователем. Здесь выполняется обновление микропрограммы ЦП. Для случаев, когда обновление выполняется удаленно (например, через радио- или спутниковое соединение), минимальная скорость этого канала должна быть 128 кбит/с.

### 5.4.2. Страница сети

Страница «Сеть» (Рисунок ниже), предназначенная для облегчения использования контроллера, позволяет изменять сетевые адреса и запускать Network Sniffer.

The screenshot shows the Altus NEXTO web interface. The sidebar on the left contains links for System, Network, USB Devices, Firewall, and OpenVPN. The main content area is titled 'Network' and contains two sections: 'Configuration' and 'Sniffer'. The 'Configuration' section has a 'Mode' dropdown menu set to 'Defined by web page'. Below it is a table with three columns: 'NET1', 'NET2', and 'NET3'. The 'Sniffer' section has two input fields: 'Number of Packages' with the value '100' and 'Idle Timeout (seconds)' with the value '10'. Below these is an 'Interfaces' section with three checkboxes: 'NET1' (checked), 'NET2' (unchecked), and 'NET3' (unchecked). At the bottom right of the 'Interfaces' section are 'Download' and 'Execute' buttons.

Рисунок 42: Веб страница сети

#### 5.4.1.1. Конфигурация страницы сети

##### 5.4.1.1.1. Определяется приложением

Поле Mode определяет, какую Конфигурацию контроллер должен загрузить для своих интерфейсов. Это поле может быть истолковано как

Определяется веб-страницей или определяется приложением.

Если установлено значение «Определено приложением», интерфейс «Таблица» отключен, что не позволяет вносить изменения, как показано на рисунке ниже. В этом режиме настройки, применяемые к контроллеру, определяются приложением.

ВНиже приведено изображение с выбранным режимом «Определено приложением», показывающее отключенный интерфейс «Таблица».

#### ВНИМАНИЕ

Таблица конфигурации сети отображается только тогда, когда на контроллере нет приложения или контроллер не работает. Невозможно изменить сетевые настройки во время работы приложения на контроллере.

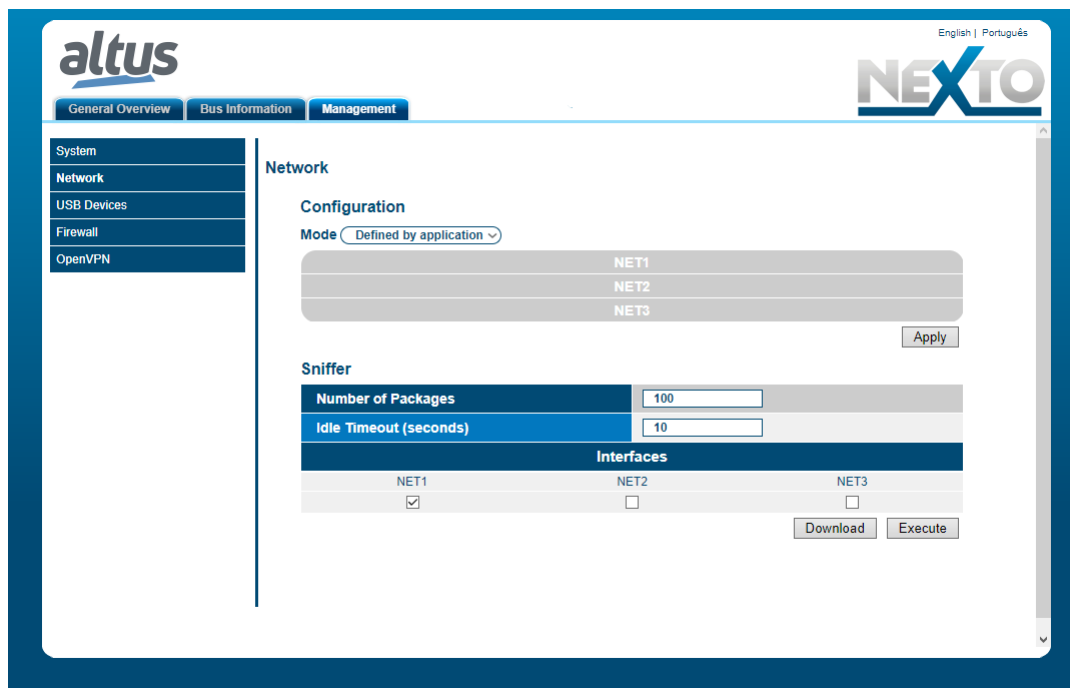
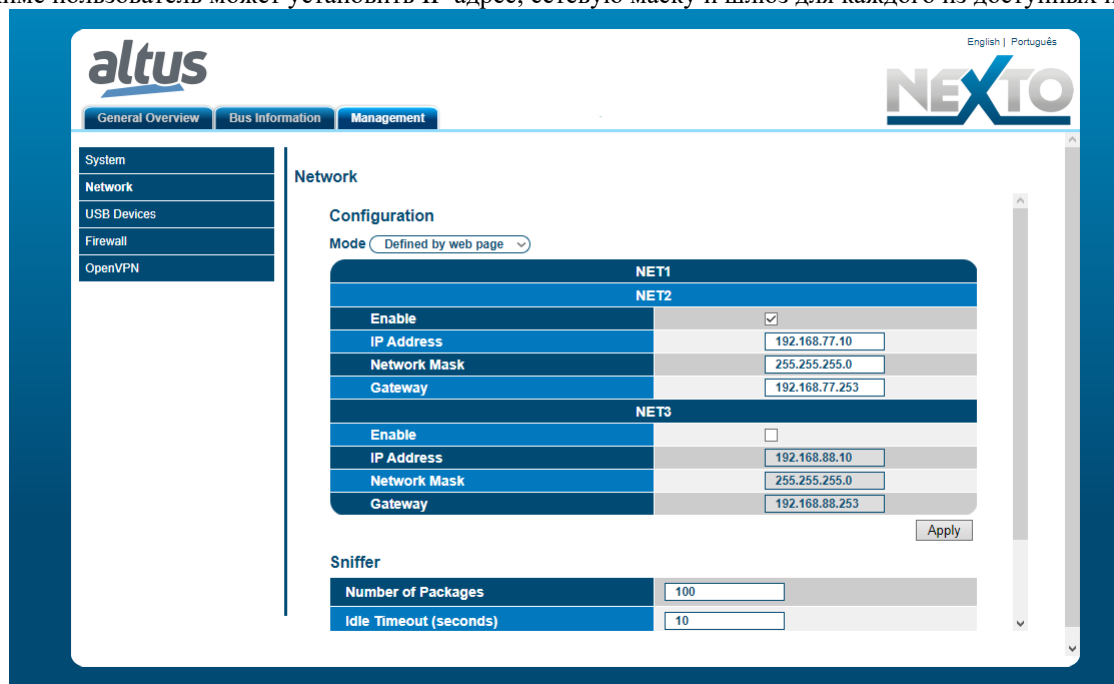


Рисунок 43: Таблица интерфейсов - Режим приложения

## 5.4.1.1.2. Определяется веб-страницей

Для режима «Определено веб-страницей» интерфейс «Таблица» остается включенным, как показано на рисунке ниже. В этом режиме пользователь может установить IP-адрес, сетевую маску и шлюз для каждого из доступных интерфейсов



Ethernet, а также включить или отключить сети 2 и 3.

Рисунок 44: Таблица интерфейсов - Веб-режим

Флажок Enable позволяет пользователю включать и отключать интерфейсы Ethernet. Это доступно только для выбора или снятия флажка, когда Конфигурация определяется веб-страницей.

Когда этот флажок снят, это означает, что интерфейс NET отключен, т. е. он не будет получать Конфигурацию и будет деактивирован, как NET 3 на рисунке выше. Когда интерфейс включен (с установленным флажком), согласно NET 2 на рисунке выше, настройки доступны для редактирования.

Чтобы применить настройки к контроллеру, просто нажмите кнопку «Применить». Этот процесс проверяет, были ли допущены какие-либо ошибки в Конфигурации, и, если они есть, выводит на экран браузера сообщение, указывающее на ошибку. Если настройки верны, после нажатия кнопки «Применить» в браузере появится окно подтверждения для применения новых настроек. При нажатии ОК настройки отправляются на контроллер и

### ВНИМАНИЕ

При изменении сети в контроллере интерфейсы будут перезапущены, что может привести к потере связи. Особенно при изменении значения IP-адреса.

### ВНИМАНИЕ

Режим «Определено веб-страницей» настраивает интерфейсы для работы в простом режиме.

применяются.

При применении настроек в режиме Определено приложением контроллер примет Конфигурацию, которая была определена загруженным приложением. Если приложения нет, текущая Конфигурация будет сохранена с изменением только режима Конфигурация.

При использовании режима Определено веб-страницей будут загружены адреса, указанные на веб-странице.

#### 5.4.1.2. Сетевой анализатор трафика

Сетевой сниффер, показанный на рисунке ниже, можно использовать для наблюдения за трафиком на физических интерфейсах, за исключением USB-устройств, таких как модемы и адаптеры Wi-Fi. Он имеет две основные настройки: Количество пакетов: это количество пакетов, которые вы хотите захватить. Настроенное значение этого параметра должно находиться в диапазоне от 1 до 25000 пакетов;

Тайм-аут простоя (секунды): Если после этого настроенного тайм-аута на интерфейсе нет пакетного трафика, выполнение анализатора прерывается. Его можно настроить со значениями от 1 до 3600 секунд.

С помощью Таблицы интерфейсов вы можете выбрать, на каких интерфейсах вы хотите, чтобы Сниффер работал, т. е. выполнял анализ сети. Вы можете выбрать все доступные интерфейсы и запустить их на всех одновременно. Для отключенных интерфейсов невозможно запустить Сниффер. Если выбранная опция отключена, в браузере будет отображаться ошибка.

Всего через несколько мгновений после открытия экрана станет доступной кнопка «Выполнить», которая запускает выполнение Сниффера. Кнопка «Загрузить» будет разблокирована только в том случае, если для загрузки доступен файл, связанный со сниффером. Если Сниффер никогда не запускался или файл удален, кнопка будет недоступна.

При запуске сетевой сниффер на странице будут отключены поля редактирования, кнопка «Загрузить» будет заблокирована, а кнопка «Выполнить» станет кнопкой «Стоп», как показано на рисунке ниже.

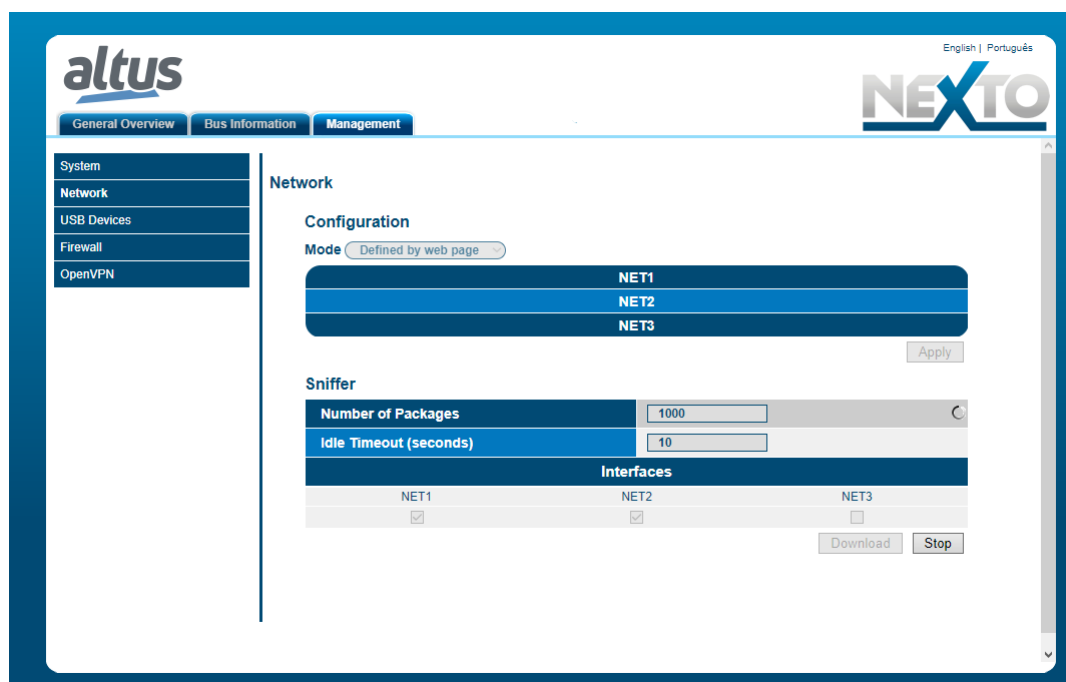


Рисунок 45: Запуск сетевого sniffера

Кнопка Stop может быть использована для прекращения выполнения sniffера в любое время после его запуска.

Для каждого интерфейса, на котором работает Sniffer, создается файл .pcap. Эти файлы названы в соответствии с именем контроллера и анализируемого интерфейса, например, NX3008\_NET1.pcap. Эти файлы находятся внутри файла .zip, также названного по имени контроллера, например, NX3008\_capture.zip.

В конце выполнения sniffера отображается сообщение с вопросом, следует ли автоматически загружать сгенерированные файлы. Эти файлы хранятся в папке InternalMemory в памяти пользовательских файлов, и к ним можно получить доступ через программное обеспечение контроллера. Загруженный файл всегда имеет расширение .zip, которое группирует другие файлы.

Если возникнут какие-либо проблемы, связанные с нехваткой памяти из-за генерации файлов sniffера, это будет указано пользователю. Рекомендуется еще раз попробовать запустить анализатор с меньшим количеством пакетов Конфигурация.

Sniffer сети может прекратить свое выполнение по трем причинам: нехватка памяти, превышение лимита времени простоя интерфейсов и ручная отмена.

## 5.5. Конфигурация интерфейса USB

Хост-порт USB, присутствующий на контроллерах серии Nexto, позволяет расширить функциональные возможности контроллера с помощью нескольких типов USB-ключей.

Управление USB-устройствами осуществляется через специальный раздел, расположенный на вкладке «Управление ПЛК» на веб-странице системы контроллера, как показано ниже:

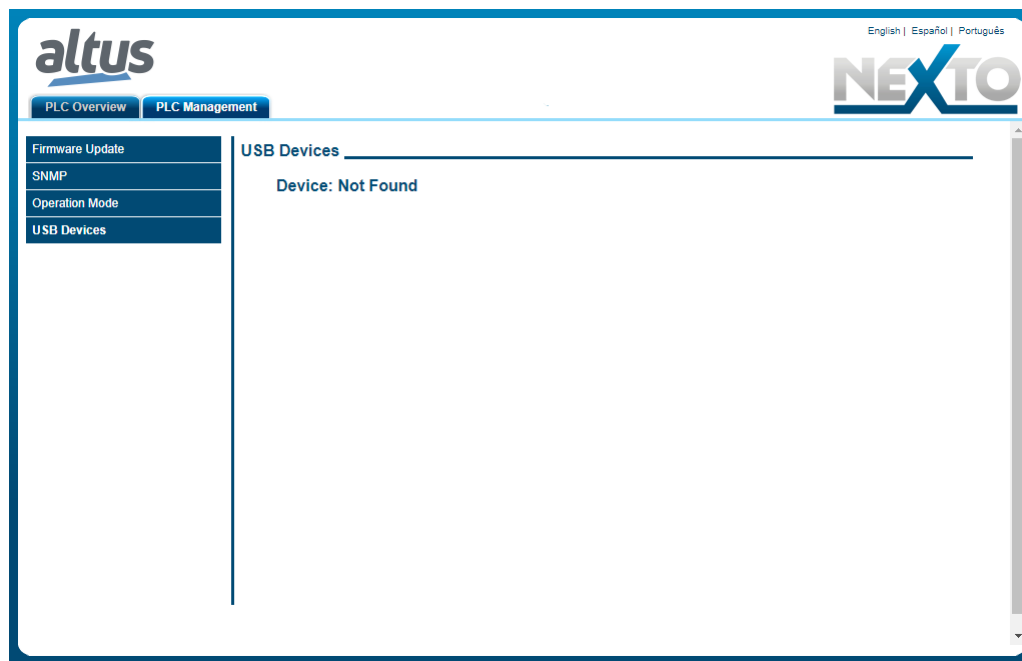


Рисунок 46: Раздел USB-устройств

Содержимое этой страницы динамически меняется в зависимости от типа подключенного USB-устройства. В приведенном выше примере нет подключенного устройства.

В следующих разделах описаны все типы USB-устройств, поддерживаемые в настоящее время. Если подключено неподдерживаемое устройство, страница сообщит, что устройство неизвестно:

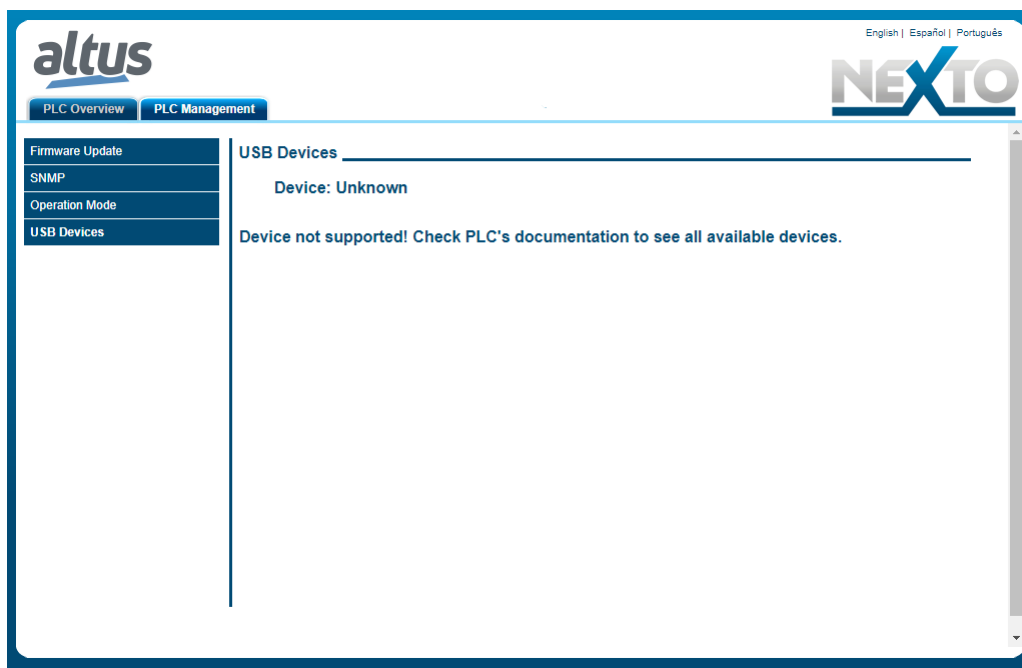


Рисунок 47: USB-устройства — неизвестно

### 5.5.1. Устройство хранения

#### 5.5.1.1. Общее хранилище

Устройства массовой памяти можно использовать для расширения флэш-памяти контроллера для хранения больших объемов данных, например, в приложениях для регистрации данных. Чтобы использовать запоминающее устройство USB, просто подключите его к порту USB. Через несколько секунд, когда устройство будет правильно обнаружено и подключено, загорится светодиод USB, и информация об устройстве появится в разделе «Устройства USB», расположенном на вкладке «Управление ПЛК» веб-страницы диагностики контроллера, как показано ниже:

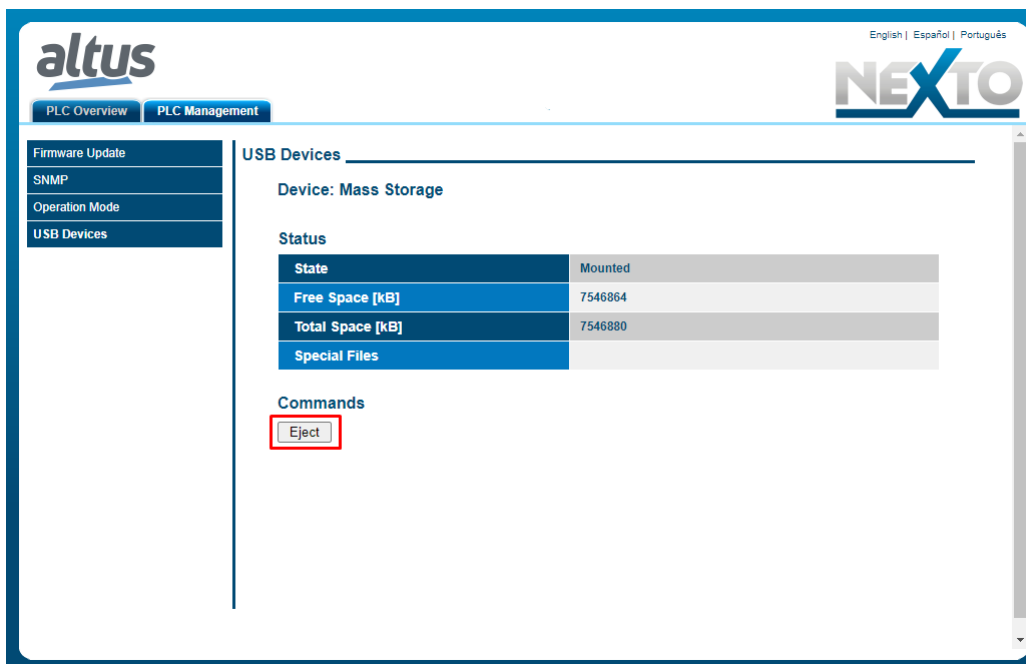


Рисунок 48: Информация о запоминающем устройстве

Информация, отображаемая в разделе состояния на этой странице, также доступна в структуре диагностики символьных переменных (см. раздел Диагностика через переменные).

#### ВНИМАНИЕ

Запоминающее устройство USB должно быть отформатировано как том FAT32. Другие форматы файловой системы не поддерживаются.

Устройство можно извлечь с помощью команды, представленной в области «Команды» на этой странице, как показано на рисунке выше.

После того, как устройство будет правильно обнаружено и смонтировано, в памяти контроллера появится новая папка Mass\_Storage, как показано на рисунке ниже:

## 5. КОНФИГУРАЦИЯ

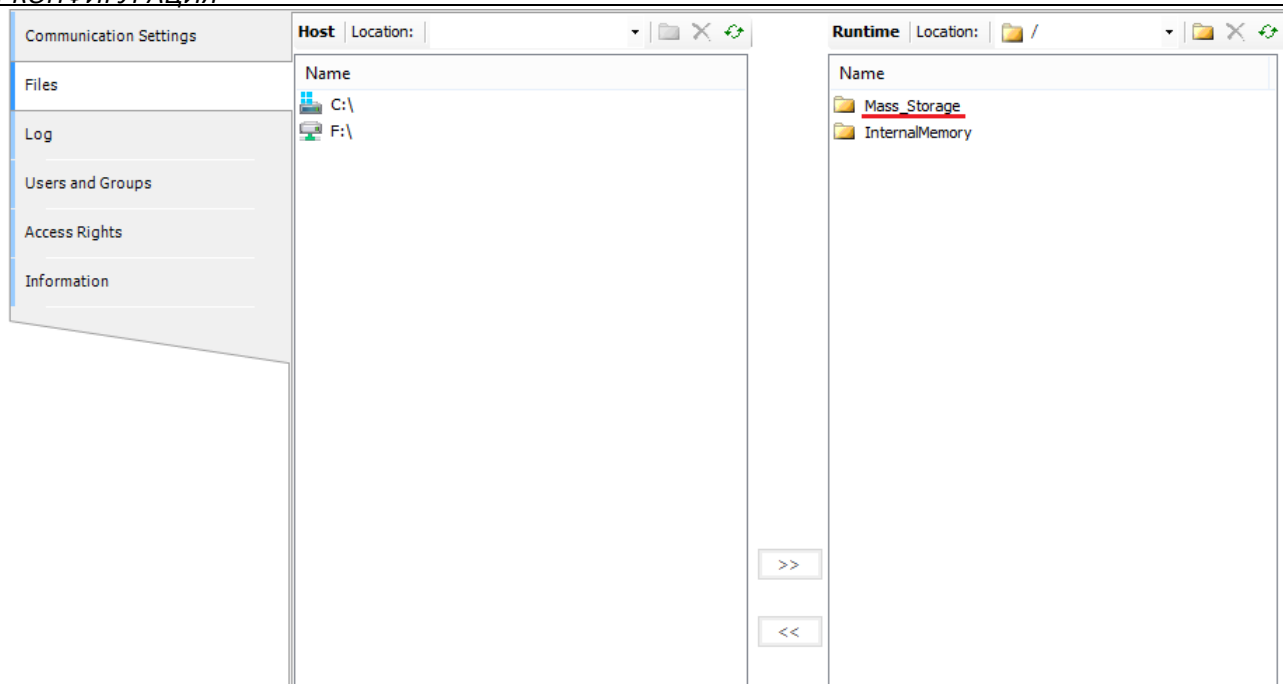


Рисунок 49: Папка для USB-накопителя

### 5.5.1.2. Приложение не загружается при запуске

Запоминающее устройство USB можно использовать для предотвращения автоматической загрузки приложения контроллером после включения питания. Для этого просто поместите пустой текстовый файл с именем «dontbootapp.txt» в корневую папку запоминающего устройства. О наличии этого файла сообщается в поле «Специальные файлы» на веб-странице системы контроллера, как показано ниже.

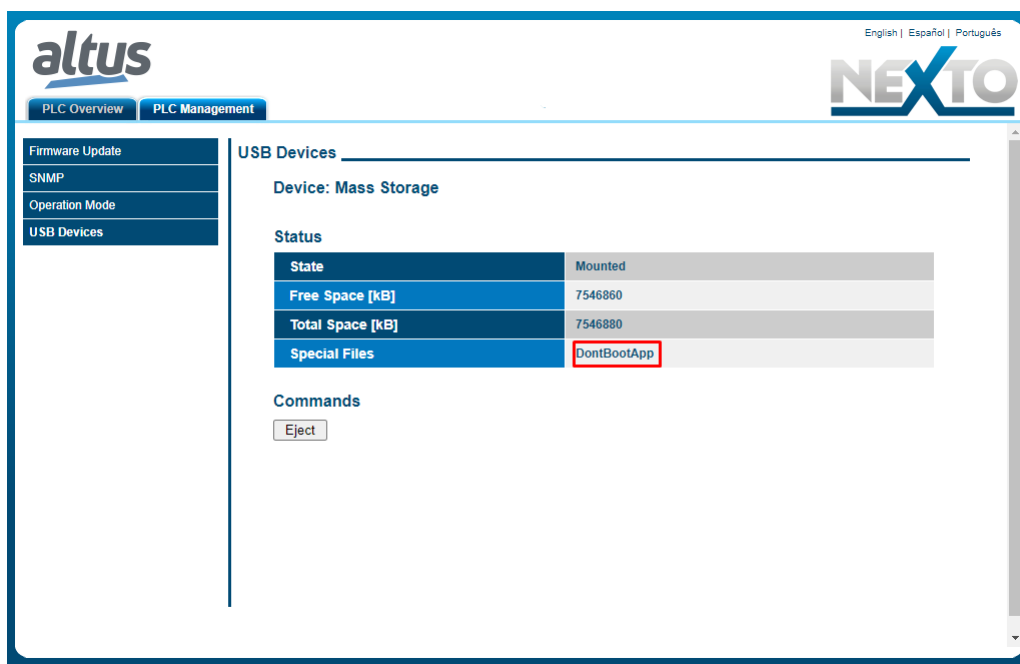


Рисунок 50: DontBootApplication на запоминающем устройстве

### 5.5.1.3. Перенос приложения с USB-устройства

Запоминающее устройство USB также можно использовать для передачи приложения на контроллер. Для этого поместите два файла Application.app и Application.crc в корневую папку запоминающего устройства (эти файлы создаются с помощью MasterTool IEC XE, выполняя команду Online -> Create boot application when offline). После включения питания, если контроллер обнаружит наличие этих файлов на запоминающем устройстве USB, произойдет следующая последовательность действий:

Контроллер начнет копирование приложения с USB-устройства во внутреннюю память. После завершения процесса копирования USB-устройство будет извлечено (USB-индикатор погаснет). Новое приложение запустится (RUN) автоматически (если «dontbootapp.txt» нет)

## 5. КОНФИГУРАЦИЯ

О наличии приложения сообщается в поле «Специальные файлы» на веб-странице системы контроллера, как показано ниже:

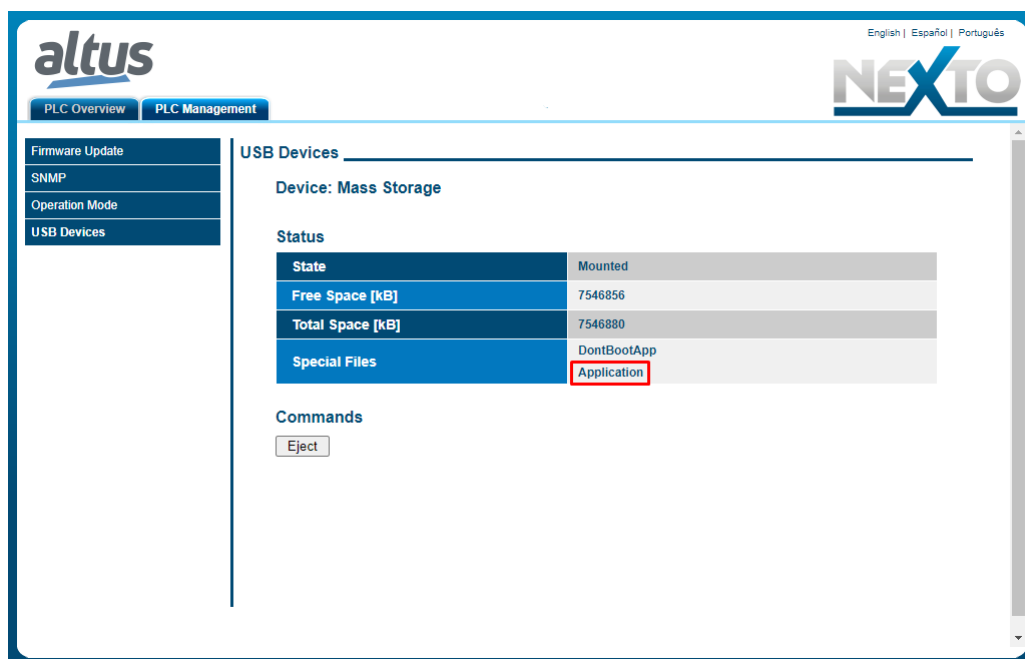


Рисунок 51: Приложение на запоминающем устройстве

Обратите внимание, что в одном устройстве Mass Storage может быть несколько специальных файлов. В приведенном выше примере ПЛК передаст новое приложение во внутреннюю память, но не загрузит его при запуске (следовательно, не перейдет в режим RUN).

### 5.5.2. Преобразователь USB в RS-232

Серия Nexto позволяет реализовать порт RS-232 с помощью преобразователя USB в последовательный порт. Эти преобразователи основаны на микросхеме внутреннего контроллера. В следующей таблице показан список поддерживаемых контроллеров:

Контроллер	Производитель
FT232	FTDI
PL2303	Prolific

Таблица 51: Поддерживаемые преобразователи USB в RS-232

Этот порт предназначен для использования исключительно с функциональными блоками последовательной связи, предоставляемыми библиотекой NextoSerial, что позволяет реализовать двухточечную связь с оборудованием, использующим простые протоколы (не критичным ко времени), таким как радиомодемы, считыватели штрих-кодов, считыватели RFID. и т. д. Кроме того, такое решение имеет следующие ограничения:

- Скорость передачи данных: значения ниже 4800 бит/с не поддерживаются.
- Биты данных: значение «5» не поддерживается (только 6, 7 или 8)
- Четность: значения «знак» и «пробел» не поддерживаются (только Odd, Even и None)
- Стопové биты: значение «1,5» не поддерживается (только 1 или 2)

После подключения преобразователя к USB-порту и правильного обнаружения и подключения устройства информация об устройстве появится в разделе «USB-устройства», расположенном на вкладке «Управление» веб-страницы диагностики контроллера, как показано ниже:

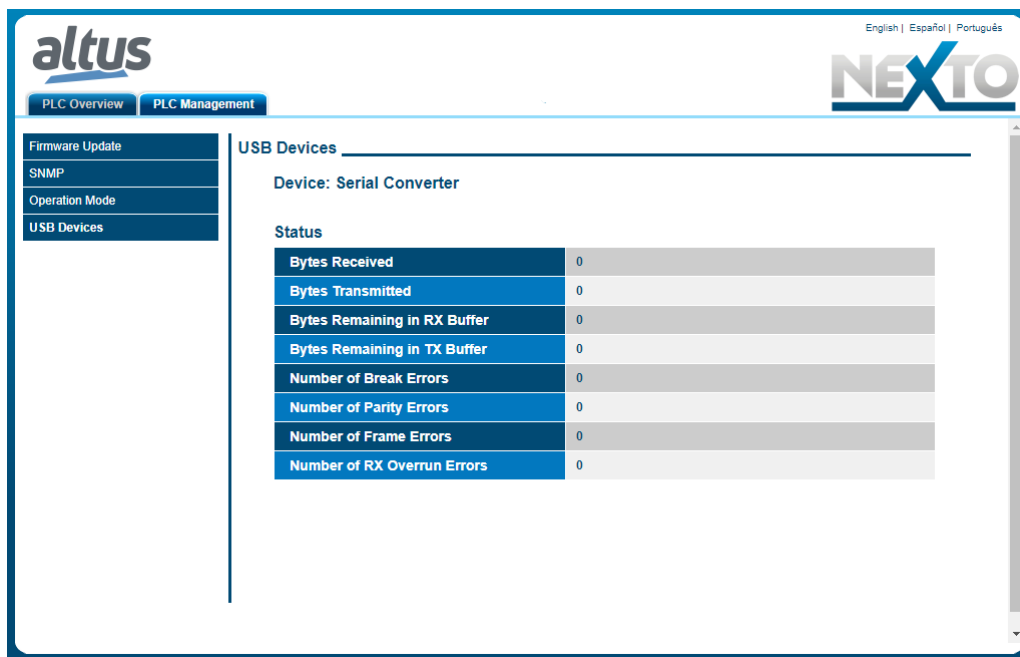


Рисунок 52: USB-устройства — последовательный преобразователь

Информация, отображаемая на этой странице, также доступна в структуре диагностики символьных переменных (см. раздел Диагностика через переменные).

Этот дополнительный последовательный порт будет внутренне идентифицирован как COM10 и не будет представлен в дереве проекта. С этого момента этот порт можно использовать для связи с использованием функций NextoSerial, аналогичных собственным портам. Для этого типа порта Конфигурация рукопожатия ограничена только RS232\_MANUAL (это необходимо учитывать при настройке порта с помощью функции SERIAL\_CFG).

### 5.5.3. Модемные устройства

USB-модем с SIM-чипом можно использовать для подключения ПЛК к Интернету через сотовую сеть передачи данных (телефонные услуги, такие как отправка SMS, не реализованы). Эта функция позволяет использовать контроллер NX3008 для реализации приложений телеметрии и IoT. В основном модемы бывают двух типов: мостовые и маршрутизаторные. В следующей таблице показан список поддерживаемых устройств.

Модель	Производитель	Тип	Примечания
E303	Huawei	Мост	-
E3272	Huawei	Мост	-
E3276	Huawei	Мост	-
E8372	Huawei	Роутер	Перенаправление веб-страницы Конфигурация (кнопка Открыть модем Конфигурация) не поддерживается для этой модели. В этом случае Конфигурацию модема необходимо производить извне, подключив его непосредственно к ПК.

Таблица 52: Поддерживаемые USB модемы

Мостовой модем — это неуправляемое устройство, которое реализует прямое (сквозное) подключение к мобильной сети передачи данных, поэтому все запросы на подключение, поступающие из Интернета, будут поступать в операционную систему контроллера. Для этого типа устройств Конфигурация выполняется через веб-страницу системы контроллера, как описано далее в этом разделе.

С другой стороны, модем-маршрутизатор — это управляемое устройство, реализующее брандмауэр с настраиваемыми сетевыми политиками. Для такого типа устройств вся Конфигурация выполняется через проприетарную веб-страницу, встроенную в устройство. По умолчанию Модем блокирует любые входящие соединения (т.е. не разрешает удаленный доступ). Чтобы разрешить это, пользователь должен настроить модем через встроенную веб-страницу устройства, чтобы открыть порт TCP, связанный с желаемой службой (некоторые производители называют эту функцию виртуальным сервером или переадресацией портов). Следующая таблица описывает номер TCP-порта, связанного с основными службами ПЛК.

Порт	Сервис
80	Веб-страница встроенной системы
8080	Встроенная веб-страница Webvisu
1217	Программирование MasterTool

Таблица 53: TCP-порты основных служб ПЛК

Управление функциями USB-модема осуществляется на странице «USB-устройства» на вкладке «Управление CP» на системной веб-странице контроллера. Как только модем будет правильно обнаружен и установлен, информация будет отображаться, как показано ниже:

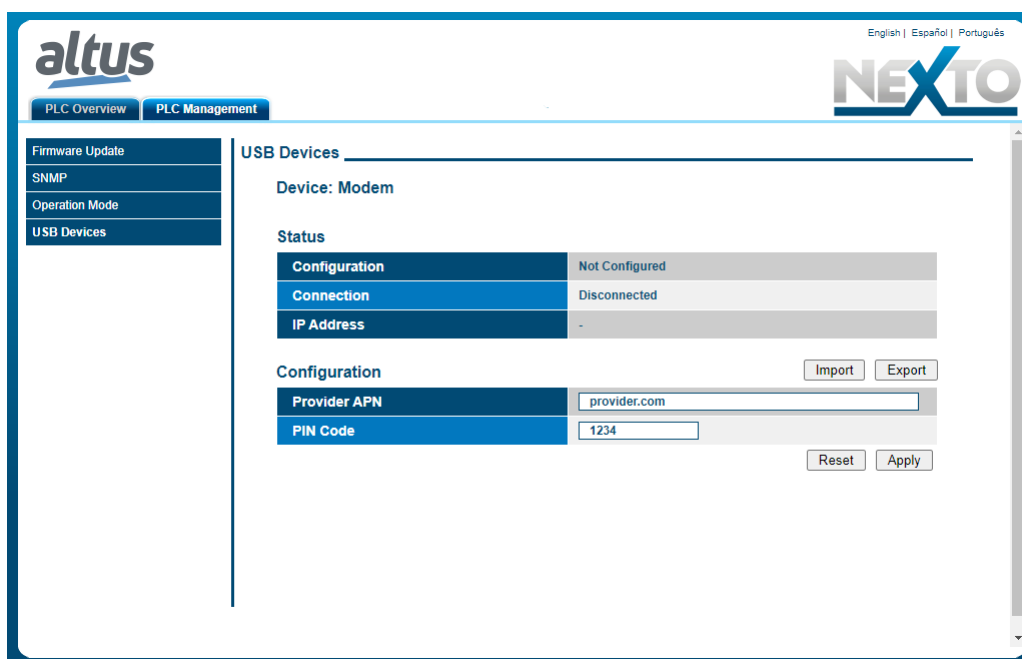


Рисунок 53: Конфигурация страницы USB-модема

Эта область состоит из двух основных разделов: Статус и Конфигурация.

В разделе «Статус» отображаются все диагностические данные, связанные с USB-модемом: состояние конфигурации (не зависящее от состояния устройства), состояние интернет-соединения и IP-адрес модема. Эти поля обновляются автоматически каждый раз, когда обновляется значение (не требует перезагрузки страницы вручную). Такая же информация содержится и в структуре диагностики символьных переменных (см. раздел Диагностика через переменные), которая также содержит подробное Описание возможных значений для каждого поля.

В разделе Конфигурация пользователь выполняет Конфигурацию модема. Для мостовых модемов этот раздел используется для ввода информации о SIM-чипе. Для модемов-маршрутизаторов в этом разделе будет отображаться только кнопка Открыть конфигурацию модема, которая будет перенаправлять на внутреннюю веб-страницу модема (проверьте в списке совместимых устройств, поддерживается ли перенаправление, в противном случае модем должен быть настроен внешне на ПК). Этот раздел содержит следующие кнопки:

- Импорт: загружает Конфигурацию из внешнего файла.

## 5. КОНФИГУРАЦИЯ

- Экспорт: сохраняет текущую Конфигурацию во внешнем файле.
- Сброс: стирает Конфигурацию модема из памяти контроллера.
- Применить: записать Конфигурацию в память контроллера

### ВНИМАНИЕ

Поля APN и PIN-кода провайдера обязательны для каждого SIM-чипа. Если провайдер сообщает эти параметры, они должны использоваться. С другой стороны, известно, что некоторые SIM-чипы просто не заботятся о содержимом этих полей, используя внутренние предопределенные значения. В этом случае эти поля веб-страницы можно оставить со значениями по умолчанию, и соединение будет успешно установлено. Такие значения, как «ноль» для PIN-кода и «пусто» для APN провайдера, не допускаются.

Принцип работы USB-модема достаточно прост. Для мостовых устройств, как только устройство правильно обнаружено и параметры SIM-карты изменены, контроллер запускает фоновый процесс, который постоянно контролирует модем, чтобы он оставался подключенным к Интернету. Это устраняет необходимость в любом ручном вмешательстве, поэтому, если соединение по какой-либо причине прервано (плохой прием, отключение несущей и т. д.), контроллер попытается восстановить соединение автоматически. Статус этого процесса можно наблюдать в поле Статус подключения. Для модемов-маршрутизаторов устройство содержит аналогичный процесс, который выполняется внутри (независимо от контроллера) и называется автоматическим подключением. Обычно он включается по умолчанию, но его можно отключить через веб-страницу модема.

Один важный аспект, который следует учитывать, заключается в том, что, если USB-модем запрещен, система контроллера установит его в качестве шлюза «По умолчанию» для всех соединений Ethernet. Это означает, что если контроллер одновременно подключен к локальной сети (NET1), которая также имеет доступ к Интернету, все Ethernet-сообщения, адресованные внешним IP-адресам, будут маршрутизироваться через USB-модем (а не через NET1). NET1 возвращается в качестве шлюза по умолчанию, когда USB-модем был удален.

Конфигурация и последовательность операций могут быть сведены к следующим шагам:

Вставьте устройство в порт USB. Через несколько секунд на странице USB-устройств устройство будет отображаться как «Модем». В противном случае устройство может быть неподдерживаемым или неисправным.

(Для мостовых устройств) Задайте параметры SIM-карты (APN провайдера — имя точки доступа и PIN-код чипа), указанные поставщиком чипа SIM-карты. После нажатия кнопки «Применить» начнется процесс подключения. После того, как вы подтвердите, больше нет необходимости устанавливать эту информацию. Он будет сохранен в памяти контроллера.

Следите за статусом соединения в соответствующем поле. Если все пройдет нормально, через несколько секунд он сообщит, что модем подключен к интернету, а в его поле появится IP-адрес.

После подключения к Интернету этот канал связи можно использовать для нескольких целей. Одним из типичных вариантов использования является реализация решения телеметрии с использованием клиентского функционального блока MQTT для публикации данных. Другой вариант использования — удаленный доступ к контроллеру. В этом случае необходимо знать IP-адрес модема в Интернете. Однако этот адрес является динамическим и изменяется при каждом процессе подключения. Один из способов обойти эту проблему — опубликовать IP-адрес (доступный в диагностических переменных модема) через MQTT.

С помощью IP-адреса модема можно осуществлять удаленный доступ к системной веб-странице контроллера для просмотра состояния и диагностики (обновление прошивки не поддерживается). Кроме того, он позволяет дистанционно программировать контроллер с помощью MasterTool IEC XE. Для этого шлюз должен быть настроен с IP-адресом модема, как показано на следующем рисунке:

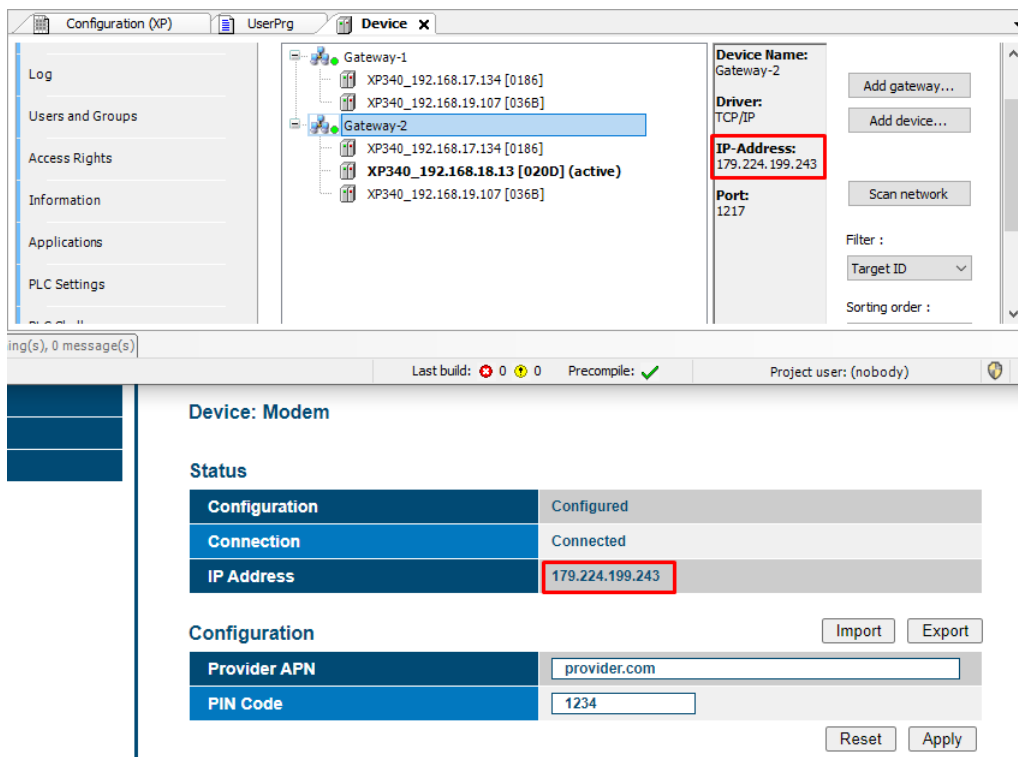


Рисунок 54: Настройка шлюза с IP-адресом модема

**ВНИМАНИЕ**

Для мостовых устройств или маршрутизаторов с включенным внешним доступом (переадресация портов) после подключения к Интернету любой, кто знает IP-адрес модема, сможет получить удаленный доступ к контроллеру. Таким образом, из соображений безопасности ЧРЕЗВЫЧАЙНО важно и рекомендуется настроить права пользователя на контроллере, чтобы ограничить онлайн-операции MasterTool IEC XE.

**5.5.4. Адаптеры WiFi**

Адаптер USB WiFi можно использовать для подключения ПЛК к существующей сети WiFi, создавая второй сетевой адаптер, который можно использовать для программирования и связи. В следующей таблице показан список поддерживаемых чипсетов.

Чипсет	Производитель	Пример коммерческих продуктов
RTL8188EU	Realtek	TP-LINK модель TL-WN725N LM Technologies модель LM007
RT28xx	Ralink/Mediatek	D-Link модель DWA-125
AR9271	Atheros/Qualcomm	TP-LINK модель TL-WN721N

Таблица 54: Поддерживаемые наборы микросхем для адаптеров USB WiFi

Управление функциями WiFi-адаптера осуществляется на странице «USB-устройства» на вкладке «Управление ПЛК» на системной веб-странице контроллера. Как только адаптер Wi-Fi будет правильно обнаружен и установлен, загорится индикатор USB, и информация об устройстве появится, как показано ниже:

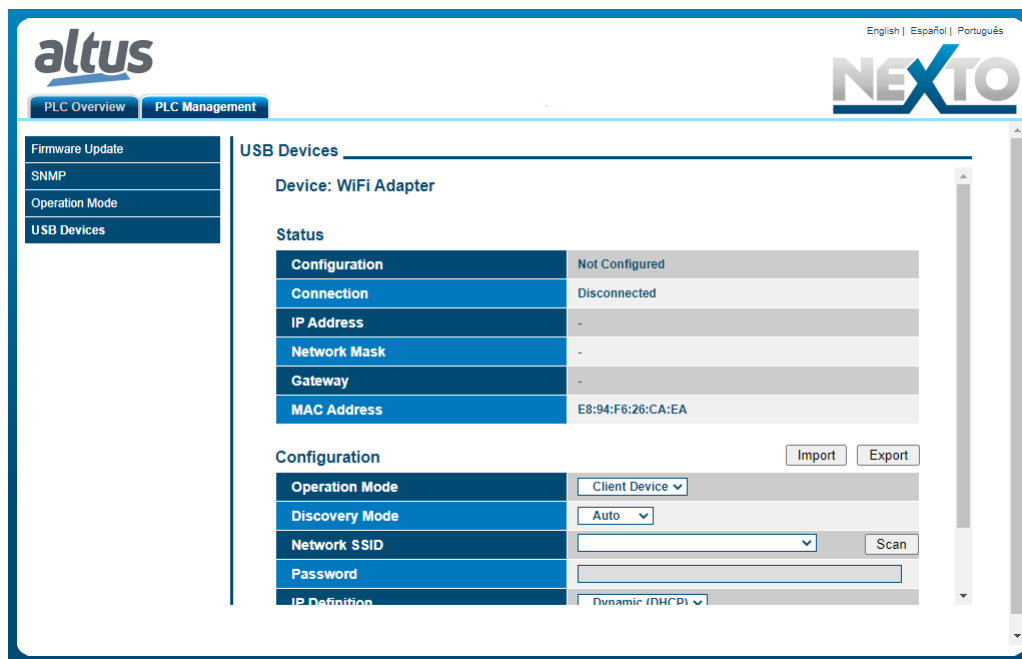


Рисунок 55: Страница USB-адаптера Wi-Fi

Эта страница состоит из двух основных разделов: Статус и Конфигурация.

В разделе «Состояние» отображаются все диагностические данные, связанные с адаптером WiFi: состояние конфигурации, состояние подключения, IP-адрес, сетевая маска, шлюз и MAC-адрес. Эти поля обновляются при изменении значений. Эта информация также предоставляется в структуре диагностики символьных переменных (см. раздел Диагностика через переменные).

Раздел Конфигурация состоит из следующих параметров:

Режим работы: определяет, как будет работать WiFi-адаптер (в настоящее время поддерживается только клиентский режим).

Режим обнаружения: определяет метод настройки сети Wi-Fi. Если выбрано значение «Авто», для выбора нужной сети необходимо использовать кнопку «Сканировать». Если выбрано «Вручную», имя SSID и тип безопасности необходимо ввести вручную.

SSID сети: когда режим обнаружения установлен на «Авто», в этом поле будут отображаться доступные сети, найденные в процессе сканирования, отсортированные от лучшего к худшему уровню сигнала (сверху вниз). Когда режим обнаружения установлен на «Ручной», это поле должно быть заполнено SSID нужной сети.

Тип безопасности: это поле доступно только в том случае, если режим обнаружения установлен на «Ручной» (режим «Авто» автоматически выбирает тип безопасности, обеспечиваемый сканируемой сетью). Это поле определяет тип безопасности, используемый в сети Wi-Fi, который может быть «Общедоступным» или «WPA2-Personal».

Пароль: здесь вам нужно ввести пароль сети Wi-Fi. Поле будет автоматически заблокировано, если для параметра «Тип безопасности» установлено значение «Общедоступный» или выбранная сканируемая сеть не использует протокол безопасности.

Определение IP: определяет, будет ли адаптер Wi-Fi устанавливать IP-адрес динамически (назначается сетевым DHCP-сервером) или статически (когда пользователю необходимо ввести настройки IP вручную).

IP-адрес, маска сети и шлюз: доступны только в том случае, если для определения IP задано значение «Статический». Эти поля будут использоваться для настройки сетевых параметров адаптера Wi-Fi.

По умолчанию Шлюз: это поле определяет, какой сетевой интерфейс будет использоваться в качестве Шлюза для выхода в Интернет. Для этой функции можно выбрать «адаптер Wi-Fi» или «локальный Ethernet».

#### ВНИМАНИЕ

Для правильной работы сеть WiFi-адаптера (определяемая по IP и маске) должна отличаться от настроенной для NET1.

Рисунок 56: Конфигурация USB-адаптера Wi-Fi

Кроме того, раздел Конфигурация содержит следующие кнопки:

- Импорт: загружает файл конфигурации
- Экспорт: загрузить файл с текущей Конфигурацией
- Сброс: возвращает Конфигурацию к По умолчанию
- Применить: применить текущую Конфигурацию

После того, как устройство правильно обнаружено и сетевые параметры настроены, контроллер всегда будет пытаться оставаться подключенным к сети WiFi. Статус этого процесса можно наблюдать в поле Статус подключения.

Конфигурация и последовательность операций могут быть сведены к следующим шагам:

#### ВНИМАНИЕ

Если шлюз по умолчанию был установлен как «адаптер WiFi», шлюз диагностики NET1 MasterTool покажет ноль (0.0.0.0). В противном случае, если он был установлен как «Локальный Ethernet», шлюз адаптера WiFi будет нулевым.

Вставьте устройство в порт USB. Через несколько секунд на странице USB-устройств устройство будет отображаться как «адаптер WiFi». В противном случае устройство может быть неподдерживаемым или неисправным.

Установить сеть Конфигурация. После нажатия кнопки «Применить» начнется процесс подключения. После того, как вы подтвердите, больше нет необходимости устанавливать эту информацию. Он будет сохранен в памяти контроллера.

Следите за статусом соединения в соответствующем поле. Если все пойдет хорошо, через несколько секунд он сообщит, что адаптер подключен к сети WiFi.

На следующем рисунке показана страница контроллера, подключенного к сети WiFi:

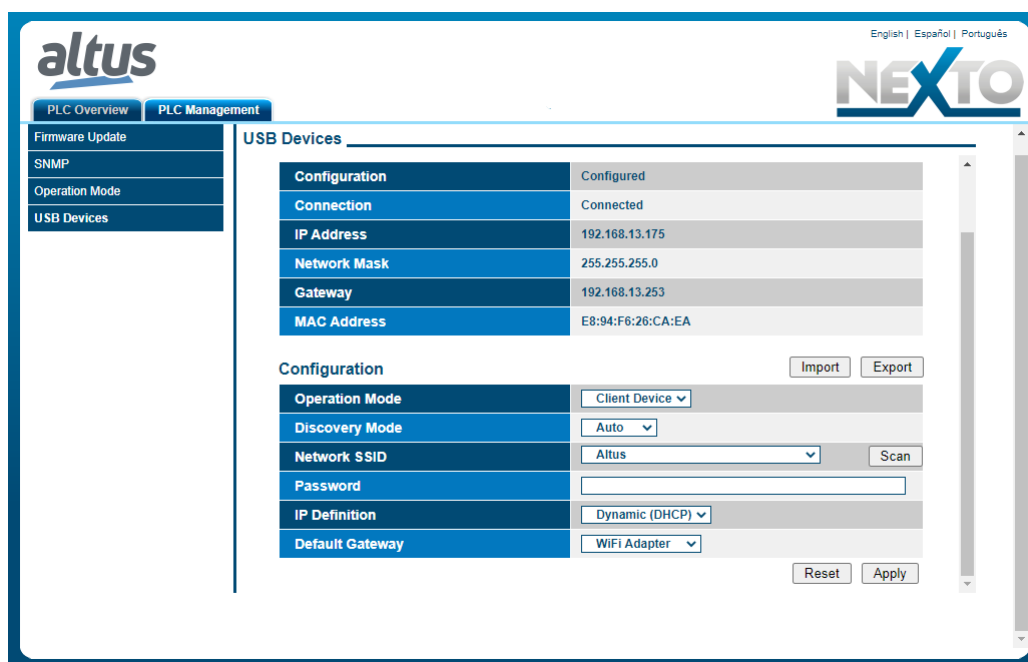


Рисунок 57: USB-адаптер Wi-Fi, подключенный к сети

После подключения к сети WiFi этот канал связи можно использовать для нескольких целей. Для программирования ПЛК с помощью MasterTool IEC XE шлюз должен быть связан с IP-адресом, присвоенным WiFi-адаптеру (аналогично USB-модему, рисунок 54). Этот IP-адрес также можно использовать для доступа к системной веб-странице контроллера, где можно выполнить обновление прошивки. Кроме того, этот канал связи также можно использовать с функциональным блоком клиента MQTT для передачи данных внешнему брокеру в Интернете (в данном случае адаптер WiFi в качестве шлюза по умолчанию).

## 5.6. Конфигурация Интерфейсов Ethernet

ЦП NX3008 имеет NET 1, NET 2 и NET 3. В дополнение к локальным интерфейсам Ethernet серия Nexto также предоставляет удаленные интерфейсы Ethernet благодаря включению модуля NX5000. Модули NX5000 имеют только интерфейс NET 1.

### 5.6.1. Внутренние интерфейсы Ethernet

#### 5.6.1.1. NET 1

Этот интерфейс состоит из коммуникационного разъема RJ45 стандарта 10Base-TE/100Base-TX/1000BASE-T. Он обеспечивает двухточечную или сетевую связь по следующим открытым протоколам: клиент MODBUS TCP, MODBUS RTU через TCP-клиент, MODBUS TCP-сервер и MODBUS RTU через TCP-сервер.

Параметры, которые необходимо настроить для правильной работы приложения, описаны ниже.

Конфигурация	Описание	По умолчанию	Варианты
Получить IP-адрес автоматически	Включает функцию DHCP-клиента на устройстве для автоматического назначения IP-адресов.	Без опознавательных знаков	С опознавательными знаками и без опознавательных знаков
Айпи адрес	IP-адрес контроллера в шине Ethernet	192.168.15.1	1.0.0.1 to 223.255.255.254
Маска подсети	Маска подсети контроллера в шине Ethernet	255.255.255.0	128.0.0.0 до 255.255.255.252
Адрес шлюза	Адрес шлюза контроллера в шине Ethernet	192.168.15.253	1.0.0.1 to 223.255.255.254

Таблица 55 Локальная конфигурация NET 1

**5.6.1.2. NET 2**

Этот интерфейс состоит из коммуникационного разъема RJ45 стандарта 10Base-TE/100Base-TX/1000BASE-T. Он обеспечивает двухточечную или сетевую связь по следующим открытым протоколам: клиент MODBUS TCP, MODBUS RTU через TCP-клиент, MODBUS TCP-сервер и MODBUS RTU через TCP-сервер.

Параметры, которые необходимо настроить для правильной работы приложения, описаны ниже.

Конфигурация	Описание	По умолчанию	Варианты
Адрес IP	IP-адрес контроллера в шине Ethernet	192.168.16.1	1.0.0.1 to 223.255.255.254
Маска подсети	Маска подсети контроллера в шине Ethernet	255.255.255.0	128.0.0.0 до 255.255.255.252
Адрес шлюза	Адрес шлюза контроллера в шине Ethernet	192.168.16.253	1.0.0.1 до 223.255.255.254

Таблица 56: Конфигурация NET 2

**ВНИМАНИЕ**

Невозможно настроить более одного интерфейса Ethernet ЦП в одной и той же подсети, и этот тип конфигурации блокируется инструментом MasterTool. Следовательно, каждый интерфейс Ethernet должен быть настроен в другой подсети.

**5.6.1.3. NET 3**

Этот интерфейс состоит из коммуникационного разъема RJ45 стандарта 10Base-TE/100Base-TX/1000BASE-T. Он обеспечивает двухточечную или сетевую связь по следующим открытым протоколам: клиент MODBUS TCP, MODBUS RTU через TCP-клиент, MODBUS TCP-сервер и MODBUS RTU через TCP-сервер.

Параметры, которые необходимо настроить для правильной работы приложения, описаны ниже.

Конфигурация	Описание	По умолчанию	Варианты
Адрес IP	Адрес IP контроллера в шине Ethernet	192.168.17.1	1.0.0.1 до 223.255.255.254
Маска подсети	Маска подсети контроллера в шине Ethernet	255.255.255.0	128.0.0.0 до 255.255.255.252
Адрес шлюза	Адрес шлюза контроллера в шине Ethernet	192.168.17.253	1.0.0.1 до 223.255.255.254

Таблица 57: NET 3 Конфигурация

**5.6.1.4. Конфигурация внутренних интерфейсов Ethernet**

Каналы Ethernet ЦП могут работать в трех различных режимах: одиночный режим, резервный режим или режим переключения. В отличие от других процессоров серии Nexto, NX3008 имеет три порта Ethernet (NET 1, NET 2 и NET 3).

По умолчанию интерфейс NET 1 нельзя использовать ни в каком режиме, кроме простого. NET 2 имеет все три варианта конфигурации, а NET 3 зависит от установленных настроек NET 2.

Когда NET 2 определяется как Redundant Mode или Switch Mode, интерфейсы NET 2 и NET 3 работают как резервная пара/коммутатор, и вся информация хранится в паре NET, в данном случае NET 2. И в этой архитектуре, в NET 3 все поля Конфигурация автоматически отключены.

Если NET 2 отключен или обозначен как Простой режим, интерфейс NET 3, если он включен, работает как Простой режим.

**5.6.1.4.1. Одиночный режим**

В этом режиме интерфейс работает как независимый порт Ethernet, не связанный с последующим интерфейсом.

Enable Interface ☒

Obtain an IP address automatically (Single Mode) ☐

Ethernet Port Parameters

IP Address: 192 . 168 . 15 . 1

Subnetwork Mask: 255 . 255 . 255 . 0

Gateway Address: 192 . 168 . 15 . 253

Mode

☒ Single ☐ Redundant ☐ Switch

## 5.6.1.4.2. Резервный режим

В этом режиме интерфейс образует резервную пару с последующим интерфейсом, работая по схеме активный/резервный. Пара резервирования связи имеет один IP-адрес, связанный с активным в данный момент интерфейсом. Таким образом, клиент в SCADA или Mastertool, подключенный к серверу на ЦП, не должен беспокоиться об изменении адреса IP, если какой-либо из портов в пире выйдет из строя. Кроме того, каждый из интерфейсов, образующих резервированную коммуникационную пару, имеет отдельную диагностику для облегчения отладки неисправностей, которые могут в конечном итоге возникнуть.

Рисунок 59: Расширенная конфигурация локальных интерфейсов Ethernet — режим резервирования

## Расширенная конфигурация локальных интерфейсов Ethernet — режим резервирования

Когда выбран резервный режим, на том же экране автоматически включаются другие параметры, которые необходимо настроить:

Период проверки избыточности (мс): Период отправки кадра проверки связи между двумя сетями NET. Можно настроить значения от 100 до 9900, по умолчанию 500

Повторы теста избыточности: максимальное количество раз, когда сеть, отправившая кадр, будет ожидать ответа. Может быть настроен на значения от 1 до 100, По умолчанию 4

Switching Period (s): максимальное время, в течение которого NET Active будет ожидать любого заданного пакета. Может быть настроен со значениями от 1 до 25, По умолчанию 10

Если время отклика теста избыточности достигает периода теста, умноженного на количество повторных попыток, а активный интерфейс остается дольше, чем период переключения, не получая никаких пакетов, происходит переключение, делающее ранее неактивный интерфейс активным. Важно отметить, что существует задержка между обнаружением неисправности и активацией неактивного интерфейса из-за времени, необходимого для его Конфигурации. Эта задержка может достигать нескольких десятков миллисекунд.

Когда одна из сетей активна, она примет настроенный IP-адрес адреса, а неактивная сеть останется с пустыми параметрами IP-адреса, маски подсети и адреса шлюза в диагностике PCU.

**ВНИМАНИЕ**

Когда команда Reset Origem выполняется на ЦП, сконфигурированном с резервным режимом на внешних интерфейсах NETx, останется доступным только интерфейс, который был активен до команды. После команды доступный интерфейс можно просмотреть в информационном и конфигурационном меню ЦП.

## 5.6.1.4.3. Режим переключения

В этом режиме интерфейс образует пару с последующим интерфейсом, работающим как Ethernet-коммутатор, что позволяет осуществлять связь через оба порта. Таким образом, этот режим позволяет «каскадировать» несколько ЦП, что позволяет реализовать топологию сети «кольцо». Это позволяет использовать два варианта конфигурации в режиме переключения.

## 5.6.1.4.4. Режим переключения — отключен

В этом режиме сети 2 и 3 ЦП действуют как простой коммутатор без какой-либо защиты от образования петель.

The screenshot shows the 'Ethernet Port Configuration' window. The 'Enable Interface' checkbox is checked. Below it, the 'Obtain an IP address automatically (Single Mode)' checkbox is unchecked. The 'Ethernet Port Parameters' section contains three input fields: 'IP Address' (192.168.15.1), 'Subnetwork Mask' (255.255.255.0), and 'Gateway Address' (192.168.15.253). The 'Mode' section has three radio buttons: 'Single', 'Redundant', and 'Switch' (which is selected). Below the 'Mode' section is the 'Switch Mode Configuration' section, which contains a dropdown menu for 'Loop Protection Mode' set to 'Disabled'.

Рисунок 60: Расширенная конфигурация локальных интерфейсов Ethernet — Режим коммутатора — Отключено

## 5.6.1.4.5. Режим переключения — RSTP

В этом режиме пара сетевых интерфейсов использует протокол Rapid Spanning-Tree (RSTP) для обработки сетевых петель и резервирования связи между устройствами. Если выбрано, устройство принудительно использует протокол RSTP на мосту. Однако, если кадр STP получен его портами, протокол Spanning-Tree автоматически понижается до протокола Spanning-Tree (STP), таким образом поддерживая связь с сетью.

The screenshot shows the 'Ethernet Port Configuration' window with the 'Switch' mode selected. The 'Loop Protection Mode' dropdown is now set to 'RSTP'. This has triggered the appearance of several additional configuration fields: 'Bridge Priority' (32768), 'Max Age' (20), 'Hello Time (s)' (2), 'Forward Delay (s)' (15), 'Tx Hold Count' (6), and 'Ports Path Cost' (200000). The 'Auto Ports Path Cost' checkbox is also checked.

Рисунок 61: Расширенная настройка локальных интерфейсов Ethernet — режим коммутатора — RSTP  
Когда режим защиты от петель установлен на RSTP, другие параметры автоматически включаются на том же экране и должны быть настроены:

**Bridge Priority:** устанавливает приоритет моста. Чем ниже значение, тем выше приоритет. Значение по умолчанию — 32768. Максимальный возраст: максимальное значение возраста BPDU. При работе в режиме RSTP это значение представляет собой максимальное количество мостов между корнем и мостом, которые получили BPDU. Однако при работе в режиме STP

это максимальное время, в течение которого он считает кадр BPDU действительным. Его можно настроить со значениями от 6 до 40, значение по умолчанию равно 20. Значение должно соответствовать следующему правилу: максимальный возраст  $\leq 2 \times (\text{задержка пересылки} - 1)$ .

**Hello Time (s):** Период отправки кадров BPDU. Может быть настроен со значениями от 1 до 10, значение по умолчанию равно 2.

**Задержка пересылки (с):** время, необходимое корневому порту и назначенному порту для перехода в состояние пересылки при работе в режиме STP. Это может быть подтверждено значениями от 4 до 30, значение по умолчанию равно 15.

## 5. КОНФИГУРАЦИЯ

Tx Hold Count: максимальное количество кадров BPDU, которые мост может отправлять за каждое время приветствия. Может быть установлено значение от 1 до 10, значение по умолчанию равно 6.

Стоимость пути портов: устанавливает стоимость портов, которые будут использоваться в сетевом пути RSTP/STP. Может принимать значения от 1 до 20000000, значение по умолчанию равно 200000.

Auto Ports Path Cost: если этот параметр включен, стоимость портов устанавливается автоматически в соответствии со скоростью их соединения, что соответствует  $(2 \times 1012) / (\text{скорость соединения})$ . В этой Конфигурации порты могут брать на себя разную стоимость. При отключении оба порта будут использовать одинаковую стоимость, установленную в предыдущем параметре.

### 5.6.2. Удаленный Ethernet-интерфейс NX5000

#### 5.6.2.1. NET 1

Интерфейс NET 1 состоит из коммуникационного разъема RJ45 стандарта 10/100Base-TX. Он обеспечивает двухточечную или сетевую связь по следующим открытым протоколам, например: MODBUS TCP Client, MODBUS RTU через TCP Client, MODBUS TCP Server и MODBUS RTU через TCP Server.

Параметры, которые необходимо учитывать для правильной работы приложения, описаны ниже.

Конфигурация	Описание	По умолчанию	Варианты
Адрес IP	Адрес IP контроллера в шине Ethernet	192.168.xx.68	1.0.0.1 до 223.255.255.254
Маска подсети	Маска подсети контроллера в шине Ethernet	255.255.255.0	128.0.0.0 до 255.255.255.252
Адрес шлюза	Адрес шлюза контроллера в шине Ethernet	192.168.xx.253	1.0.0.1 to 223.255.255.254

Таблица 58: Конфигурация удаленной сети NX5000 1

#### 5.6.2.2. Режимы работы удаленного интерфейса Ethernet NX5000

Модули NX5000 могут быть вставлены в проект для увеличения количества интерфейсов Ethernet, если локальных интерфейсов ЦП недостаточно.

Каналы Ethernet модулей NX5000 можно использовать по отдельности или организовать резервными парами.

##### 5.6.2.2.1. Резервный режим

Пара из двух портов Ethernet, образующих избыточную пару, имеет один IP-адрес, привязанный к паре портов. Таким образом, клиент, такой как SCADA или MasterTool, подключенный к серверу на ЦП, не должен беспокоиться об изменении IP-адреса в случае отказа некоторых портов в избыточной паре.

Чтобы объединить два модуля NX5000 в качестве резервной пары, эти два модуля должны обязательно занимать соседние позиции в стойке объединительной платы, и должен быть установлен флажок Redundant из модуля слева, как показано на рисунке ниже. При этом блокируется редактирование параметров модуля справа. Параметры, редактируемые в модуле, вставленном слева, становятся общими для двух модулей.

С другой стороны, снятие флажка Redundant с модуля слева приводит к разделению модулей, которые снова ведут себя как отдельные модули без избыточности.

Когда выбран резервный режим, на том же экране автоматически включаются другие параметры, которые необходимо настроить:

Период проверки избыточности (мс): Период отправки кадра проверки связи между двумя сетями NET. Можно настроить значения от 100 до 9900, по умолчанию 500

Повторы теста избыточности: максимальное количество раз, когда сеть, отправившая кадр, будет ожидать ответа. Может быть настроен на значения от 1 до 100, По умолчанию 4

Switching Period (s): максимальное время, в течение которого NET Active будет ожидать любого заданного пакета. Может быть настроен со значениями от 1 до 25, По умолчанию 10

Если время отклика теста избыточности достигает периода теста, умноженного на количество повторных попыток, а активный интерфейс остается дольше, чем период переключения, не получая никаких пакетов, происходит переключение, делающее ранее неактивный интерфейс активным. Важно отметить, что существует задержка между обнаружением неисправности и активацией неактивного интерфейса из-за времени, необходимого для его Конфигурации. Эта задержка может достигать нескольких десятков миллисекунд.

Когда одна из сетей активна, она примет настроенный IP-адрес адреса, а неактивная сеть останется с пустыми параметрами IP-адреса, маски подсети и адреса шлюза в диагностике ЦП.

#### 5.6.3. Зарезервированные порты TCP/UDP

Следующие порты TCP/UDP интерфейсов Ethernet, как локальные, так и удаленные, используются службами ЦП (в зависимости от доступности в соответствии с таблицами протоколов) и, следовательно, зарезервированы и не должны использоваться пользователем.

Сервис	TCP	UDP
Диагностики веб-страницы	80	-

SNTp	-	123
SNMP	-	161
MODBUS TCP	502*	-
MasterTool MT8500	1217*	1740:1743
SQL сервер	1433	-
MQTT	1883* / 8883*	-
EtherNet/IP	44818	2222
IEC 60870-5-104	2404*	-
OPC UA	4840	-
WEBVISU	8080	-
CODESYS ARTI	11740	-
PROFINET	-	34964

Таблица 59: Зарезервированные порты TCP/UDP

\* Порт по умолчанию, но может быть изменен пользователем.

## 5.7. Конфигурация протоколов

Независимо от протоколов, используемых в каждом приложении, ЦП серии Nexto имеют некоторые максимальные ограничения для каждой модели ЦП. В основном существует два различных типа протоколов связи: символическое отображение и отображение прямого представления. Максимальный лимит сопоставлений, а также максимальное количество протоколов (экземпляров) определяется в Таблице ниже:

	<b>NX3008</b>
<b>Сопоставленные точки</b>	20480
<b>Сопоставления (на экземпляр / всего)</b>	5120 / 20480
<b>Запросы</b>	512
<b>Сети — экземпляры клиента или сервера (на сеть / всего)</b>	4 / 16
<b>COM (n) — главные или подчиненные экземпляры</b>	1
<b>Центры управления</b>	3

Таблица 60: Ограничения протоколов на ЦП

### Примечания:

**Сопоставленные точки:** это относится к максимальному количеству сопоставленных точек, поддерживаемых ЦП. Каждое отображение поддерживает одну или несколько отображаемых точек, в зависимости от размера данных при использовании с переменными типа ARRAY.

**Сопоставления:** «Отображение» — это отношение между внутренней переменной приложения и объектом протокола приложения. В этом поле указывается максимальное количество отображений, поддерживаемых ЦП. Это соответствует сумме всех сопоставлений, выполненных в экземплярах протоколов связи и их соответствующих устройств.

**Запросы:** Сумма запросов протоколов связи, заявленных на устройствах, не может превышать максимальное количество запросов, поддерживаемых ЦП.

**NETs — Clients or Servers Instances:** это поле определяет максимальное количество экземпляров протокола на интерфейс Ethernet, а также общее максимальное количество экземпляров, распределенных по всем интерфейсам Ethernet системы.

**COM (n) — экземпляры Master или Slave:** в силу своих характеристик каждый последовательный интерфейс поддерживает только один экземпляр протокола связи. Примеры экземпляров, совместимых с последовательными интерфейсами: MODBUS RTU Master и MODBUS RTU Slave.

**Центры управления:** «Центр управления» — это все клиентские устройства, подключенные к ЦП по протоколу IEC 60870-5-104. В этом поле указывается максимальное количество клиентских устройств типа центра управления, поддерживаемых ЦП. Соответствует сумме всех клиентских устройств протокола связи IEC 60870-5-104 Server (не включает ведущее устройство или клиенты протоколов MODBUS RTU Slave, MODBUS Server и DNP3 Server).

Ограничения протокола MODBUS для прямого представления и символического отображения для процессоров можно увидеть в таблицах 61 и 62 соответственно.

## 5. КОНФИГУРАЦИЯ

Ограничения	MODBUS RTU Master	MODBUS RTU Slave	MODBUS Ethernet Client	MODBUS Ethernet Server
Сопоставления на экземпляр	128	32	128	32
Устройств на экземпляр	64	1 <sup>(1)</sup>	64	64 <sup>(2)</sup>
Сопоставления на устройство	32	32	32	32
Одновременных запросов на экземпляр	-	-	128	64
Одновременные запросы на устройство	-	-	8	64

Таблица 61: Ограничения протокола MODBUS для прямого представления

### Примечания:

#### Устройств на экземпляр:

Ведущие или клиентские протоколы: количество ведомых или серверных устройств, поддерживаемых каждым экземпляром ведущего или ведомого протокола.

Ведомый протокол MODBUS RTU: указанное ограничение (1) относится к последовательным интерфейсам, которые не позволяют ведомому устройству устанавливать связь через один и тот же последовательный интерфейс одновременно с более чем одним ведущим устройством. Нет необходимости и невозможно объявить или констатировать Ведущее устройство в случае ведомого протокола MODBUS RTU. Ведущее устройство будет иметь доступ ко всем сопоставлениям, выполненным непосредственно на экземпляре ведомого протокола MODBUS RTU.

Протокол сервера MODBUS RTU: указанное ограничение (2) относится к интерфейсам Ethernet, которые ограничивают количество соединений, которые могут быть установлены с другими устройствами через один интерфейс Ethernet. Нет необходимости и невозможно объявлять или настроен клиентских устройств в экземпляре протокола MODBUS Server. Все клиентские устройства будут иметь доступ ко всем сопоставлениям, выполненным непосредственно в экземпляре протокола MODBUS Server.

Сопоставления на устройство: максимальное количество сопоставлений на устройство, несмотря на то, что оно указано выше, также ограничено максимальным числом сопоставлений протокола. Также следует учитывать максимальные сопоставления ЦП, как в Таблице 5.7.

Одновременных запросов на экземпляр: количество запросов, которые могут быть одновременно переданы каждым экземпляром клиентского протокола или которые могут быть получены одновременно каждым экземпляром протокола сервера. Экземпляры протокола MODBUS RTU, Master или Slave, не поддерживают одновременные запросы.

Одновременных запросов на устройство: Количество запросов, которые могут быть одновременно переданы каждому серверному устройству MODBUS или могут быть получены одновременно каждым клиентским устройством MODBUS. Устройства MODBUS RTU, главные или ведомые, не поддерживают одновременные запросы.

Ограничения	MODBUS RTU Master	MODBUS RTU Slave	MODBUS Ethernet Client	MODBUS Ethernet Server
Устройств на экземпляр	64	1 <sup>(1)</sup>	64	64 <sup>(2)</sup>
Запросов на устройство	32	-	32	-
Одновременных запросов на экземпляр	-	-	128	64
Одновременные запросы на устройство	-	-	8	64

Таблица 62: MODBUS Protocol Limitations for Symbolic Mappings

### Примечания:

#### Устройств на экземпляр:

Ведущий или клиентский протокол: количество ведомых или серверных устройств, поддерживаемых каждым ведущим или клиентским протоколом.

Ведомый протокол MODBUS RTU: указанное ограничение (1) относится к последовательным интерфейсам, которые не позволяют ведомому устройству устанавливать связь через один и тот же последовательный интерфейс одновременно с более чем одним ведущим устройством. Нет необходимости и невозможно объявить или констатировать Ведущее устройство в случае ведомого протокола MODBUS RTU. Ведущее устройство будет иметь доступ ко всем сопоставлениям, выполненным непосредственно на экземпляре ведомого протокола MODBUS RTU.

Протокол сервера MODBUS RTU: указанное ограничение (2) относится к интерфейсам Ethernet, которые ограничивают количество соединений, которые могут быть установлены с другими устройствами через один интерфейс Ethernet. Нет необходимости и невозможно объявлять или настраивать клиентские устройства в экземпляре протокола MODBUS Server. Все клиентские устройства будут иметь доступ ко всем сопоставлениям, выполненным непосредственно в экземпляре протокола MODBUS Server.

## 5. КОНФИГУРАЦИЯ

Запросы по устройствам: количество запросов, таких как чтение или запись регистров временного хранения, которые могут быть подтверждены для каждого из устройств (подчиненных или серверов) из протоколов Мастера или Клиента. Этот параметр не применяется к экземплярам протоколов Slave или Server.

Одновременных запросов на экземпляр: количество запросов, которые могут быть одновременно переданы каждым экземпляром протокола клиента или которые могут быть получены одновременно каждым экземпляром протокола сервера. Экземпляры протокола MODBUS RTU, Master или Slave, не поддерживают одновременные запросы.

Одновременных запросов на устройство: Количество запросов, которые могут быть одновременно переданы для каждого серверного устройства MODBUS или могут быть получены одновременно от каждого клиентского устройства MODBUS. Устройства MODBUS RTU, Master или Slave, не поддерживают одновременные запросы.

Ограничения сервера протокола IEC 60870-5-104 можно посмотреть в Таблице ниже.

### ВНИМАНИЕ

Не поддерживаются одновременные запросы к переменной, привязанной к точкам связи, поддерживающим режим работы SBO (Выбор перед исполнением), даже поступающие с разных устройств. После начала выбора/управления точкой определенным устройством это должно быть завершено, прежде чем эта точка сможет управляться другим устройством.

Ограничения	Сервер IEC 60870-5-104
Устройств на экземпляр	3
Одновременных запросов на экземпляр	3
Одновременные запросы на устройство	1

Таблица 63: Протокол IEC 60870-5-104 Пределы сервера

#### Примечания:

**Устройств на экземпляр:** Количество клиентских устройств типа центра управления, поддерживаемое для каждого экземпляра протокола сервера IEC 60870-5-104. Сообщенный лимит может быть меньше из-за общих лимитов ЦП (см. Таблицу 5.7).

Одновременных запросов на экземпляр: количество запросов, которые могут быть получены одновременно каждым экземпляром протокола сервера.

Одновременных запросов на устройство: Количество запросов, которые могут быть получены одновременно от каждого клиентского устройства IEC 60870-5-104.

#### 5.7.1. Поведение протокола в состоянии ЦП

В таблице ниже подробно показано поведение каждого настраиваемого протокола в ЦП серии Nexto в каждом рабочем состоянии.

Протокол	Тип	CPU operational state					
		СТО П			ЗАП УСК		
		После загрузки, перед запуском приложения	После того, как приложение переходит в СТОП (ПАУЗА)	После исключения	Не резервный или активный	Резервный в режиме ожидания	После точки останова в MainPrg
Символ MODBUS	Ведомый/Сервер	✓	✓	✓	✓	✓	✓
	Мастер/Клиент	✗	✗	✗	✓	✓	✓
MODBUS	Ведомый/Сервер	✗	✗	✗	✓	✓	✗
	Мастер/Клиент	✗	✗	✗	✓	✓	✓
SOE (DNP3)	дальняя станция	✓	✓	✓	✓	✗	✓
IEC 60870-5-104	Сервер	✗	✗	✗	✓	✗	✓

## 5. КОНФИГУРАЦИЯ

<b>EtherCAT</b>	Владелец	✓	✓	✗	✓	NA	✓
<b>OPC DA</b>	Сервер	✓	✓	✓	✓	✗	✓
<b>OPC UA</b>	Сервер	✓	✓	✓	✓	✓	✓
<b>SNTP</b>	Клиент	✓	✓	✓	✓	✓	✓
<b>HTTP</b>	Сервер	✓	✓	✓	✓	✓	✓
<b>SNMP</b>	Агент	✓	✓	✓	✓	✓	✓
<b>EtherNet/IP</b>	Сканер	✓	✓	✗	✓	NA	✗
	Адаптер	✗	✓	✗	✓	NA	✗

Таблица 64: Поведение протокола x состояние ЦП

### Примечания:

**Символ ✓:** Протокол остается активным и работает нормально.

**Символ ✗:** Протокол отключен.

**EtherCAT:** Тесты проводились с использованием значения По умолчанию, определенного в настройках ПЛК (см. Таблицу настроек ПЛК), с опцией

Обновить вход/выход во время остановки и опция Настроить все выходы на По умолчанию.

Ведомый/сервер символа MODBUS: чтобы поддерживать связь по протоколу, когда ЦП не находится в режиме RUN или после точки останова, необходимо установить флажок «Поддерживать связь при остановке ЦП».

### 5.7.2. Очередь событий процессора

ЦП владеет очередью событий типа FIFO (First In, First Out), используемой для временного хранения событий, связанных с точками связи, до тех пор, пока они не будут перемещены в свое окончательное место назначения.

Все события точек связи, сгенерированные в ЦП, направляются и сохраняются в очереди ЦП. Эта очередь имеет следующие особенности:

- Размер: 1000 событий
- Сохраняемость: не сохраняется
- Политика переполнения: сохраняйте самые новые

#### ВНИМАНИЕ

В ПЛК Nexto очередь событий хранится в энергонезависимой области памяти. Таким образом, события, присутствующие в очереди ЦП, которые еще не были переданы в центр управления, будут потеряны при возможном отключении питания ЦП.

Очередь событий ЦП избыточна, это означает, что она синхронизируется каждый цикл между обоими ЦП, когда используется избыточность ЦП. Дополнительную информацию можно найти в разделе о резервировании ЦП.

Входящие и исходящие события в этой очереди соответствуют концепции производителя/потребителя. Производители — это те системные элементы, которые способны генерировать события, добавляя события в очередь ЦП, а потребители — это те системные элементы, которые получают и используют эти события, забирая их из очереди ЦП. Рисунок ниже описывает эту работу, включая пример некоторых потребителей и производителей событий.

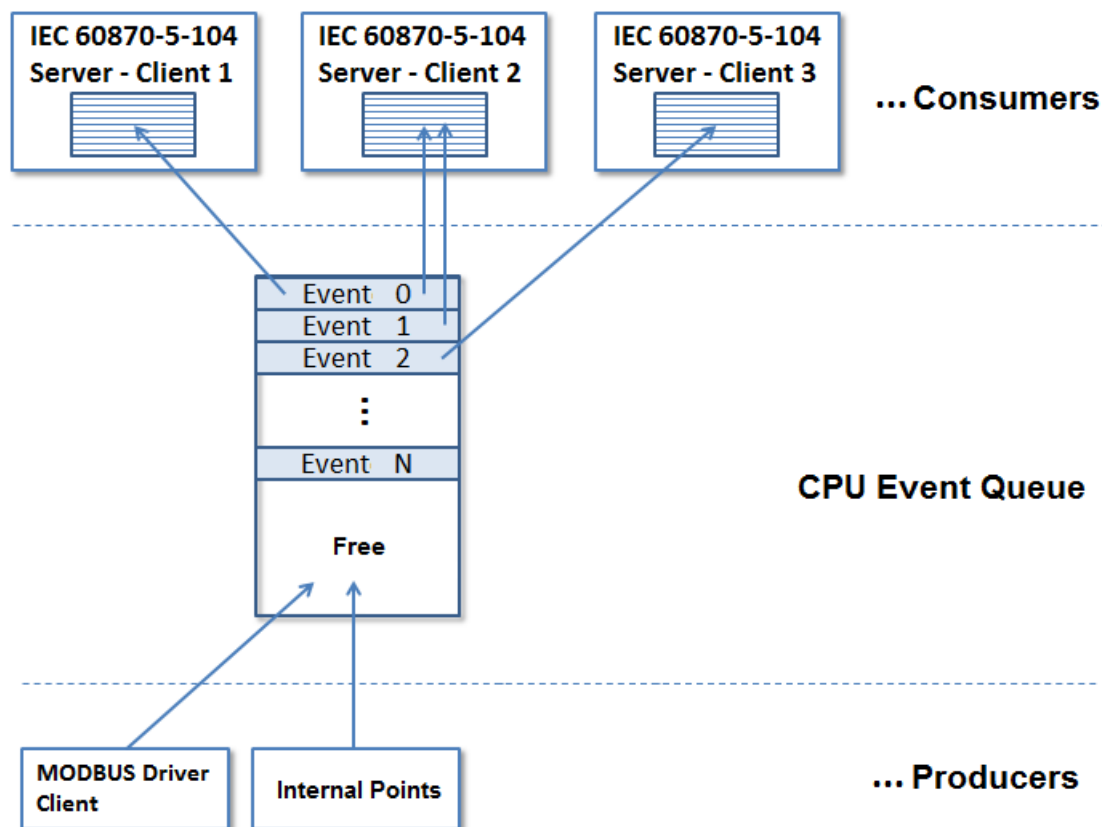


Рисунок 62: Очередь событий процессора

#### 5.7.2.1. Потребители

Потребители обычно являются коммуникационными драйверами, которые взаимодействуют со SCADA или HMI. После сохранения в очереди ЦП потребители получают события, связанные с точками связи, отображаемыми в его Конфигурации. Затем эти события сохраняются в собственной очереди событий потребителя, размер и работа которой описаны в разделе, посвященном конкретному коммуникационному драйверу.

#### 5.7.2.2. Принципы работы очереди

После сохранения в очереди ЦП каждое событие передается потребителю, в базе данных которого есть эта точка связи. На рисунке выше событие 0 относится к точке связи, сопоставленной с двумя центрами управления IEC 60870-5-104 (клиент 1 и 2). Таким образом, Событие 1 относится к точке связи, сопоставленной только с одним центром управления IEC 60870-5-104 (Клиент 2). По своему времени Событие 2 относится к точке связи, сопоставленной с другим центром управления IEC 60870-5-104 (Клиент 3).

События остаются в очереди ЦП до тех пор, пока все его потребители не подтвердят их получение. Критерии, используемые для подтверждения получения, являются специфическими для каждого потребителя. В случае сервера МЭК 60870-5-104 подтверждение происходит, когда событие передается клиенту МЭК 60870-5-104.

В случае с Nexto Series нет доступных диагностических средств для наблюдения за занятостью очереди событий ЦП, нет даже информации о переполнении очереди. Однако у потребителей есть группа диагностики, ссылающаяся на свою очередь событий. Дополнительную информацию можно найти в разделе связи с конкретным водителем.

##### 5.7.2.2.1. Знак переполнения

Признак переполнения очереди событий потребителей возникает в двух случаях:

Когда в очереди потребительских событий не хватает места для хранения новых событий

Если ЦП прервал генерацию события (поскольку в одном цикле выполнения произошло больше событий, чем общий размер очереди событий)

#### 5.7.2.3. Производители

Производители обычно являются коммуникационными драйверами или внутренними элементами ПЛК, способными генерировать события. На предыдущем рисунке показано несколько примеров.

Внутренние точки: это внутренний элемент встроенного программного обеспечения ПЛК, который обнаруживает события в каждом цикле выполнения (MainTask) для тех точек связи, которые не имеют определенного источника, а затем вставляет события в очередь ЦП. Максимальное количество событий, которые могут быть обнаружены в каждом цикле MainTask, равно размеру очереди событий ЦП. В случае, если количество сгенерированных событий больше, чем это, в одном цикле превышение будет потеряно.

Драйвер MODBUS (Клиент/Сервер/Главный/Подчиненный): Изменение значения переменных, вызванное чтением/записью MODBUS, обнаруживается в каждом цикле MainTask, а затем события вставляются в очередь ЦП. В случаях клиент/мастер также генерируются события качества, когда происходит сбой связи с ведомым устройством.

### **5.7.3. Перехват команд, поступающих из Центра управления**

ПЛК Nexто имеет функциональный блок, который позволяет командам выбора и операциям с выходными точками, полученными драйверами сервера (сервер IEC 60870-5-104), обрабатываться пользовательской логикой. Этот ресурс позволяет реализовать блокировку, а также обработку полученных командных данных в пользовательской логике или, тем не менее, перенаправление команд на разные устройства IED.

Перехват команд реализован функциональным блоком `CommandReceiver`, определенным в `LibRtuStandard`. Входные и выходные параметры описаны в следующих таблицах:

Параметр	Тип	Описание
<b>bExec</b>	BOOL	Когда TRUE, выполняет перехват команды
<b>bDone</b>	BOOL	Указывает, что выходные данные команды уже обработаны, освобождая функциональный блок для приема другой команды.
<b>dwVariableAddr</b>	DWORD	Переменный адрес, отображаемый в драйвере сервера, который будет получать клиентскую команду
<b>eCommandResult</b>	ENUM	Действие ввода, определяемое пользователем из следующего списка: SUCCESS(0) NOT_SUPPORTED(1) BLOCKED_BY_SWITCHING_HIERARCHY(2) SELECT_FAILED(3) INVALID_POSITION(4) POSITION_REACHED(5) PARAMETER_CHANGE_IN_EXECUTION(6) ) STEP_LIMIT(7) BLOCKED_BY_MODE(8) BLOCKED_BY_PROCESS(9) BLOCKED_BY_INTERLOCKING(10) BLOCKED_BY_SYNCHRONIZATION(11) COMMAND_ALREADY_IN_EXECUTION(12) BLOCKED_BY_HEALTH (13) ONE_OF_N_CONTROL(14) ABORTION_BY_CANCEL(15) TIME_LIMIT_OVER(16) ABORTION_BY_TRIP(17) OBJECT_NOT_SELECTED(18) OBJECT_ALREADY_SELECTED(19) NO_ACCESS_AUTHORITY(20) ENDED_WITH_OVERSHOOT(21) ABORTION_DUE_TO_DEVIATION(22) ABORTION_BY_COMMUNICATION_LOSS(23) BLOCKED_BY_COMMAND(24) NONE(25) INCONSISTENT_PARAMETERS(26) LOCKED_BY_OTHER_CLIENT(27) HARDWARE_ERROR(28) UNKNOWN(29)
<b>dwTimeout</b>	DWORD	Время ожидания [мс] до обработки по пользовательской логике

Таблица 65: Входные параметры функционального блока CommandReceiver

**Примечания:**

**bExec:** При значении FALSE команда просто перестает перехватываться пользовательским приложением, но сервер продолжает нормально обрабатывать ее.

**bDone:** После перехвата команды пользователь будет нести ответственность за ее обработку. В конце лечения этот вход должен быть активирован, чтобы можно было получить новую команду. Если этот вход не включен, блок будет ждать время, определенное в dwTimeout, чтобы затем стать способным перехватывать новые команды.

**eCommandResult:** результаты обработки команды, перехваченной пользователем. Результат, возвращаемый клиенту, отправившему команду, который необходимо атрибутировать вместе с вводом bDone, преобразуется в формат протокола, от которого была получена команда. В серии Nexto поддерживается только перехват команд, исходящих из протокола IEC 60870-5-104. При перехвате протокола любой результат, отличный от SUCCESS, приводит к отрицательному подтверждению.

**ВНИМАНИЕ**

Не рекомендуется одновременный перехват команд одной и той же переменной двумя или более функциональными блоками CommandReceiver. Только один из функциональных блоков будет правильно перехватывать команду, имея возможность столкнуться с нежелательными помехами со стороны других функциональных блоков, если они адресованы той же переменной..

Параметр	Тип	Описание
<b>bCommandAvailable</b>	BOOL	Указывает, что команда была перехвачена и данные доступны для обработки
<b>sCommand</b>	STRUCT	В этой структуре хранятся полученные данные команды, которые состоят из следующих полей: eCommand sSelectParameters sOperateParameters Описание каждого поля находится в этом разделе
<b>eStatus</b>	ENUM (ТИП_RESULT)	Действие вне функции от полученного результата согласно списку: OK_SUCCESS(0) ERROR_FAILED(1)

Таблица 66: Выходные параметры функционального блока CommandReceiver

**Примечание:**

**eStatus:** Возврат процесса регистрации перехвата команды точки связи. Когда перехват успешно зарегистрирован, возвращается OK\_SUCCESS, в противном случае возвращается ERROR\_FAILED. В случае сбоя регистра-перехватчика команды в определенную точку не перехватываются этим функциональным блоком. ТИП\_RESULT определен в библиотеке LibDataTypes.

Поддерживаемые команды описаны на Таблице ниже:

Параметр	Тип	Описание
<b>eCommand</b>	ENUM	NO_COMMAND(0) SELECT(1) OPERATE(2) )

Таблица 67: Команды, поддерживаемые функциональным блоком CommandReceiver

Параметры, из которых строятся структуры Select Parameters, Operate Parameters и Cancel Parameters, описаны в следующей Таблице:

Параметр	Тип	Описание
<b>sSelectConfig</b>	STRUCT	Получена команда выбора Конфигурация. Параметры этой структуры описаны в Таблице 69.
<b>sValue</b>	STRUCT	Полученное значение при выборе, когда получена команда выбора со значением. Параметры этой структуры описаны в Таблице 72.

Таблица 68: выберите параметры

Параметр	Тип	Описание
<b>bSelectWithValue</b>	BOOL	Значение true указывает на прием команды выбора со значением.

Таблица 69: Параметры sSelectConfig

Параметр	Тип	Описание
<b>sOperateConfig</b>	STRUCT	Получена команда выбора конфигурации. Параметры этой структуры описаны в Таблице 71.
<b>sValue</b>	STRUCT	Поле принятого значения команды операции. Параметры этой структуры описаны в Таблице 72.

Таблица 70: Параметры sOperateParameters

Параметр	Тип	Описание
<b>bDirectOperate</b>	BOOL	Значение true указывает, что была получена рабочая команда без выбора.
<b>bNoAcknowledgement</b>	BOOL	Значение true указывает, что была получена команда, не требующая подтверждения приема.
<b>bTimedOperate</b>	BOOL	Значение true указывает на то, что была получена рабочая команда, активированная по времени.
<b>liOperateTime</b>	LINT	Программирование момента, в который должна быть запущена команда. Это поле допустимо, только если bTimedOperate имеет значение true.
<b>bTest</b>	BOOL	Значение true указывает, что полученная команда была отправлена только для проверки, поэтому команду запускать нельзя.

Таблица 71: Параметры sOperateConfig

Параметр	Тип	Описание
<b>eParamТип</b>	ENUM	Сообщает тип полученной команды: NO_COMMAND(0) SINGLE_POINT_COMMAND(1) DOUBLE_POINT_COMMAND(2) INTEGER_STATUS_COMMAND(3) ENUMERATED_STATUS_COMMAND(4) ) ANALOGUE_VALUE_COMMAND(5)
<b>sSinglePoint</b>	STRUCT	Когда команда получена, в функции типа принятой команды, определенной eParamТип, заполняется соответствующая структура данных. Параметры этой структуры описаны в таблицах с 73 по 77.
<b>sDoublePoint</b>	STRUCT	
<b>sIntegerStatus</b>	STRUCT	
<b>sEnumeratedStatus</b>	STRUCT	
<b>sAnalogueValue</b>	STRUCT	

Таблица 72: Параметры sValue

Параметр	Тип	Описание
bValue	BOOL	Значение операции точки.
sPulseConfig	STRUCT	В этой структуре хранятся параметры импульсной команды Конфигурация. Параметры этой структуры описаны в Таблице 78.

Таблица 73: Параметры sSinglePoint

Параметр	Тип	Описание
bValue	BOOL	Значение операции точки.
sPulseConfig	STRUCT	В этой структуре хранятся параметры конфигурации импульсной команды. Параметры этой структуры описаны в Таблице 78.

Таблица 74: Параметры sDoublePoint

Параметр	Тип	Описание
diValue	DINT	Значение операции точки.

Таблица 75: Параметры sIntegerStatus

Параметр	Тип	Описание
dwValue	DWORD	Значение операции точки.

Таблица 76: Параметры sEnumeratedStatus

Параметр	Тип	Описание
eТип	ENUM	Сообщает тип данных полученного аналогового значения . INTEGER (0) FLOAT (1)
diValue	DINT	Значение операции точки, целочисленный формат.
fValue	REAL	Значение операции точки, формат с плавающей запятой.

Таблица 77: Параметры sAnalogueValue

Параметр	Тип	Описание
bPulseCommand	BOOL	Значение true указывает, что полученная команда является импульсной.
dwOnDuration	DWORD	Это время в миллисекундах, в течение которого выход должен оставаться включенным.
dwOffDuration	DWORD	Это время в миллисекундах, в течение которого выход должен оставаться выключенным.
dwPulseCount	DWORD	Количество раз, которое должна быть выполнена команда.

Таблица 78: Параметры sPulseConfig

Для перехвата команд в конкретную соответствующий точке, в которую необходимо перехватить команды, а затем выполнить импульс в параметре bExec. Как только команда была перехвачена, функциональный блок сообщает, что команда была перехвачена через параметр bCommandAvailable. Информация о перехваченной команде затем заносится в выходные параметры sCommand и eStatus в соответствии с типом полученной команды. Эта операция зависит только от типа полученной команды, не имеет значения тип данных переменной, в которую перехватывается команда. Перехват завершен, и тогда функциональный блок может быть освобожден для перехвата новой команды, когда параметр bDone имеет значение true. Еще необходимо указать результат обработки команды в eCommandResult.

#### 5.7.4. Ведущее устройство MODBUS RTU

Этот протокол доступен для ЦП серии Nexto в его последовательных каналах. При выборе этой опции в MasterTool ИЕС ХЕ ЦП становится ведущим устройством связи MODBUS, обеспечивая доступ к другим устройствам с тем же

## 5. КОНФИГУРАЦИЯ

протоколом, когда он находится в режиме выполнения (режиме выполнения).

Для этого протокола существует два режима Конфигурация. Один использует прямое представление (%Q), в котором переменные определяются своим адресом. Другой, называемый Symbolic Mapping, имеет переменные, определенные его именем.

Вне зависимости от режима Конфигурация шаги по вставке экземпляра протокола и подключению к последовательному интерфейсу одинаковы. Процедура вставки экземпляра протокола подробно описана в Руководстве пользователя MasterTool IEC XE — MU299609 или в разделе «Вставка экземпляра протокола». Остальные этапы Конфигурации описаны ниже для каждого режима.

Добавьте экземпляр основного протокола MODBUS RTU в последовательный канал COM 1 или COM 2 (или оба, в случае двух сетей связи). Чтобы выполнить эту процедуру, см. раздел «Вставка экземпляра протокола».

Настройте последовательный интерфейс, выберите скорость передачи, поведение сигналов RTS/CTS, четность, стоповые биты канала, среди прочего Конфигурация двойным щелчком на последовательном канале COM 1 или COM 2. См. раздел Конфигурация последовательных интерфейсов.

### 5.7.4.1. Конфигурация главного протокола MODBUS с помощью символьного отображения

Чтобы подтвердить этот протокол с помощью символьного отображения, необходимо выполнить следующие шаги:

Подпишите общие параметры протокола MODBUS Master, такие как: время задержки передачи и минимальный интервал между кадрами, как показано на рисунке 63.

Добавьте и подключите устройства через вкладку «Общие параметры», указав адрес подчиненного устройства, время ожидания связи и количество повторных попыток связи, как показано на рисунке 64.

Добавьте и подтвердите отображение MODBUS на вкладке Mappings как Рисунок 65, указав имя переменной, тип данных и начальный адрес данных, размер данных и диапазон заполняются автоматически.

Добавьте и подтвердите запросы MODBUS, как показано на Рисунке 66, указав функцию, время сканирования запроса, начальный адрес (чтение/запись), размер данных (чтение/запись), сгенерируйте диагностические переменные и отключите запрос через кнопки в нижней части окна.

#### 5.7.4.1.1. Основные параметры основного протокола MODBUS — символическое отображение Конфигурация

Общие параметры, находящиеся на начальном экране протокола MODBUS (Рисунок ниже), определяются как: точку сначала необходимо загрузить в параметр dwVariableAddr адрес переменной,

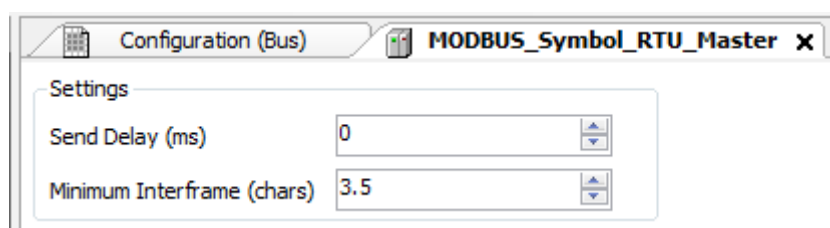


Рисунок 63: Экран конфигурации главного устройства MODBUS RTU

Конфигурация	Описание	По умолчанию	Варианты
Задержка отправки (мс)	Задержка передачи ответа.	0	0 to 65535
Минимальный межкадровый интервал (символы)	Минимальное время молчания между разными кадрами.	3.5	3.5 to 100.0

Таблица 79: Общие конфигурации MODBUS RTU Master

#### Примечания:

**Задержка отправки:** Ответ на протокол MODBUS может вызвать проблемы в определенные моменты, например, в интерфейсе RS-485 или другом полудуплексе. Иногда существует задержка между временем ответа подчиненного устройства и тишиной на физической линии (задержка подчиненного устройства для обнуления RTS и перевода RS-485 в состояние высокого импеданса). Чтобы решить эту проблему, мастер может подождать определенное время в этом поле перед отправкой нового запроса. В противном случае первые байты, переданные мастером, могут быть потеряны.

Минимальный межкадровый интервал: стандарт MODBUS определяет это время как 3,5 символа, но этот параметр настраивается для обслуживания устройств, которые не соответствуют стандарту.

Диагностика протокола MODBUS и команды conRiseунокd, либо символическим отображением, либо прямым представлением, хранятся в переменных T\_DIAG\_MODBUS\_RTU\_MASTER\_1. Для отображения прямого представления они также состоят из 4 байтов и 8 слов, которые описаны в Таблице ниже (где «n» — значение configured в поле %Q Начальный адрес области диагностики):

## 5. КОНФИГУРАЦИЯ

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS _RTU_MASTER_1.*	Размер	Описание
<b>Биты диагностики:</b>			
%QX(n).0	tDiag. bRunning	BIT	Запуск Мастера.
%QX(n).1	tDiag. bNotRunning	BIT	Мастер не работает (см. бит:bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, так как мастер был прерван пользователем через командные биты.
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращенный диагноз.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращенный диагноз.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращенный диагноз.
%QX(n).6	tDiag. bModuleFailure	BIT	Указывает на неисправность модуля или отсутствие модуля.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Сдержанный
<b>Коды ошибки:</b>			

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS _RTU_MASTER_1.*	Размер	Описание
%QB(n+1)	eErrorCode	SERIAL_STATU S (BYTE)	0: ошибок нет. 1: неверный последовательный порт. 2: неверный режим последовательного порта 3: неверная скорость передачи данных 4: неверные биты данных 5: неверная четность 6: неверные стоповые биты 7: неверный параметр сигнала модема 8: неверный параметр порога приема UART 9: неверный параметр тайм-аута 10: последовательный порт занят 11: аппаратная ошибка UART 12: удаленная аппаратная ошибка 20: неверный размер буфера передачи 21: неверный сигнал модема 22: тайм-аут CTS = true 23: тайм-аут CTS = ложь 24: ошибка тайм-аута передачи 30: неверный размер буфера приема 31: ошибка тайм-аута приема 32: управление потоком выполняется иначе, чем вручную 33: недопустимое управление потоком для последовательного порта conRiseunokd 34: прием данных в обычном режиме запрещен 35: прием данных запрещен в расширенном режиме 36: Прерывание DCD не разрешено 37: Прерывание CTS не разрешено 38: Прерывание DSR не разрешено 39: Последовательный порт не разрешен 50: внутренняя ошибка в последовательном порту
<b>Командные биты, автоматически инициализируются:</b>			
%QX(n+2).0	tCommand. bStop	BIT	Остановка мастера.
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск мастера.
%QX(n+2).2	tCommand. bResetCounter	BIT	Перезапустите статистику диагностики (счетчики).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Зарезервировано
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Зарезервировано
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Зарезервировано
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Зарезервировано

## 5. КОНФИГУРАЦИЯ

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS S_RTU_MASTER_1.*	Размер	Описание
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Зарезервировано
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано
<b>Статистика связи:</b>			
%QW(n+4)	tStat. wTXRequests	WORD	Счетчик запросов, переданных мастером (от 0 до 65535).
%QW(n+6)	tStat. wRXNormalResponses	WORD	Счетчик нормальных ответов, полученных мастером (от 0 до 65535).
%QW(n+8)	tStat. wRXExceptionResponses	WORD	Счетчик ответов с кодами исключений, полученных мастером (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalResponses	WORD	Счетчик недопустимых ответов, полученных мастером — неверный синтаксис, недостаточно полученных байтов, неверный CRC — (от 0 до 65535).
%QW(n+12)	tStat. wRXOverrunErrors	WORD	Счетчик ошибок переполнения при приеме - UART FIFO или линия RX - (от 0 до 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	WORD	Счетчик ответов с ошибками построения, четности или отказа при приеме (от 0 до 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	WORD	Счетчик ошибок тайм-аута CTS при использовании квитирования RT-S/CTS во время передачи (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано

Таблица 80: Диагностика мастер MODBUS RTU

### Примечание:

**Счетчики:** Все счетчики диагностики главного устройства MODBUS RTU обнуляются при превышении предельного значения 65535.

#### 5.7.4.1.2. Конфигурация устройств — Конфигурация символического отображения

Конфигурация устройств, показанная на рисунке ниже, соответствует следующим параметрам:

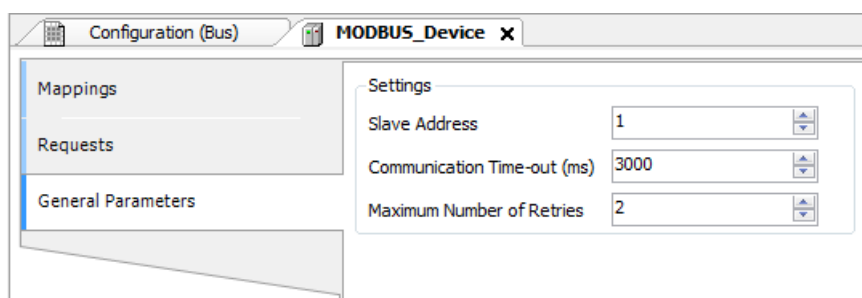


Рисунок 64: Настройки общих параметров устройства

Конфигурация	Описание	По умолчанию	Варианты
Адрес подчиненного устройства	Адрес подчиненного устройства MODBUS	1	0 до 255
Время ожидания связи (мс)	Определяет время ожидания уровня приложения	3000	10 до 65535
Максимальное количество попыток	Определяет количество повторных попыток перед сообщением об ошибке связи	2	0 до 9

Таблица 81: Конфигурация устройств

**Примечания**

**Адрес подчиненного устройства:** В соответствии со стандартом MODBUS действительные адреса ведомых устройств от 0 до 247, где адреса от 248 до 255 зарезервированы. Когда мастер отправляет команду записи с нулевым адресом, он выполняет широковещательные запросы в сети.

**Тайм-аут связи:** Тайм-аут связи — это время, в течение которого мастер ожидает ответа от ведомого на запрос. Для ведущего устройства MODBUS RTU необходимо учитывать как минимум следующие системные переменные: время, необходимое ведомому для передачи кадра (согласно скорости передачи), время, необходимое ведомому для обработки запроса и отправки ответа, задержка, если она настроена в ведомом устройстве. Рекомендуется, чтобы время ожидания было равно или больше, чем время на передачу кадра плюс задержка отправки ответа и удвоенное время обработки запроса. Дополнительные сведения см. в разделе «Производительность связи».

**Максимальное количество повторных попыток:** устанавливает количество повторных попыток до сообщения об ошибке связи. Например, если ведомое устройство не отвечает на запрос, а ведущее устройство настроено на отправку трех повторных попыток, номер счетчика ошибок увеличивается на единицу при выполнении этих трех повторных попыток. После увеличения ошибки связи попытка перезапустить процесс и, если число повторных попыток будет достигнуто снова, новая ошибка увеличит счетчик.

## 5.7.4.1.3. Конфигурация сопоставлений – Настройки символического сопоставления

Отношения MODBUS Конфигурация, показанная на рисунке ниже, соответствует параметрам, описанным в таблице ниже:

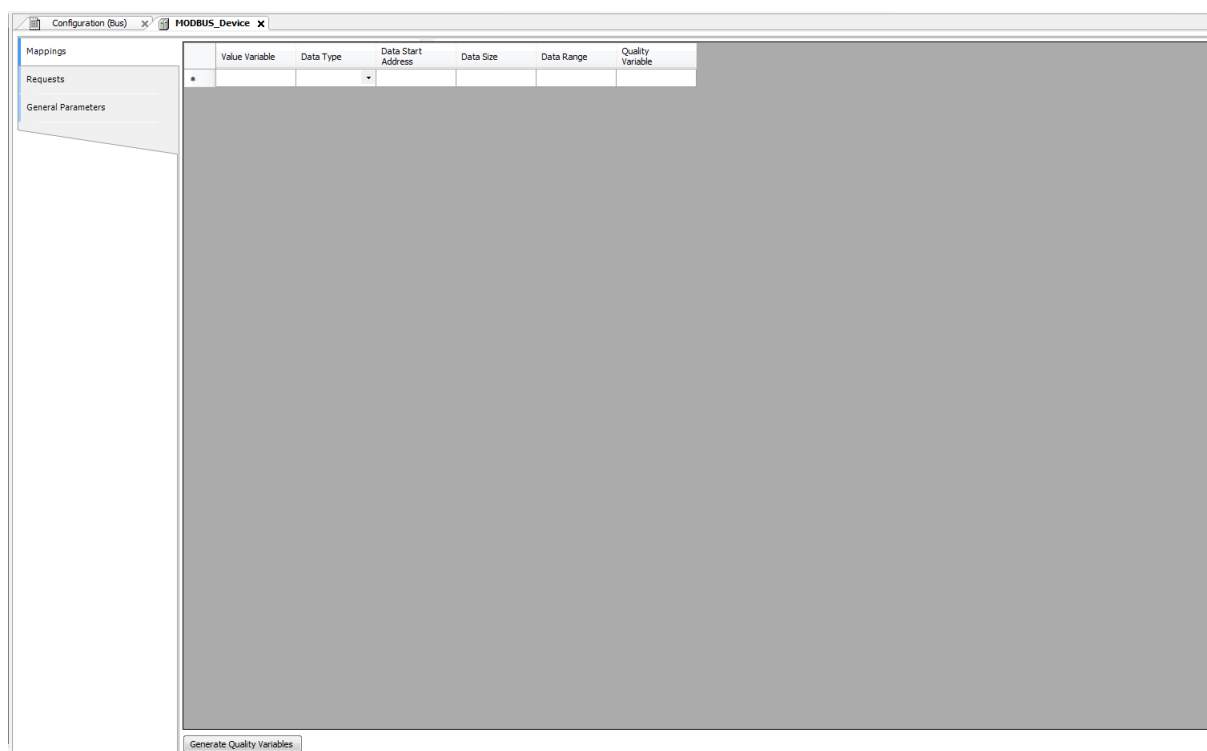


Рисунок 65: Экран отображения данных MODBUS

Конфигурация	Описание	По	Варианты
--------------	----------	----	----------

## 5. КОНФИГУРАЦИЯ

		умолча нию	
<b>Переменная значений</b>	Символьное наименование переменной	-	Имя переменной, объявленной в программе или GVL
<b>Тип данных</b>	Тип данных MODBUS	-	Катушка — запись (1 бит) Катушка — чтение (1 бит) Регистр хранения — запись (16 бит) Регистр хранения - чтение (16 бит) Регистр временного хранения – Маска И (16 бит) Регистр временного хранения — Маска ИЛИ (16 бит) Входной регистр (16 бит) Состояние входа (1 бит)
<b>Начальный адрес данных</b>	Начальный адрес данных MODBUS	-	1 до 65536
<b>Размер данных</b>	Размер данных MODBUS	-	1 до 65536
<b>Диапазон данных</b>	Диапазон адресов настроенных данных	-	-

Таблица 82: Настройки отображения MODBUS

### Примечания:

**Переменная значения:** это поле используется для указания символической переменной в отношении MODBUS.

**Тип данных:** это поле используется для указания типа данных, используемых в отношении MODBUS.

Тип данных	Размер [биты]	Описание
Катушка – для записи	1	Запись цифрового выхода.
Катушка – для чтения	1	Чтение цифрового выхода.
Проведение реестра - запись	16	Запись аналогового выхода.
Холдинговый регистр - чтение	16	Чтение аналогового выхода.
Холдинговый регистр – Маска И	16	Аналоговый выход, который может быть прочитан или записан с маской И.
Холдинговый регистр — маска ИЛИ	16	Аналоговый выход, который может быть прочитан или записан с маской ИЛИ.
Входной регистр	16	Аналоговый вход, который можно только читать.
Статус ввода	1	Цифровой вход, который можно только прочитать.

Таблица 83: Типы данных, поддерживаемые в MODBUS

**Начальный адрес данных:** начальный адрес данных отображения MODBUS.

**Размер данных:** значение размера определяет максимальный объем данных, к которым может получить доступ интерфейс MODBUS, начиная с начального адреса. Таким образом, чтобы прочитать Продолжающийся диапазон адресов, необходимо, чтобы все адреса были объявлены в одном интерфейсе. Это поле зависит от типа данных MODBUS.

**Диапазон данных:** это поле показывает пользователю диапазон адресов памяти, используемый интерфейсом MODBUS.

#### 5.7.4.1.4. Конфигурация запросов — Настройки символического сопоставления

Конфигурация запросов MODBUS, показанная на рисунке ниже, следуйте параметрам, описанным в Таблице ниже:

## 5. КОНФИГУРАЦИЯ

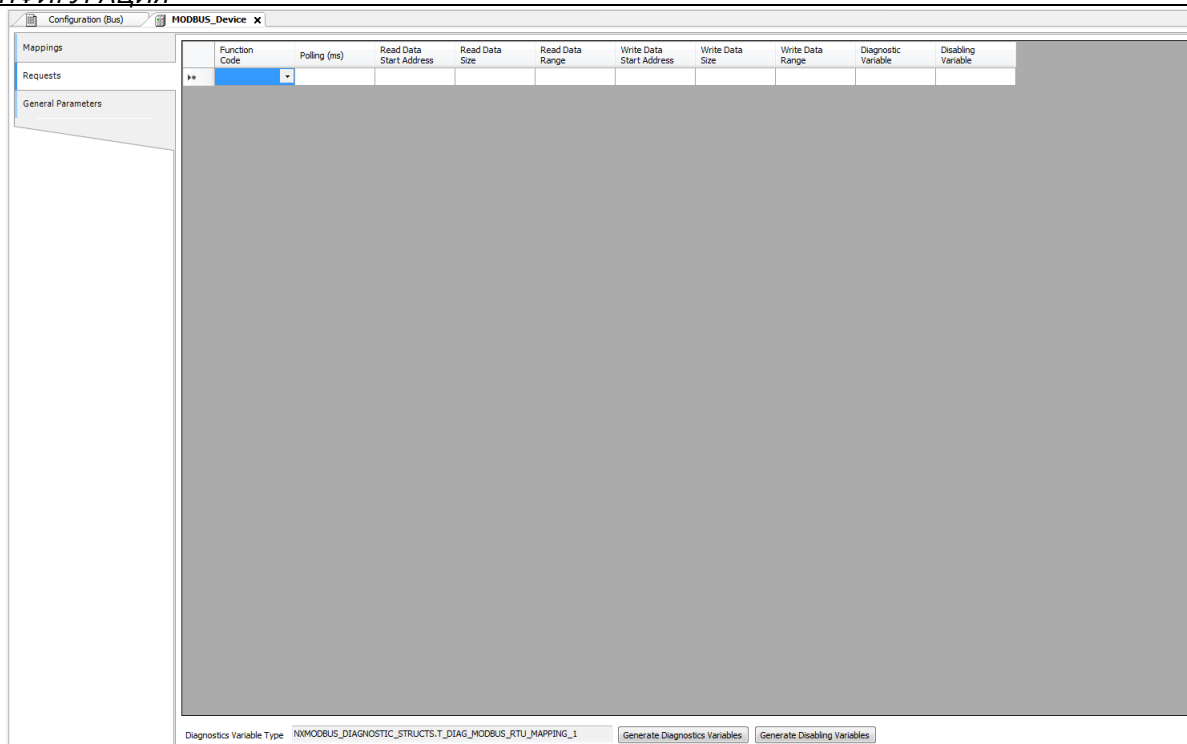


Рисунок 66: Экран запроса данных Ведущее устройство MODBUS

Конфигурация	Описание	Значение по умолчанию	Варианты
Функциональный код	Тип функций MODBUS	-	01 — Чтение катушек 02 — Чтение состояния входа 03 — Чтение регистров хранения 04 — Чтение входных регистров 05 — Написать одиночную катушку 06 — Запись одного регистра 15 - Написать несколько катушек 16 — Запись нескольких регистров 22 - Регистр записи маски 23 — Чтение/запись нескольких регистров
Опрос (мс)	Период связи (мс)	100	0 до 3600000
Чтение начального адреса данных	Начальный адрес считанных данных MODBUS	-	1 до 65536
Чтение размера данных	Размер считанных данных MODBUS	-	Зависит от используемой функции
Чтение диапазона данных	Диапазон адресов данных MODBUS для чтения	-	0 до 2147483646
Запись начального адреса данных	Начальный адрес данных записи MODBUS	-	1 до 65536
Запись размера данных	Размер данных списания MODBUS	-	Зависит от используемой функции
Запись диапазона данных	Диапазон адресов данных MODBUS для записи	-	0 до 2147483647

Конфигурация	Описание	Значение по умолчанию	Варианты
Диагностическая переменная	Наименование диагностической переменной	-	Имя переменной, объявленной в программе или GVL
Отключение переменной	Переменная, используемая для отключения связи MODBUS	-	Поле для символической переменной, используемой для индивидуального отключения настроенных запросов MODBUS. Эта переменная должна быть типа BOOL. Переменная может быть простой или элементом массива и может находиться в структурах.

Таблица 84: Конфигурация отношений MODBUS

**Примечания:**

**Настройки:** количество заводских настроек по умолчанию и значения столбца Варианты могут различаться в зависимости от типа данных и функции MODBUS (FC).

Код функции: Доступны следующие функции MODBUS (FC):

Код		Описание
DEC	HEX	
1	0x01	Читать катушки (FC 01)
2	0x02	Чтение состояния входа (FC 02)
3	0x03	Чтение регистров хранения (FC 03)
4	0x04	Чтение входных регистров (FC 04)
5	0x05	Запись одиночной катушки (FC 05)
6	0x06	Запись единого регистра временного хранения (FC 06)
15	0x0F	Запись нескольких катушек (FC 15)
16	0x10	Запись нескольких регистров временного хранения (FC 16)
22	0x16	Регистр хранения записи по маске (FC 22)
23	0x17	Чтение/запись нескольких регистров временного хранения (FC 23)

Таблица 85: Функции MODBUS, поддерживаемые процессорами Nexto

**Опрос:** этот параметр указывает, как часто должна выполняться связь, установленная для этого запроса. По окончании связи будет ожидаться время, равное значению configured в поле polling, после чего будет выполнена новая коммуникация. Начальный адрес чтения данных: поле для начального адреса считанных данных MODBUS.

Размер прочитанных данных: минимальное значение размера прочитанных данных равно 1, а максимальное значение зависит от используемой функции MODBUS (FC), как показано ниже:

Читать катушки (FC 01): 2000

ReadInputStatus (FC 02): 2000

Чтение регистров хранения (FC 03): 125

Чтение входных регистров (FC 04): 125

Чтение/запись нескольких регистров (FC 23): 121

Диапазон считанных данных: в этом поле отображается диапазон считанных данных MODBUS, указанный для каждого запроса. Начальный адрес вместе с размером прочитанных данных будет определять диапазон прочитанных данных для каждого запроса.

Начальный адрес записи данных: поле для начального адреса записи данных MODBUS.

Размер записываемых данных: минимальное значение размера записываемых данных равно 1, а максимальное значение зависит от используемой функции MODBUS (FC), как показано ниже:

Запись одиночной катушки (FC 05): 1

Запись одного регистра (FC 06): 1

Напишите несколько катушек (FC 15): 1968 г.

Запись нескольких регистров (FC 16): 123

Регистр записи по маске (FC 22): 1

## 5. КОНФИГУРАЦИЯ

Чтение/запись нескольких регистров (FC 23): 121

Диапазон записи данных: в этом поле отображается диапазон данных записи MODBUS, сконфигурированный для каждого запроса. Начальный адрес вместе с размером считанных данных приведет к диапазону записываемых данных для каждого запроса.

Диагностическая переменная: Диагностика запросов MODBUS, настроенная символическим отображением или прямым представлением, хранится в переменных типа T\_DIAG\_MODBUS\_RTU\_MAPPING\_1 для ведущих устройств и T\_DIAG\_MODBUS\_ETH\_CLIENT\_1 для клиентских устройств, а отображение путем прямого представления состоит из 4 байтов и 2 слов, которые описаны в Таблице 86 ("n" — значение, настроенное в поле %Q Начальный адрес области диагностики).

Переменная прямого представления	Диагностическая переменная типа T_DIAG_MODBUS_RTU_MAPPING_1.*	Размер	Описание
<b>Биты состояния связи:</b>			
%QX(n).0	byStatus. bCommIdle	BIT	Связь простаивает (ожидает выполнения).
%QX(n).1	byStatus. bCommExecuting	BIT	Активное общение.
%QX(n).2	byStatus. bCommPostponed	BIT	Связь задержана, так как достигнуто максимальное количество одновременных запросов. Отложенные коммуникации будут осуществляться в той же последовательности, в которой они были заказаны, чтобы избежать неопределенности. Время, проведенное в этом состоянии, не засчитывается для тайм-аута. Биты bCommIdle и bCommExecuting ложны, когда бит bCommPostponed истинен.
%QX(n).3	byStatus. bCommDisabled	BIT	Связь отключена. В этом состоянии бит bCommIdle перезапускается.
%QX(n).4	byStatus. bCommOk	BIT	Прерванная ранее связь прошла успешно.
%QX(n).5	byStatus. bCommError	BIT	Связь была прервана ранее из-за ошибки. Проверьте код ошибки.
%QX(n).6	byStatus. bCommAborted	BIT	Не используется в ведущем устройстве MODBUS RTU.
%QX(n).7	byStatus. bDiag_7_reserved	BIT	Сдержанный
<b>Код последней ошибки (включен, если bCommError = true):</b>			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Сообщает возможную причину последней ошибки в отображении MODBUS. Обратитесь к Таблице 109 для получения дополнительной информации.
<b>Последний код исключения, полученный мастером:</b>			
%QB(n+2)	eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*

## 5. КОНФИГУРАЦИЯ

Переменная прямого представления	Диагностическая переменная типа T_DIAG_MODBUS_RTU_MAPPING_1.*	Размер	Описание
Статистика общения:			
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.
%QW(n+4)	wCommCounter	WORD	Счетчик завершенных коммуникаций (с ошибками или без). Пользователь может проверить изменение этого счетчика после завершения связи. При достижении значения 65535 счетчик возвращается к нулю.
%QW(n+6)	wCommErrorCounter	WORD	Счетчик завершенных коммуникаций (с ошибками). При достижении значения 65535 счетчик возвращается к нулю.

Таблица 86: Диагностика отношений MODBUS

### Примечания:

**Коды исключений:** Коды исключений, представленные в этом поле, являются значениями, возвращаемыми ведомым устройством. Определения кодов исключений 128, 129 и 255, представленные в Таблице, действительны только при использовании ведомых устройств Altus. Ведомые устройства других производителей могут использовать другие определения для каждого кода.

**Отключающая переменная:** переменная логического типа, используемая для индивидуального отключения запросов MODBUS, отображаемых на вкладке запроса с помощью кнопки в нижней части окна. Запрос отключен, когда переменная, соответствующая запросу, равна 1, в противном случае запрос включен.

Код последней ошибки: Ниже приведены коды возможных ситуаций, вызывающих ошибку связи по протоколу MODBUS:

Код	Enumerable	Описание
1	ERR_EXCEPTION	Ответ находится в коде исключения (см. eLastExceptionCode= Код исключения).
2	ERR_CRC	Ответ с неверным CRC.
3	ERR_ADDRESS	Адрес MODBUS не найден. Адрес, ответивший на запрос, отличается от ожидаемого.
4	ERR_FUNCTION	Неверный код функции. Код функции ответа отличался от ожидаемого.
5	ERR_FRAME_DATA_COUNT	Объем данных в ответе отличается от ожидаемого.
7	ERR_NOT_ECHO	Ответ не является эхом запроса (FC 05 и 06).
8	ERR_REFERENCE_NUMBER	Неверный ссылочный номер (FC 15 и 16).
9	ERR_INVALID_FRAME_SIZE	Ответ короче, чем ожидалось.
20	ERR_CONNECTION	Ошибка при установлении соединения.
21	ERR_SEND	Ошибка на этапе передачи.
22	ERR_RECEIVE	Ошибка на этапе приема.
40	ERR_CONNECTION_TIMEOUT	Время ожидания на уровне приложения во время соединения.
41	ERR_SEND_TIMEOUT	Тайм-аут на уровне приложения во время передачи.
42	ERR_RECEIVE_TIMEOUT	Время ожидания на уровне приложения при ожидании ответа.
43	ERR_CTS_OFF_TIMEOUT	Тайм-аут при ожидании CTS = false при передаче.
44	ERR_CTS_ON_TIMEOUT	Тайм-аут при ожидании CTS = true при передаче.
128	NO_ERROR	Нет ошибок с момента запуска.

Таблица 87: Коды ошибок отношений MODBUS

**ВНИМАНИЕ**

В отличие от других прикладных задач, при достижении отметки очистки в MainTask задача ведущего экземпляра MODBUS RTU и любая другая задача MODBUS перестанут выполняться в тот момент, когда она попытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных областей памяти, когда основная задача не завершена.

**5.7.4.2. Конфигурация главного протокола MODBUS для прямого представления (%Q)**

Чтобы настроить этот протокол с использованием прямого представления (%Q), необходимо выполнить следующие шаги:

настроить общие параметры протокола MODBUS, такие как: время связи и переменные прямого представления (%Q) для получения диагностики.

Добавьте и настройте устройства, задав адрес, переменные прямого представления (%Q) для отключения отношений, тайм-ауты связи и т. д.

Добавьте и настройте отношения MODBUS, указав тип данных и функцию MODBUS, тайм-ауты, переменные прямого представления (%Q) для получения диагностики отношения и другие для приема/записи данных, объем данных для передачи и опрос отношения.

Описание каждой Конфигурации приведено ниже в этом разделе.

**5.7.4.2.1. Общие параметры ведущего протокола MODBUS - настройка прямым представлением (%Q)**

Общие параметры, находящиеся на главном экране конфигурации протокола MODBUS (Рисунок ниже), определяются как:

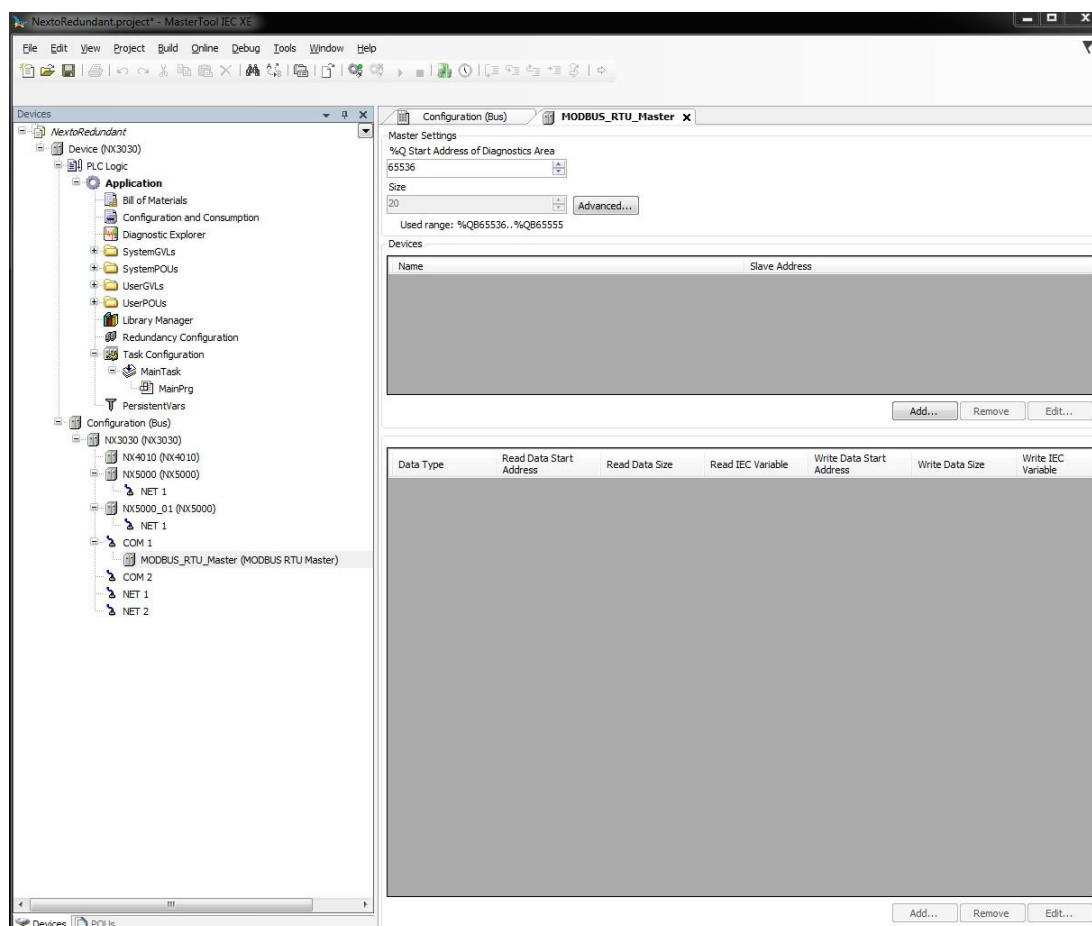


Рисунок 67: Экран настройки главного устройства MODBUS RTU

Переменные прямого представления (%Q) для диагностики протокола:

Конфигурация	Описание	Значение по умолчанию	Варианты
<b>%Q Начальный адрес области диагностики</b>	Начальный адрес диагностических переменных	-	0 до 2147483628
<b>Размер</b>	Размер области диагностики	20	Отключено для редактирования

Таблица 88: Конфигурация ведущего устройства MODBUS RTU

**Примечания:**

**Начальный адрес диагностики в %Q:** это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого ЦП, который можно найти в разделе Память.

Значение по умолчанию: заводское значение по умолчанию не может быть установлено в поле %Q Начальный адрес области диагностики, поскольку создание экземпляра протокола может быть приостановлено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), еще не использованных.

Диагностика и команды протокола MODBUS описаны в Таблице 80.

Время связи ведущего протокола MODBUS, указанное в кнопке Дополнительно... на экране Конфигурация, разделено на Задержку отправки и Минимальное межкадровое сообщение, более подробная информация описана в разделе Общие параметры ведущего протокола MODBUS — Конфигурация символического отображения.

#### 5.7.4.2.2. Конфигурация устройств — Конфигурация для прямого представительства (%Q)

Конфигурация устройств, показанная на рисунке ниже, включает следующие параметры:

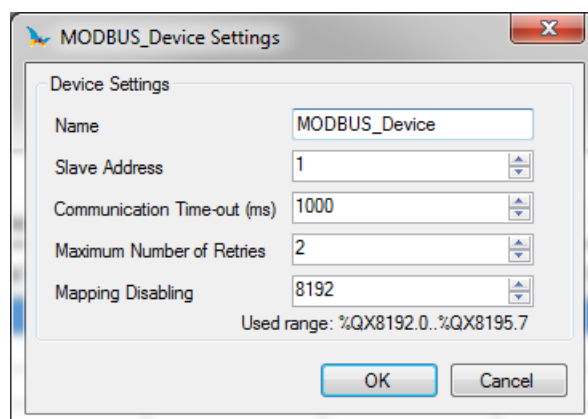


Рисунок 68: Конфигурация устройства

Конфигурация	Описание	Значение по умолчанию	Варианты
Наименование	Имя экземпляра	MODBUS_Device	Идентификатор согласно IEC 61131-3
Адрес подчиненного устройства	Адрес подчиненного устройства MODBUS	1	0 до 255
Время ожидания связи (мс)	Устанавливает тайм-аут уровня приложения	1000	10 до 65535
Максимальное количество попыток	Устанавливает количество повторных попыток	2	0 до 9
Отключение отображения	перед сообщением об ошибке связи	-	0 до 2147483644

Таблица 89: Конфигурация устройства - Ведущее устройство MODBUS

**Примечания:**

**Имя экземпляра:** это поле является идентификатором устройства, которое проверяется в соответствии с IEC 61131-3, т. е. не допускает пробелов, специальных символов и начинается с цифры. Он ограничен 24 символами.

Отключение сопоставления: состоит из 32 битов, используемых для индивидуального отключения 32 отношений MODBUS, настроенных в пространстве сопоставлений устройств. Отношение отключено, когда бит, соответствующий отношению, равен 1, в противном случае отображение включено. Это поле ограничено размером выходных переменных (%Q) адресуемой памяти каждого ЦП, который можно найти в разделе Память.

Значение по умолчанию: заводское значение по умолчанию не может быть установлено в поле отключения сопоставления, поскольку создание экземпляра протокола может быть приостановлено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), еще не использованных.

Дополнительные сведения о параметрах «Адрес подчиненного устройства», «Время ожидания связи» и «Максимальное количество повторных попыток» см. в примечаниях в разделе «Конфигурация устройств — Конфигурация символического сопоставления».

#### 5.7.4.2.3. Конфигурация сопоставлений — конфигурация для прямого представления (%Q)

## 5. КОНФИГУРАЦИЯ

Параметры отношений MODBUS, показанные на рисунках ниже, соответствуют параметрам, описанным в таблице ниже:

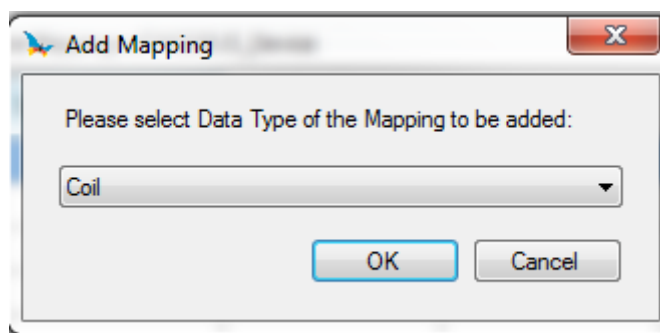


Рисунок 69: Тип данных MODBUS

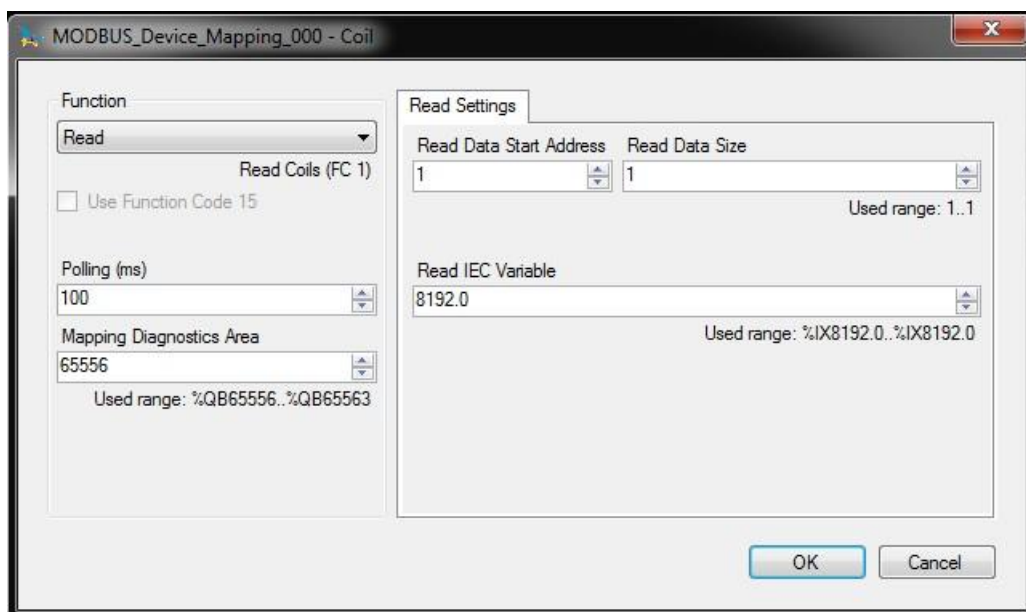


Рисунок 70: Функция MODBUS Function

В приведенной ниже Таблице количество заводских настроек По умолчанию и значения столбца Варианты могут различаться в зависимости от типа данных и функции MODBUS (FC).

Конфигурация	Описание	Значение по умолчанию	Варианты
Функция	Тип функции MODBUS	чтение	Чтение Запись Чтение/запись Запись маски
Опрос (мс)	Период связи (мс)	100	0 до 3600000
Отображение области диагностики	Начальный адрес диагностики отношения MODBUS (%Q)	-	0 до 2147483640
Чтение начального адреса данных	Начальный адрес считанных данных MODBUS	1	1 до 65536
Чтение размера данных	Количество считанных данных MODBUS	-	Зависит от используемой функции
Чтение переменной IEC	Начальный адрес переменных чтения (%I)	-	0 до 2147483646
Запись начального адреса данных	Начальный адрес данных записи MODBUS	1	1 до 65536
Запись размера данных	Количество данных записи MODBUS	-	Зависит от используемой функции

## 5. КОНФИГУРАЦИЯ

Запись переменной IEC	Начальный адрес переменных записи (%Q)	-	0 до 2147483647
Маска записи переменных IEC	Начальный адрес переменных для маски записи (%Q)	-	0 до 2147483644

Таблица 90: Сопоставление устройств

### Примечания:

**Функция:** доступные типы данных подробно описаны в Таблице 109, а функции MODBUS (FC) доступны в Таблице 107.

**Опрос:** этот параметр указывает, как часто должна выполняться связь, установленная для данного отношения. В конце связи будет ожидаться время, равное настроенному опросу, и после этого будет выполнена новая коммуникация как можно скорее.

**Область диагностики сопоставлений:** это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦП, который можно найти в разделе Память. Диагностика отношений conRiseupokd MODBUS описана в Таблице 86.

**Чтение/запись размера данных:** сведения о размере данных, поддерживаемом каждой функцией, описаны в примечаниях к разделу Конфигурация запросов — Настройки символического отображения.

### ВНИМАНИЕ

При доступе к памяти данных связи между устройствами с разным порядком байтов (Little-Endian и Big-Endian) может произойти инверсия данных чтения/записи. В этом случае пользователь должен скорректировать данные в приложении.

**Чтение переменной IEC:** если типом данных MODBUS является Coil или Input Status (bit), начальный адрес переменных чтения IEC будет иметь, например, формат %IX10.1. Однако, если типом данных MODBUS является регистр хранения или регистр ввода (16 бит), начальный адрес переменных чтения IEC будет %IW. Это поле ограничено размером адресуемой памяти входных переменных (%I) на ЦП, который можно найти в разделе Память.

**Запись переменной IEC:** если тип данных MODBUS — Coil, начальный адрес записываемых переменных IEC будет, например, в формате %QX10.1. Однако если типом данных MODBUS является регистр временного хранения (16 бит), начальный адрес записываемых переменных IEC будет %QW. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦП, который можно найти в разделе Память.

**Маска записи:** функция Запись маски (FC 22) использует логику между уже записанным значением и двумя словами, которые согласовываются в этом поле, используя %QW(0) для маски И и %QW(2) для маски ИЛИ. ; позволяя пользователю обрабатывать слово. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе Память.

**Значение по умолчанию:** заводское значение по умолчанию не может быть установлено для полей «Область диагностики сопоставления», «Чтение переменной IEC», «Запись переменной IEC» и «Маска записи переменных IEC», поскольку создание связи может быть выполнено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), которые все еще не используются. Заводские настройки по умолчанию не могут быть установлены в полях Размер данных для чтения/записи, так как они будут различаться в зависимости от выбранного типа данных MODBUS.

### 5.7.5. Ведомое устройство MODBUS RTU

Этот протокол доступен для серии Nexto на ее последовательных каналах. При выборе этой опции в MasterTool IEC XE ЦП становится ведомым устройством связи MODBUS, позволяя подключаться к ведущим устройствам MODBUS RTU.

### ВНИМАНИЕ

В отличие от других задач приложения, при достижении отметки в отладке MainTask задача экземпляра MODBUS RTU Master или любая другая задача MODBUS перестанет выполняться в момент попытки записи в область памяти. Это происходит для того, чтобы поддерживать согласованность данных областей памяти, пока MainTask не запишет.

Существует два способа настройки этого протокола. В первом используется прямое представление (%Q), в котором переменные определяются вашим адресом. Второй — через символическое отображение, где переменные определяются вашим именем.

Независимо от режима Конфигурация шаги по вставке экземпляра протокола и конфигунок последовательного

## 5. КОНФИГУРАЦИЯ

интерфейса одинаковы. Процедура вставки экземпляра протокола подробно описана в Руководстве пользователя MasterTool IEC XE — MU299609. Остальные шаги Конфигурации описаны ниже для каждого режима:

Добавьте экземпляр ведомого протокола MODBUS RTU в последовательный канал COM 1 или COM 2 (или оба, в случае двух коммуникационных сетей). Чтобы выполнить эту процедуру, см. раздел «Вставка экземпляра протокола».

Настройте последовательный интерфейс, выберите скорость передачи данных, поведение сигналов RTS/CTS, четность, канал стоповых битов и другие. См. раздел Конфигурация последовательных интерфейсов.

### 5.7.5.1. Конфигурация ведомого протокола MODBUS с помощью символьного отображения

Чтобы настроить этот протокол с помощью символьного отображения, необходимо выполнить следующие шаги:

Настройте общие параметры протокола ведомого устройства MODBUS, такие как: адрес ведомого устройства и время связи (доступно с помощью кнопки дополнительных настроек ведомого устройства).

Добавьте и настройте отношения MODBUS, указав имя переменной, тип данных MODBUS и начальный адрес данных. Автоматически размер и диапазон данных будут заполнены в соответствии с объявленным типом переменной.

#### 5.7.5.1.1. Общие параметры ведомого протокола MODBUS – Конфигурация через символическое отображение

Общие параметры, находящиеся на начальном экране протокола MODBUS (Рисунок ниже), определяются как.

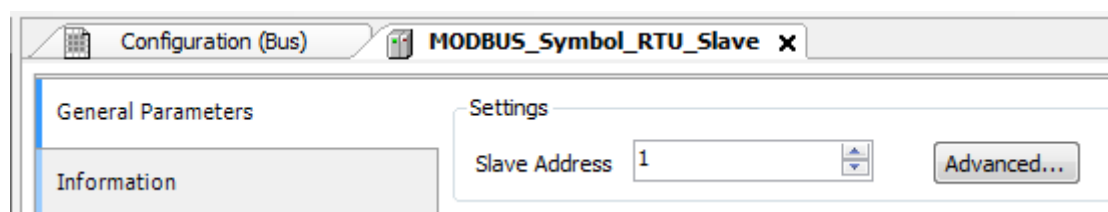


Рисунок 71: Экран конфигурации подчиненного устройства MODBUS RTU

Конфигурация	Описание	По умолчанию	Варианты
Адрес подчиненного устройства	Адрес подчиненного устройства MODBUS	1	1 до 255

Таблица 91: Конфигурация подчиненного устройства

Время связи ведомого протокола MODBUS, которое можно найти с помощью кнопки «Дополнительно...» на экране «Конфигурация», делится на: цикл задачи, задержку отправки и минимальный межкадровый интервал, как показано на рисунке ниже и в таблице ниже.

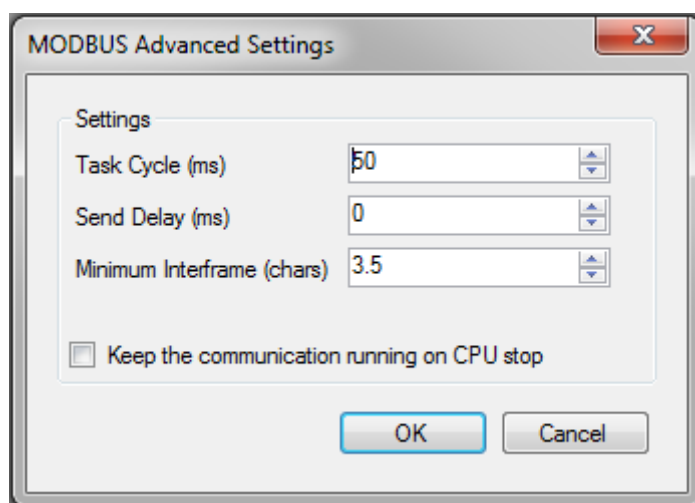


Рисунок 72: Расширенные конфигурации ведомого Modbus

Конфигурация	Описание	По умолчанию	Варианты
--------------	----------	--------------	----------

## 5. КОНФИГУРАЦИЯ

Цикл задачи (мс)	Время выполнения экземпляра в цикле, без учета собственного времени выполнения	50	20 до 100
Задержка отправки (мс)	Задержка ответа на передачу	0	0 до 65535
Минимальный межкадровый интервал (символы)	Минимальное время молчания между разными кадрами	3.5	3.5 до 100.0
Сохраняйте связь при остановке ЦП	Разрешить работу ведомого символа MODBUS, когда ЦП находится в состоянии STOP или находится в точке останова.	непроверенный	Проверенный или непроверенный

Таблица 92: Расширенные конфигурации ведомого Modbus

### Примечания:

**Цикл задач:** пользователю следует быть осторожным при изменении этого параметра, так как он напрямую влияет на время ответа, объем данных для сканирования и, главным образом, на баланс ресурсов ЦП между коммуникациями и другими задачами.

Задержка отправки: ответ на протокол MODBUS может вызвать проблемы в определенные моменты, например, в интерфейсе RS-485 или другом полудуплексе. Иногда возникает задержка между временем запроса от ведущего и тишиной на физической линии (задержка ведомого для обнуления RTS и перевода RS-485 в состояние высокого импеданса). Чтобы решить эту проблему, мастер может подождать определенное время в этом поле перед отправкой нового запроса. В противном случае первые байты, переданные мастером, могут быть потеряны.

Минимальный межкадровый интервал: стандарт MODBUS определяет это время как 3,5 символа, но этот параметр настраивается для обслуживания устройств, которые не соответствуют стандарту.

Диагностика протокола MODBUS Slave и команды conRiseynokd, либо символическим отображением, либо прямым представлением, хранятся в переменных T\_DIAG\_MODBUS\_RTU\_SLAVE\_1. Для отображения прямого представления они также состоят из 4 байтов и 8 слов, которые описаны в Таблице ниже (где «n» — значение configured в %Q Начальный адрес области диагностики):

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_RTU_SLAVE_1.*	Размер	Описание
<b>Биты диагностики:</b>			
%QX(n).0	tDiag. bRunning	BIT	Ведомый находится в режиме выполнения.
%QX(n).1	tDiag. bNotRunning	BIT	Подчиненное устройство не находится в процессе выполнения (см. бит: bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, так как ведомое устройство было прервано пользователем через командные биты.
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращенный диагноз.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращенный диагноз.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращенный диагноз.
%QX(n).6	tDiag. bModuleFailure	BIT	Прекращенный диагноз.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Сдержанный.
<b>Коды ошибки:</b>			

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS _RTU_SLAVE_1.*	Размер	Описание
%QB(n+1)	eErrorCode	SERIAL_STATU S (BYTE)	0: ошибки нет. 1: неверный последовательный порт. 2: неверный режим последовательного порта 3: неверная скорость передачи данных 4: неверные биты данных 5: неверная четность 6: неверные стоповые биты 7: неверные параметры сигнала модема 8: неверный пороговый параметр UART RX 9: неверный параметр тайм-аута 10: последовательный порт занят 11: аппаратная ошибка UART 12: удаленная аппаратная ошибка 20: неверный размер буфера передачи 21: неверный метод сигнала модема 22: тайм-аут CTS = true 23: тайм-аут CTS = ложь 24: ошибка тайм-аута передачи 30: неверный размер буфера приема 31: ошибка тайм-аута приема 32: управление потоком настроено иначе, чем вручную 33: неверный контроль потока для настроенного последовательного порта 34: прием данных в обычном режиме запрещен 35: прием данных запрещен в расширенном режиме 36: прерывание DCD не разрешено 37: прерывание CTS не разрешено 38: прерывание DSR не разрешено 39: последовательный порт не настроен 50: внутренняя ошибка в последовательном порту
<b>Командные биты, автоматически инициализируются:</b>			
%QX(n+2).0	tCommand. bStop	BIT	Остановка подчиненного устройства
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск подчиненного устройства
%QX(n+2).2	tCommand. bResetCounter	BIT	Перезапустите статистику диагностики (счетчики).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Зарезервировано.
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Зарезервировано.
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Зарезервировано.
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Зарезервировано.

Переменная прямого представлен ия	Диагностическая переменная T_DIAG_MODBUS _RTU_SLAVE_1.*	Размер	Описание
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.
<b>Статистика связи:</b>			
%QW(n+4)	tStat. wRXRequests	WORD	Счетчик обычных запросов, полученных ведомым устройством и нормально отвеченных. В случае широковещательной команды этот счетчик увеличивается, но не передается (0 до 65535).
%QW(n+6)	tStat. wTXExceptionResponses	WORD	Счетчик обычных запросов, полученных ведомым устройством и на которые был дан ответ с кодом исключения. В случае широковещательной команды этот счетчик увеличивается, но не передается (от 0 до 65535). Коды исключений: 1: код функции (FC) разрешен, но не поддерживается. 2: отношение не найдено в этих данных MODBUS. 3: недопустимое значение для этого поля. 128: мастер/клиент не имеет права на запись или чтение. 129: связь MODBUS отключена.
%QW(n+8)	tStat. wRXFrames	WORD	Счетчик кадров, полученных слейвом. Это считается кадром, который обрабатывается, и за ним следует минимальное межкадровое молчание, другими словами, также вычисляется недопустимое сообщение (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalRequests	WORD	Счетчик недопустимых запросов. Это кадры, которые начинаются с адреса 0 (широковещательный) или с адреса подчиненного устройства MODBUS, но не являются законными запросами — неверный синтаксис, меньшие кадры, неверный CRC — (от 0 до 65535).
%QW(n+12)	tStat. wRXOverrunErrors	WORD	Счетчик кадров с ошибками переполнения за время прием — UART FIFO или линия RX — (от 0 до 65535).
%QW(n+14)	tStat. wRXIncompleteFrames	WORD	Счетчик кадров с ошибками построения, четности или сбоя при приеме (от 0 до 65535).
%QW(n+16)	tStat. wCTSTimeOutErrors	WORD	Счетчик ошибок тайм-аута CTS при использовании квитирования RTS/CTS во время передачи (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано.

## 5. КОНФИГУРАЦИЯ

Таблица 93: Диагностика подчиненного устройства MODBUS RTU

### Примечание

**Счетчики:** все счетчики диагностики ведомых устройств MODBUS RTU обнуляются при превышении предельного значения 65535.

### 5.7.5.1.2. Конфигурация отношений — настройка символического отображения

Отношения конфигурации MODBUS, показанная на рисунке ниже, соответствует параметрам, описанным в таблице ниже:

Mappings

	Value Variable	Data Type	Data Start Address	Absolute Data Start Address	Data Size	Data Range
*						

Рисунок 73: MODBUS Data Mappings Screen

Конфигурация	Описание	По умолчанию	Варианты
Переменная значения	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Тип данных	Тип данных MODBUS	-	Катушка Статус ввода Холдинговый реестр Входной регистр
Начальный адрес данных	Начальный адрес данных MODBUS	-	1 до 65536
Абсолютный начальный адрес данных	Абсолютный начальный адрес данных MODBUS в соответствии с его типом	-	-
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Настроен диапазон адресов данных	-	-

Таблица 94: Конфигурации отображения MODBUS

### Примечания:

**Переменная значения:** это поле используется для указания символической переменной в отношении MODBUS.

**Тип данных:** это поле используется для указания типа данных, используемых в отношении MODBUS.

Тип данных	Размер [биты]	Описание
Катушка	1	Цифровой выход, который может быть прочитан или записан.
Статус ввода	1	Цифровой вход (только чтение).
Холдинговый реестр	16	Аналоговый выход, который может быть прочитан или записан.
Входной регистр	16	Аналоговый вход (только чтение).

Таблица 95: Типы данных MODBUS, поддерживаемые процессорами Nexto

**Начальный адрес данных:** начальный адрес данных отношения MODBUS.

**Размер данных:** значение размера данных устанавливает максимальный объем данных, к которым отношение MODBUS может получить доступ с начального адреса. Таким образом, чтобы прочитать Продолжающийся диапазон адресов, необходимо, чтобы все адреса были объявлены в одном отношении. Это поле варьируется в зависимости от типа данных MODBUS.

**Диапазон данных:** это поле показывает пользователю диапазон адресов памяти, используемый связью MODBUS.

**ВНИМАНИЕ**

В отличие от других прикладных задач, при достижении метки очистки в MainTask задача экземпляра MODBUS RTU Slave и любая другая задача MODBUS перестанут выполняться в тот момент, когда она попытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных областей памяти, когда основная задача не запущена..

### 5.7.5.2. Конфигурация ведомого протокола MODBUS через прямое представление (%Q)

Чтобы настроить этот протокол с использованием прямого представления (%Q), необходимо выполнить следующие шаги:

Подтвердите общие параметры подчиненного протокола MODBUS, такие как: время связи, адрес и переменные прямого представления (%Q) для получения диагностических и контрольных соотношений.

Добавьте и настройте соотношения MODBUS, указав тип данных MODBUS, переменные прямого представления (%Q) для получения/записи данных и объем данных для передачи.

Описание каждой настройки приведено ниже в этом разделе.

#### 5.7.5.2.1. Общие параметры ведомого протокола MODBUS — настройка через прямое представление (%Q)

Общие параметры, находящиеся на главном экране конфигурации протокола MODBUS (Рисунок ниже), определяются как:

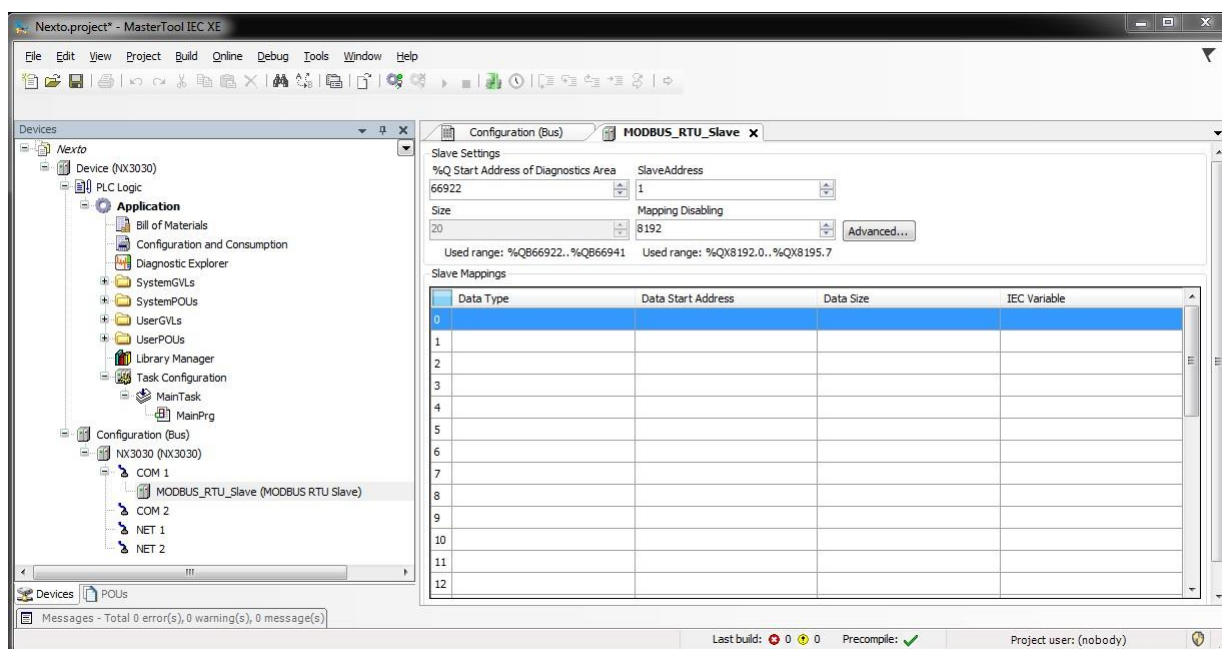


Рисунок 74: Экран конфигурации ведомого устройства MODBUS RTU от прямого представителя. Переменные адресации и прямого представления (%Q) для управления взаимосвязями и диагностики:

Конфигурация	Описание	Значение по умолчанию	Варианты
%Q Начальный адрес области диагностики	Начальный адрес диагностических переменных	-	0 до 2147483628
Размер	Размер области диагностики	-	Отключено для редактирования
Адрес подчиненного устройства	Адрес подчиненного устройства MODBUS	1	1 до 255
Отключение отображения	Исходный адрес, используемый для отключения соотношений MODBUS	-	0 до 2147483644

Таблица 96: Настройки переменных адреса и прямого представления

**Примечания:**

**%Q Начальный адрес области диагностики:** это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе Память.

Адрес ведомого устройства: важно отметить, что ведомое устройство принимает широковещательные запросы, когда ведущее устройство отправляет команду с нулевым адресом. Кроме того, в соответствии со стандартом MODBUS действительный диапазон адресов для ведомых устройств составляет от 1 до 247. Адреса с 248 по 255 зарезервированы.

Отключение отображения: состоит из 32 битов, используемых для индивидуального отключения 32 отношений MODBUS, настроенных в пространстве подчиненных отображений. Отношение отключено, когда соответствующий бит равен 1, в противном случае отображение включено. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе «Память».

Значение по умолчанию: заводское значение по умолчанию не может быть установлено для полей %Q Start Address of Diagnostic Area и Mapping Disabled, поскольку создание отношения может быть выполнено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), которые все еще не используются.

Ведомое устройство MODBUS по протоколу прямого представления прекращает обмен данными, когда ЦП находится в состоянии STOP или останавливается в точке останова.

Диагностика и команды протокола MODBUS описаны в Таблице 93.

Время связи протокола MODBUS Slave, указанное на кнопке Дополнительно... экрана Конфигурация, описано в Таблице 92.

**5.7.5.2.2. Конфигурация отображений — конфигурация через прямое представление (%Q)**

Параметры отношений MODBUS, показанные на рисунках ниже, соответствуют параметрам, описанным в таблице ниже:

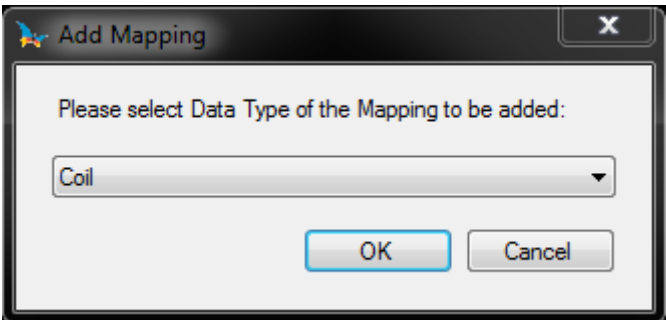


Рисунок 75: Добавление отношений MODBUS

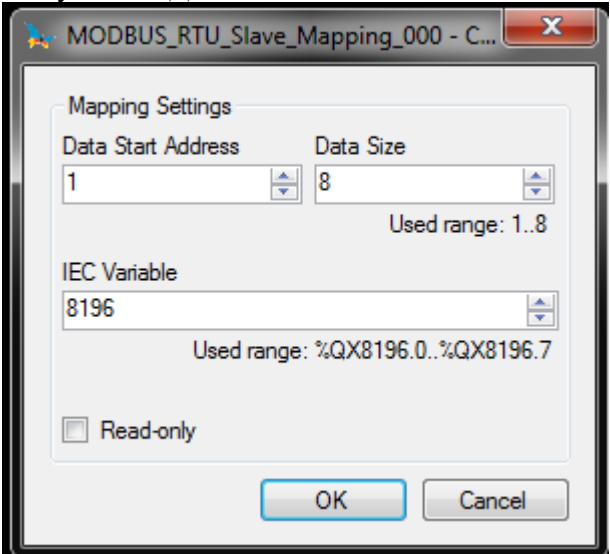


Рисунок 76: Настройка отношения MODBUS

Конфигурация	Описание	По умолчанию Value	Варианты
--------------	----------	--------------------	----------

## 5. КОНФИГУРАЦИЯ

<b>Data Тип</b>	MODBUS data Тип	Coil	Coil (1 bit) Holding Register (16 bits) Input Register (16 bits) Input Status (1 bit)
<b>Data Start Address</b>	Initial address of the MOD-BUS data	1	1 to 65536
<b>Data Size</b>	Number of MODBUS data	-	1 to 65536
<b>IEC Variable</b>	Initial address of variables (%Q)	-	0 to 2147483647
<b>Read-only</b>	Only allows reading	Disabled	Enabled or disabled

Таблица 97: Сопоставления подчиненных

### Примечания:

**Варианты:** значения, записанные в столбце Варианты, могут варьироваться в зависимости от настроенных данных MODBUS.

**Размер данных:** значение размера данных определяет максимальный объем данных, к которым отношение MODBUS может получить доступ, начиная с начального адреса. Таким образом, чтобы прочитать Продолжающийся диапазон адресов, необходимо, чтобы все адреса были объявлены в одном интерфейсе. Это поле зависит от настроенного типа данных MODBUS, т. е. при выборе Coil или Input Status поле Data Size должно быть кратно восьми. Кроме того, максимальный объем не должен превышать размер выходной адресуемой памяти и не должен назначать те же значения, которые используются в приложении.

**Переменная МЭК:** в случае, если типом данных MODBUS является Coil или Input Status (bit), начальный адрес переменных IEC будет иметь формат %QX10.1. Однако, если типом данных MODBUS является регистр хранения или регистр ввода (16 бит), начальный адрес переменных IEC будет иметь формат %QW. Это поле ограничено размером памяти адресуемых выходных переменных (%Q) каждого ЦП, что можно увидеть в разделе «Память».

### ВНИМАНИЕ

При доступе к памяти данных связи между устройствами с разным порядком байтов (Little-Endian и Big-Endian) может произойти инверсия данных чтения/записи. В этом случае пользователь должен скорректировать данные в приложении.

**Только для чтения:** при включении он позволяет ведущему устройству связи только считывать переменные данные. Это не позволяет писать. Эта опция действительна только для функций записи.

**Значение по умолчанию:** Значение по умолчанию не может быть определено для поля IEC Variable, так как создание отношения может быть выполнено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), которые все еще не используются. Значение По умолчанию не может быть определено для поля Размер данных, поскольку оно будет варьироваться в зависимости от выбранного типа данных MODBUS.

В ранее определенных соотношениях максимальный размер данных MODBUS может быть 65535 (максимальное значение configured в поле Data Size). Однако запрос, поступающий на ведомое устройство MODBUS RTU, должен адресоваться к подгруппе этого сопоставления, и эта группа должна иметь не более размера данных, зависящего от функционального кода, который определен ниже:

Читать катушки (FC 1): 2000

Чтение состояния входа (FC 2): 2000

Чтение регистров хранения (FC 3): 125

Чтение входных регистров (FC 4): 125

Запись одиночной катушки (FC 5): 1

Запись одного регистра хранения (FC 6): 1 Force Multiple Coils (FC 15): 1968

Регистры хранения записи (FC 16): 123

Регистр записи маски (FC 22): 1 регистр хранения чтения/записи (FC 23):

- Прочитано: 121

- Запись: 121

### ВНИМАНИЕ

В отличие от других прикладных задач, при достижении метки очистки в MainTask задача подчиненного экземпляра MODBUS RTU и любая другая задача MODBUS перестанут выполняться в тот момент, когда она попытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных областей памяти, когда основная задача не запущена..

#### 5.7.6. MODBUS Ethernet

Связь с несколькими мастерами позволяет ЦП Nexto считывать или записывать переменные MODBUS в другие контроллеры или ЧМИ, совместимые с протоколом MODBUS TCP или MODBUS RTU, через TCP. ЦП Nexto может одновременно быть клиентом и сервером в одной коммуникационной сети или даже иметь несколько экземпляров, связанных с интерфейсом Ethernet. Неважно, MODBUS TCP или MODBUS RTU через TCP, как описано в Таблице 5.7.

На рисунке ниже представлены некоторые возможности связи с использованием протокола MODBUS TCP одновременно с MODBUS RTU по протоколу TCP.

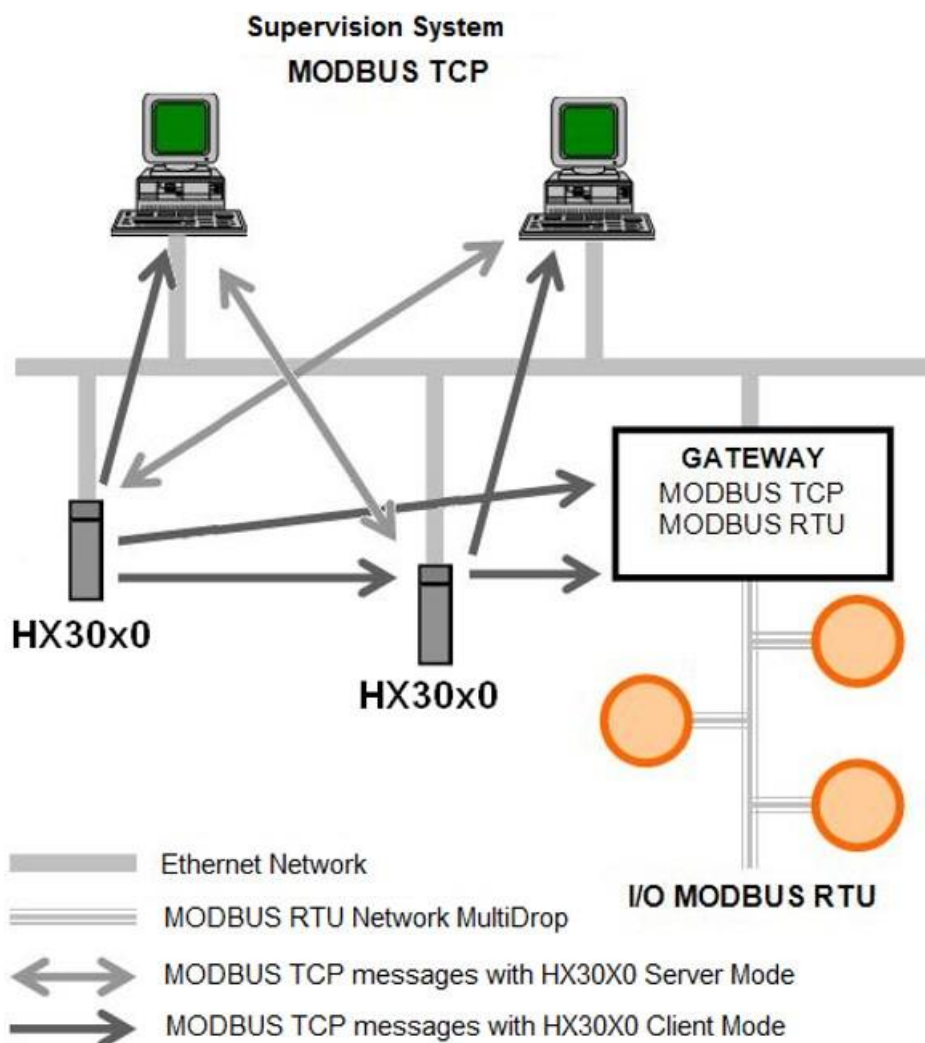


Рисунок 77: Сеть связи MODBUS TCP

Связь переменных MODBUS с символическими переменными ЦП осуществляется пользователем посредством определения отношений с помощью инструмента Конфигурация MasterTool IEC XE. Можно настроить до 32 отношений для режима сервера и до 128 отношений для режима клиента. Отношения в клиентском режиме, с другой стороны, должны учитывать максимальный размер данных функции MODBUS: 125 регистров (регистры ввода или регистры хранения) или 2000 бит (катушки или состояние ввода). Эта информация подробно описана в описании каждого протокола.

Все отношения в режиме клиента или в режиме сервера могут быть отключены с помощью переменных прямого представления (%Q), определенных MasterTool IEC XE как Отключающие переменные. Отключение может происходить через общие биты, влияющие на все отношения режима работы, или через определенные биты, влияющие на определенные отношения.

Для отношений режима сервера кластеры адресов IPes могут быть определены с разрешениями на запись и чтение, называемыми фильтрами. Это делается путем определения сетевого IP-адреса и маски подсети, что приводит к группе клиентских IP-адресов, которые могут читать и записывать переменные отношения. Функции чтения/записи фильтруются, другими словами, они не могут быть запрошены любым клиентом, независимо от IP-адреса Адреса. Эта информация подробно описана в протоколе сервера Ethernet MODBUS.

Когда протокол MODBUS TCP используется в клиентском режиме, можно использовать функцию множественных запросов с одним и тем же TCP-соединением для ускорения связи с серверами. Если эта функция нежелательна или не поддерживается сервером, ее можно отключить (действие на уровне отношений). Важно подчеркнуть, что максимальное количество TCP-соединений между клиентом и сервером равно 63. При изменении некоторых параметров неактивные соединения могут быть закрыты, что позволяет открывать новые соединения.

Таблицы ниже содержат, соответственно, полный список данных и функций MODBUS, поддерживаемых процессорами Nexto.

Тип данных	Размер [биты]	Описание
Катушка	1	Цифровой выход, который может быть прочитан или записан.
Статус ввода	1	Цифровой вход (только чтение).
Холдинговый реестр	16	Аналоговый выход, который может быть прочитан или записан.
Входной регистр	16	Аналоговый вход (только чтение).

Таблица 98: Типы данных MODBUS, поддерживаемые процессорами Nexto

Код		Описание
DEC	HEX	
1	0x01	Читать катушки (FC 01)
2	0x02	Чтение состояния входа (FC 02)
3	0x03	Чтение регистров хранения (FC 03)
4	0x04	Чтение входных регистров (FC 04)
5	0x05	Запись одиночной катушки (FC 05)
6	0x06	Запись единого регистра временного хранения (FC 06)
15	0x0F	Запись нескольких катушек (FC 15)
16	0x10	Запись нескольких регистров временного хранения (FC 16)
22	0x16	Регистр хранения записи по маске (FC 22)
23	0x17	Чтение/запись нескольких регистров временного хранения (FC 23)

Таблица 99: Функции MODBUS, поддерживаемые процессорами Nexto

Независимо от режима конфигурации шаги по вставке экземпляра протокола и настройке интерфейса Ethernet одинаковы. Остальные этапы настройки описаны ниже для каждой модальности.

Добавьте один или несколько экземпляров клиентского или серверного протокола MODBUS Ethernet в канал Ethernet. Чтобы выполнить эту процедуру, обратитесь к разделу Вставка экземпляра протокола.

Настройте интерфейс Ethernet. Чтобы выполнить эту процедуру, см. раздел Конфигурация Ethernet-интерфейсов.

### 5.7.7. Ethernet-клиент MODBUS

Этот протокол доступен для всех ЦП серии Nexto на его каналах Ethernet. При выборе этой опции в MasterTool IEC XE ЦП становится коммуникационным клиентом MODBUS, разрешая доступ к другим устройствам с тем же протоколом, когда он находится в режиме выполнения (режиме выполнения).

Существует два способа настройки этого протокола. В первом используется прямое представление (%Q), в котором переменные определяются вашим адресом. Второй — через символическое отображение, где переменные определяются вашим именем.

Процедура вставки экземпляра протокола подробно описана в Руководстве пользователя MasterTool IEC XE — MU299609 или в разделе «Вставка экземпляра протокола».

#### 5.7.7.1. Конфигурация клиента MODBUS Ethernet через символическое отображение

Чтобы подтвердить этот протокол с помощью Symbolic Mapping, необходимо выполнить следующие шаги:

Подпишитесь на общие параметры клиента протокола MODBUS, с протоколом управления передачей (TCP) или RTU через TCP.

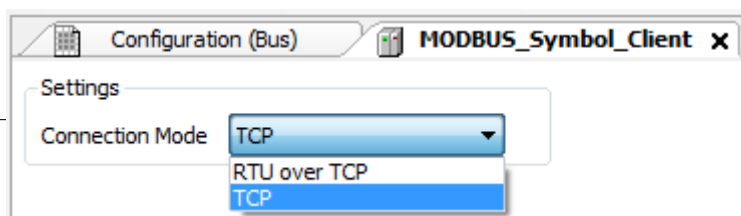
Добавьте и настройте устройства, задав IP-адрес, порт, адрес ведомого устройства и время ожидания связи (доступно с помощью кнопки «Дополнительные параметры» устройства).

Добавьте и настройте сопоставления MODBUS, указав имя переменной, тип данных, начальный адрес данных, размер данных и переменную, которая будет получать данные о качестве.

Добавьте и настройте запрос MODBUS, указав желаемую функцию, время сканирования запроса, начальный адрес (чтение/запись), размер данных (чтение/запись), переменную, которая будет получать качество данных и переменную ответственный за отключение запроса.

##### 5.7.7.1.1. Общие параметры клиентского протокола MODBUS – конфигурация с помощью символического отображения

Общие параметры, находящиеся на начальном экране конфигурации протокола MODBUS (Рисунок ниже), определяются как:



Конфигурация	Описание	По умолчанию	Варианты
Режим подключения	Выбор протокола	TCP	RTU через TCP TCP

Таблица 100: Общие конфигурации клиента MODBUS

Диагностика клиентского протокола MODBUS и команды, сконфигурированные с помощью символического отображения или прямого представления, хранятся в переменных T\_DIAG\_MODBUS\_ETH\_CLIENT\_1. Для прямого отображения представления они также состоят из 4 байтов и 8 слов, которые описаны в Таблице ниже (где «п» — настроенное значение в %Q Начальный адрес области диагностики):

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_CLIENT_1.*	Размер	Описание
<b>Диагностические биты:</b>			
%QX(n).0	tDiag. bRunning	BIT	Клиент находится в режиме выполнения.
%QX(n).1	tDiag. bNotRunning	BIT	Клиент не находится в режиме выполнения (см. бит bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, так как клиент был прерван пользователем через командные биты.
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращена диагностика.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращена диагностика.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращена диагностика.
%QX(n).6	tDiag. bModuleFailure	BIT	Указывает на неисправность модуля или отсутствие модуля.
%QX(n).7	tDiag. bAllDevicesCommFailure	BIT	Указывает, что все устройства, обнаруженные в Клиенте, неисправны.
%QB(n+1)	byDiag_1_reserved	BYTE	Зарезервировано.
<b>Командные биты, автоматически инициализируются:</b>			
%QX(n+2).0	tCommand. bStop	BIT	Стоп клиент.
<b>Диагностическая переменная T_DIAG_MODBUS_ETH_CLIENT_1.*</b>			
%QX(n+2).1	tCommand. bRestart	BIT	Перезапустите клиент.
%QX(n+2).2	tCommand. bResetCounter	BIT	Перезапустите диагностическую статистику (счетчики).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Зарезервировано.
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Зарезервировано.

## 5. КОНФИГУРАЦИЯ

%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Зарезервировано.
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Зарезервировано.
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.
<b>Статистика связи:</b>			
%QW(n+4)	tStat. wTXRequests	WORD	Счетчик количества запросов, переданных клиентом (от 0 до 65535).
%QW(n+6)	tStat. wRXNormalResponses	WORD	Счетчик нормальных ответов, полученных клиентом (от 0 до 65535).
%QW(n+8)	tStat. wRXExceptionResponses	WORD	Счетчик ответов с кодом исключения (от 0 до 65535).
%QW(n+10)	tStat. wRXIllegalResponses	WORD	Счетчик недопустимых ответов, полученных клиентом — неверный синтаксис, неверный CRC или получено недостаточно байтов (от 0 до 65535).
%QW(n+12)	tStat. wDiag_12_reserved	WORD	Зарезервировано.
%QW(n+14)	tStat. wDiag_14_reserved	WORD	Зарезервировано.
%QW(n+16)	tStat. wDiag_16_reserved	WORD	Зарезервировано.
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано.

Таблица 101: Диагностика клиентского протокола MODBUS

### Примечание:

**Счетчики:** все счетчики диагностики клиента MODBUS TCP обнуляются при превышении предельного значения 65535.

5.7.7.1.2. Конфигурация устройства — Конфигурация через символическое сопоставление  
Конфигурация устройств, показанная на рисунке ниже, соответствует следующим параметрам:

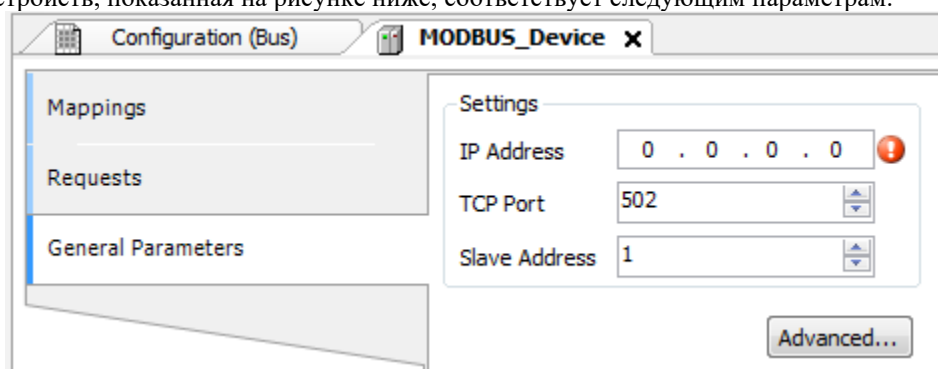


Рисунок 79: Настройки общих параметров устройства

Конфигурация	Описание	По умолчанию	Варианты
Адрес IP	Server Адрес IP	0.0.0.0	1.0.0.1 до 223.255.255.255
Порт TCP	TCP port	502	2 до 65534
Адрес подчиненного устройства	MODBUS Slave address	1	0 до 255

Таблица 102: Общие конфигурации клиента MODBUS

**Примечания:**

**Адрес IP:** Адрес IP устройства сервера Modbus.

**Порт TCP:** если к одному интерфейсу Ethernet добавлено несколько экземпляров протокола, для каждого экземпляра необходимо выбрать разные порты TCP. Некоторые TCP-порты, среди упомянутых выше возможностей, зарезервированы и поэтому не могут использоваться. См. раздел Таблица зарезервированных портов TCP/UDP.

**Адрес подчиненного устройства:** в соответствии со стандартом MODBUS допустимый диапазон адресов для подчиненных устройств составляет от 0 до 247, где адреса с 248 по 255 зарезервированы. Когда мастер отправляет команду записи с нулевым адресом, он выполняет ширококестельные запросы в сети.

Параметры в расширенных настройках клиентского устройства MODBUS, доступные по кнопке Дополнительно... на вкладке Общие параметры, делятся на: Максимальное количество одновременных запросов, Время ожидания связи, Время ожидания режима соединения и Время неактивности.

Конфигурация	Описание	По умолчанию	Варианты
Максимальный одновременный запрос	Количество одновременных запросов, которые клиент может запросить у сервера	1	1 до 8
Время ожидания связи (мс)	Тайм-аут уровня приложения в мс	3000	10 до 65535
Режим	Определяет, когда соединение с сервером завершено клиентом	Соединение закрывается после неактивного времени (с): от 10 до 3600.	Соединение закрывается после тайм-аута. Соединение закрывается в конце каждого сеанса связи. Соединение закрывается после неактивного времени (с): от 10 до 3600.
Неактивное время (с)	Время бездействия	10	3600

Таблица 103: Расширенные настройки клиента MODBUS

**Примечания:**

**Максимальное количество одновременных запросов:** он используется с высоким циклом сканирования. Этот параметр имеет фиксированное значение 1 (недоступно для редактирования), если протоколом conRiseунокd является MODBUS RTU через TCP.

**Время ожидания связи:** время ожидания связи — это время, в течение которого клиент будет ожидать ответа сервера на запрос. Для клиентского устройства MODBUS необходимо учитывать две переменные системы: время, необходимое серверу для обработки запроса, и задержку отправки ответа, если она установлена на сервере. Рекомендуется, чтобы время ожидания было равно удвоенной сумме этих параметров или превышало ее. Для получения дополнительной информации см. раздел «Производительность связи».

**Режим:** определяет, когда клиент завершает соединение с сервером. Ниже приведены доступные Варианты:

Соединение закрывается по истечении тайм-аута или Соединение никогда не закрывается в обычных ситуациях: Эти Варианты представляют такое же поведение Клиента, закрывают соединение из-за отсутствия ответа на запрос со стороны Сервера до достижения тайм-аута связи.

Соединение закрывается в конце каждой связи: Соединение закрывается Клиентом после завершения каждого запроса. Соединение закрывается после неактивного времени: соединение будет закрыто клиентом, если оно достигает неактивного времени без выполнения запроса к серверу.

**Время бездействия:** время бездействия соединения.

#### 5.7.7.1.3. Конфигурация сопоставлений — конфигурация через символическое сопоставление

Конфигурация отношений MODBUS, показанная на рисунке ниже, соответствует параметрам, описанным в таблице ниже:

## 5. КОНФИГУРАЦИЯ

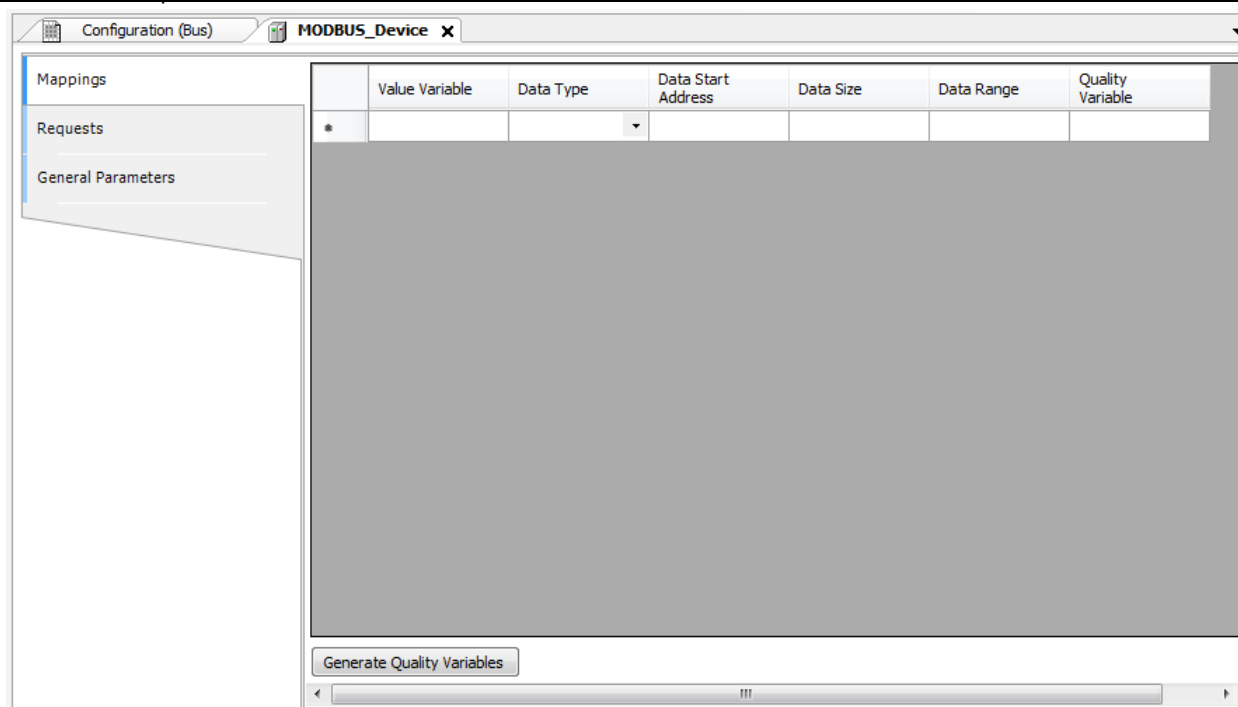


Рисунок 80: Тип данных MODBUS

Конфигурация	Описание	По умолчанию	Варианты
Переменная значения	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Тип данных	Тип данных MODBUS	-	Катушка — запись (1 бит) Катушка — чтение (1 бит) Регистр хранения — запись (16 бит) Регистр хранения - чтение (16 бит) Регистр временного хранения – Маска И (16 бит) Регистр временного хранения — Маска ИЛИ (16 бит) Входной регистр (16 бит) Состояние входа (1 бит)
Начальный адрес данных	Начальный адрес данных MODBUS	-	1 до 65536
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Диапазон адресов настроенных данных	-	-

Таблица 104: Настройки отображения MODBUS

**Примечания:**

**Переменная значения:** это поле используется для указания символической переменной в отношении MODBUS.

**Тип данных:** это поле используется для указания типа данных, используемых в отношении MODBUS.

Тип данных	Размер [биты]	Описание
Катушка - записать	1	Запись цифрового выхода.
Катушка - читать	1	Чтение цифрового выхода.
Проведение реестра - запись	16	Запись аналогового выхода.
Холдинговый регистр - чтение	16	Чтение аналогового выхода.
Холдинговый регистр – Маска И	16	Аналоговый выход, который может быть прочитан или записан с маской И.
Холдинговый регистр — маска ИЛИ	16	Аналоговый выход, который может быть прочитан или записан с маской ИЛИ.
Входной регистр	16	Аналоговый вход, который можно только читать.
Статус ввода	1	Цифровой вход, который можно только прочитать.

Таблица 105: Типы данных, поддерживаемые в MODBUS

**Начальный адрес данных:** начальный адрес данных отображения MODBUS.

**Размер данных:** значение размера определяет максимальный объем данных, к которым может получить доступ интерфейс MODBUS, начиная с начального адреса. Таким образом, чтобы прочитать Продолжающийся диапазон адресов, необходимо, чтобы все адреса были объявлены в одном интерфейсе. Это поле зависит от настроенного типа данных MODBUS.

**Диапазон данных:** это поле показывает пользователю диапазон адресов памяти, используемый интерфейсом MODBUS.

## 5.7.7.1.4. Конфигурация запросов — конфигурация через символическое сопоставление

Конфигурация запросов MODBUS, показанная на рисунке ниже, соответствует параметрам, описанным в таблице ниже:

## 5. КОНФИГУРАЦИЯ

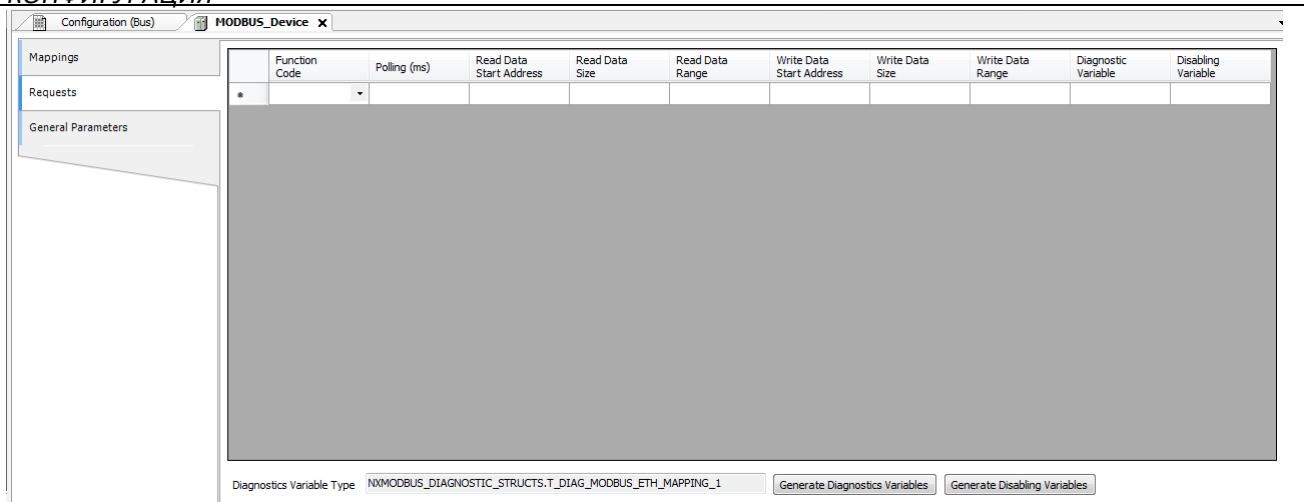


Рисунок 81: Экран запроса данных MODBUS

Конфигурация	Описание	Значение по умолчанию	Варианты
Функциональный код	Тип функции MODBUS	-	1 — Чтение катушек 02 — Чтение состояния входа 03 — Чтение регистров хранения 04 — Чтение входных регистров 05 — Запись одиночной катушки 06 — Запись в один регистр 15 — Запись в несколько катушек 16 — Запись в несколько регистров 22 - Регистр записи маски 23 — Чтение/запись нескольких регистров
Опрос (мс)	Период связи (мс)	100	0 до 3600000
Чтение начального адреса данных	Начальный адрес считанных данных MODBUS	-	1 до 65536
Чтение размера данных	Размер MODBUS	-	Зависит от используемых функций
Чтение диапазона данных	Чтение данных MODBUS	-	0 до 2147483646
Запись начального адреса данных	Чтение диапазона адресов данных	-	1 до 65536
Запись размера данных	Начальный адрес данных записи MODBUS	-	Зависит от используемых функций
Запись диапазона данных	Размер данных MODBUS	-	0 до 2147483647
Диагностическая переменная	Написать MODBUS	-	Имя переменной, объявленной в программе или GVL

Конфигурация	Описание	Значение по умолчанию	Варианты
--------------	----------	-----------------------	----------

## 5. КОНФИГУРАЦИЯ

Диагностическая переменная	Переменная, используемая для отключения связи MODBUS	-	Поле для символической переменной, используемой для индивидуального отключения настроенных запросов MODBUS. Эта переменная должна быть типа BOOL. Переменная может быть простой или элементом массива и может находиться в структурах.
----------------------------	--	---	--

Таблица 106: Конфигурация отношений MODBUS

### Примечания:

**Настройки:** количество заводских настроек По умолчанию и значения столбца Варианты могут различаться в зависимости от типа данных и функции MODBUS (FC).

**Код функции:** Доступны следующие функции MODBUS (FC):

Код		Описание
DEC	HEX	
1	0x01	Чтение катушек (FC 01)
2	0x02	Чтение состояния входа (FC 02)
3	0x03	Чтение регистров хранения (FC 03)
4	0x04	Чтение входных регистров (FC 04)
5	0x05	Запись одиночной катушки (FC 05)
6	0x06	Запись единого регистра временного хранения (FC 06)
15	0x0F	Запись нескольких катушек (FC 15)
16	0x10	Запись нескольких регистров временного хранения (FC 16)
22	0x16	Регистр хранения записи по маске (FC 22)
23	0x17	Чтение/запись нескольких регистров временного хранения (FC 23)

Таблица 107: Функции MODBUS, поддерживаемые процессорами Nexio

Опрос: этот параметр указывает, как часто должна выполняться связь, установленная для этого запроса. По окончании связи будет ожидаться время, равное значению, настроенному в поле polling, и после этого будет выполнена новая коммуникация.

Начальный адрес чтения данных: поле для начального адреса считанных данных MODBUS.

Размер прочитанных данных: минимальное значение размера прочитанных данных равно 1, а максимальное значение зависит от используемой функции MODBUS (FC), как показано ниже:

Читать катушки (FC 01): 2000

Чтение состояния входа (FC 02): 2000

Чтение регистров хранения (FC 03): 125

Чтение входных регистров (FC 04): 125

Чтение/запись нескольких регистров (FC 23): 121

Диапазон считанных данных: в этом поле отображается диапазон считываемых данных MODBUS, настроенный для каждого запроса. Начальный адрес вместе с размером прочитанных данных будет определять диапазон прочитанных данных для каждого запроса.

Начальный адрес записи данных: поле для начального адреса записи данных MODBUS.

Размер записываемых данных: минимальное значение размера записываемых данных равно 1, а максимальное значение зависит от используемой функции MODBUS (FC), как показано ниже:

Запись одиночной катушки (FC 05): 1

Запись одного регистра (FC 06): 1

Напишите несколько катушек (FC 15): 1968 г.

Запись нескольких регистров (FC 16): 123

Регистр записи по маске (FC 22): 1

Чтение/запись нескольких регистров (FC 23): 121

Диапазон записи данных: в этом поле отображается диапазон данных записи MODBUS, сконфигурированный для каждого запроса. Начальный адрес вместе с размером считанных данных приведет к диапазону записываемых данных для каждого запроса.

Диагностическая переменная: Диагностика запроса MODBUS, настроенная символическим отображением или прямым представлением, хранится в переменных типа T\_DIAG\_MODBUS\_RTU\_MAPPING\_1 для ведущих устройств и T\_DIAG\_MODBUS\_ETH\_CLIENT\_1 для клиентских устройств, а отображение путем прямого представления имеет 4 байта и 2 слова, которые описаны в Таблице 86 ("n" — значение, настроенное в поле %Q

## 5. КОНФИГУРАЦИЯ

Начальный адрес области диагностики).

Переменная прямого представл ения	Диагностика переменной T_DIAG_MODBUS _ETH_MAPPING_1.*	Размер	Описание
<b>Биты состояния связи:</b>			
%QX(n).0	byStatus. bCommIdle	BIT	Связь простаивает (ожидает выполнения).
%QX(n).1	byStatus. bCommExecuting	BIT	Активное общение.
%QX(n).2	byStatus. bCommPostponed	BIT	Связь отложена, так как достигнуто максимальное количество одновременных запросов. Отложенные коммуникации будут осуществляться в той же последовательности, в которой они были заказаны, чтобы избежать неопределенности. Время, проведенное в этом состоянии, не учитывается при определении тайм-аута. Биты bCommIdle и bCommExecuting ложны, когда бит bCommPostponed истинен.
%QX(n).3	byStatus. bCommDisabled	BIT	Связь отключена. В этом состоянии бит bCommIdle перезапускается.
%QX(n).4	byStatus. bCommOk	BIT	Прерванная ранее связь прошла успешно.
%QX(n).5	byStatus. bCommError	BIT	Связь прервана, ранее была ошибка. Проверьте код ошибки.
%QX(n).6	byStatus. bCommAborted	BIT	Ранее прерванная связь была прервана из-за сбоя подключения.
%QX(n).7	byStatus. bDiag_7_reserved	BIT	Зарезервировано.
<b>Код последней ошибки (включен, если bCommError = true):</b>			
%QB(n+1)	eLastErrorCode	MASTER_ERROR_CODE (BYTE)	Сообщает возможную причину последней ошибки в отображении MODBUS. Обратитесь к Таблице 109 для получения дополнительной информации.
<b>Последний код исключения, полученный мастером:</b>			
%QB(n+2)	eLastExceptionCode	MODBUS_EXCEPTION (BYTE)	NO_EXCEPTION (0) FUNCTION_NOT_SUPPORTED (1) MAPPING_NOT_FOUND (2) ILLEGAL_VALUE (3) ACCESS_DENIED (128)* MAPPING_DISABLED (129)* IGNORE_FRAME (255)*

Переменная прямого представл ения	Диагностика переменной T_DIAG_MODBUS _ETH_MAPPING_1.*	Размер	Описание
<b>Статистика общения:</b>			
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.

## 5. КОНФИГУРАЦИЯ

%QW(n+4)	wCommCounter	WORD	Счетчик связи прерывается с ошибками или без них. Пользователь может проверить изменение этого счетчика после завершения связи. При достижении значения 65535 счетчик возвращается к нулю.
%QW(n+6)	wCommErrorCounter	WORD	Счетчик связи завершен с ошибками. При достижении значения 65535 счетчик возвращается к нулю.

Таблица 108: Диагностика отношений с клиентами MODBUS

### Примечания:

**Коды исключений:** коды исключений, показанные в этом поле, являются значениями, возвращаемыми сервером. Определения кодов исключений 128, 129 и 255 действительны только для ведомых устройств Altus. Для ведомых устройств других производителей эти коды исключений могут иметь другое значение.

Отключающая переменная: поле для переменной, используемой для отключения запросов MODBUS, индивидуально настроенных в запросах. Запрос отключен, когда переменная, соответствующая запросу, равна 1, в противном случае запрос включен.

Код последней ошибки: Ниже приведены коды возможных ситуаций, вызывающих ошибку связи по протоколу MODBUS:

Код	Исчисляемый	Описание
1	ERR_EXCEPTION	Ответ находится в коде исключения (см. eLastExceptionCode= код исключения).
2	ERR_CRC	Ответ с неверным CRC.
3	ERR_ADDRESS	Адрес MODBUS не найден. Адрес, ответивший на запрос, отличается от ожидаемого.
4	ERR_FUNCTION	Неверный код функции. Код функции ответа отличался от ожидаемого.
5	ERR_FRAME_DATA_COUNT	Объем данных в ответе отличается от ожидаемого.
7	ERR_NOT_ECHO	Ответ не является эхом запроса (FC 05 и 06).
8	ERR_REFERENCE_NUMBER	Неверный ссылочный номер (FC 15 и 16).
9	ERR_INVALID_FRAME_SIZE	Ответ короче, чем ожидалось.
20	ERR_CONNECTION	Ошибка при установлении соединения.
21	ERR_SEND	Ошибка на этапе передачи.
22	ERR_RECEIVE	Ошибка на этапе приема.
40	ERR_CONNECTION_TIMEOUT	Время ожидания на уровне приложения во время соединения.
41	ERR_SEND_TIMEOUT	Тайм-аут на уровне приложения во время передачи.
42	ERR_RECEIVE_TIMEOUT	Время ожидания на уровне приложения при ожидании ответа.
43	ERR_CTS_OFF_TIMEOUT	Тайм-аут при ожидании CTS = false при передаче.
44	ERR_CTS_ON_TIMEOUT	Тайм-аут при ожидании CTS = true при передаче.
128	NO_ERROR	Нет ошибок с момента запуска.

Таблица 109: Коды ошибок отношений MODBUS

### ВНИМАНИЕ

В отличие от других задач приложения, при достижении отметки в отладке MainTask, задача экземпляра MODBUS Ethernet Client или любая другая задача MODBUS перестанет выполняться в момент попытки записи в область памяти. Это происходит для того, чтобы поддерживать согласованность данных областей памяти, пока MainTask не запущен.

### 5.7.7.2. Конфигурация клиента MODBUS Ethernet через прямое представление (%Q)

Чтобы настроить этот протокол с использованием прямого представления (%Q), необходимо выполнить следующие шаги:

Настройте общие параметры протокола MODBUS, такие как: время связи и переменные прямого представления (%Q) для получения диагностики.

Добавляйте и настраивайте устройства, устанавливая адрес, переменные прямого представления (%Q) для

## 5. КОНФИГУРАЦИЯ

отключения взаимосвязей, тайм-ауты связи и т. д.

Добавьте и настройте отношения MODBUS, указав тип данных и функцию MODBUS, тайм-ауты, переменные прямого представления (%Q) для получения диагностики отношения и другие для приема/записи данных, объем данных для передачи и опрос отношения. Описание каждой конфигурации приведено ниже в этом разделе.

### 5.7.7.2.1. Общие параметры клиента протокола MODBUS - конфигурация для прямого представления (%Q)

Общие параметры, находящиеся на главном экране конфигурации протокола MODBUS (Рисунок ниже), определяются как:

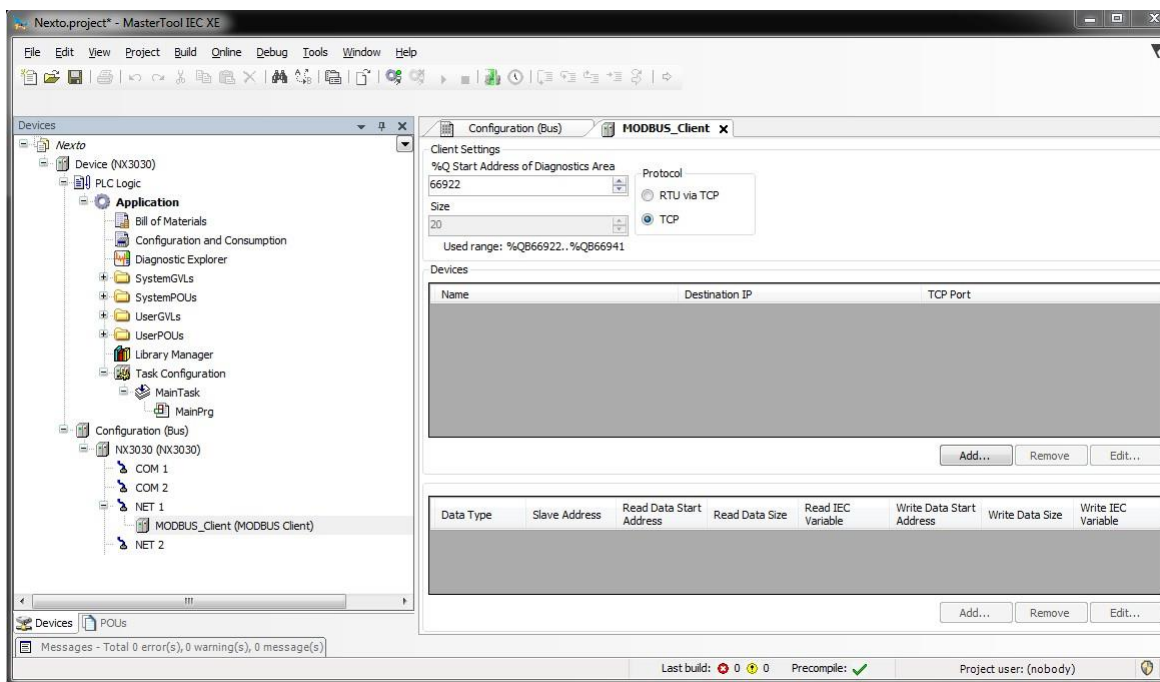


Рисунок 82: Экран настройки клиента MODBUS

Выбор протокола и переменные прямого представления (%Q) для диагностики:

Настройка	Описание	Значение по умолчанию	Варианты
<b>%Q Начальный адрес области диагностики</b>	Начальный адрес диагностических переменных	-	0 до 2147483628
<b>Размер</b>	Размер диагностики	20	Отключено для редактирования
<b>Протокол</b>	Выбор протокола	TCP	RTU через TCP TCP

Таблица 110: Настройки клиента MODBUS

#### Примечания:

%Q Начальный адрес области диагностики: это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦП, который можно найти в разделе Память.

Значение по умолчанию: значение по умолчанию не может быть определено для поля %Q Start Address of Diagnostic Area, поскольку создание экземпляра протокола может быть выполнено в любой момент разработки приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), которые все еще не используются.

Диагностика и команды MODBUS описаны в Таблице 101.

### 5.7.7.2.2. Конфигурация устройства — Конфигурация через прямое представление (%Q)

Конфигурация устройств, показанная на рисунке ниже, включает следующие параметры:

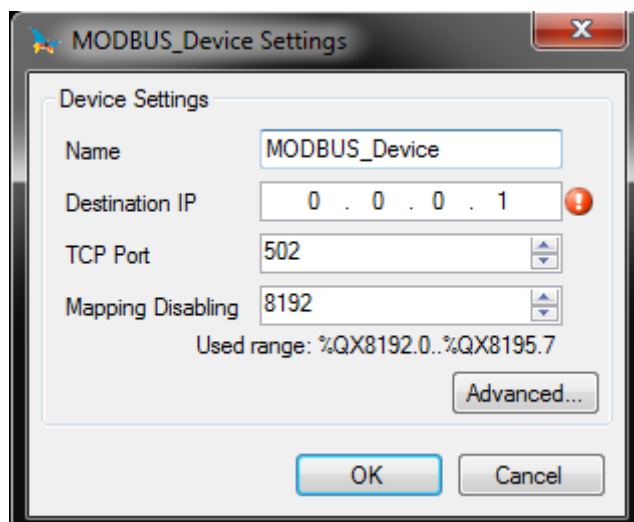


Рисунок 83: Настройка клиента MODBUS

Конфигурация	Описание	По умолчанию	Варианты
Наименование	Имя экземпляра	MODBUS_Device	Идентификатор согласно IEC 61131-3
IP-адрес назначения	Адрес IP сервера	0. 0. 0.1	1.0.0.1 до 223.255.255.255
ТСР-порт	ТСР-порт	502	2 до 65534
Отключение отображения	Исходный адрес, используемый для отключения отношений MODBUS	-	Любой адрес области %Q, ограниченный моделью ЦП

Таблица 111: Конфигурация устройства клиентов

**Примечания:**

Имя экземпляра: это поле является идентификатором устройства, которое проверяется в соответствии с IEC 61131-3, т. е. не допускает пробелов, специальных символов и начинается с цифры. Он ограничен 24 символами.

Порт ТСР: если к одному интерфейсу Ethernet добавлено несколько экземпляров протокола, для каждого экземпляра необходимо выбрать разные порты ТСР. Некоторые ТСР-порты, среди упомянутых выше возможностей, зарезервированы и поэтому не могут использоваться. См. раздел Таблица зарезервированных портов ТСР/UDP.

Отключение сопоставления: состоит из 32 бит, используется для индивидуального отключения 32 отношений MODBUS, указанных в пространстве сопоставлений устройств. Отношение отключено, когда соответствующий бит равен 1, в противном случае отображение включено. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦП, который можно найти в разделе Память.

Значение по умолчанию: factory Значение по умолчанию не может быть установлено для поля отключения сопоставления, поскольку создание экземпляра протокола может быть выполнено в любой момент разработки приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), которые все еще не используются.

Тайм-аут связи: настройки, присутствующие на кнопке Дополнительно... в ТСР-соединении, описаны в примечаниях к разделу Конфигурация устройства — Конфигурация через символическое сопоставление.

**5.7.7.2.3. Сопоставление Конфигурация – Конфигурация через прямое представление (%Q)**

Параметры отношений MODBUS, показанные на рисунках ниже, соответствуют параметрам, описанным в таблице ниже:

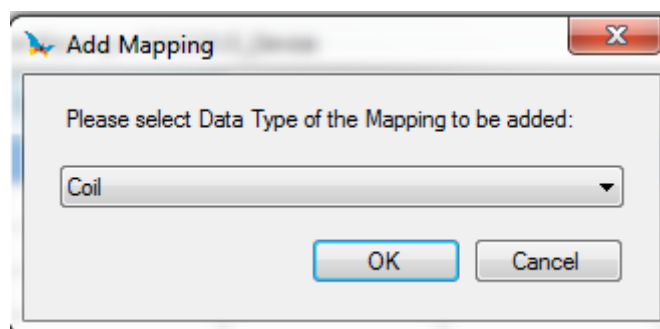


Рисунок 84: Тип данных MODBUS

Рисунок 85: Функция MODBUS

В приведенной ниже Таблице количество заводских настроек По умолчанию и значения столбца Варианты могут различаться в зависимости от типа данных и функции MODBUS (FC).

Конфигурация	Описание	Значение по умолчанию	Варианты
Функция	Тип функции MODBUS	Чтение	Чтение Запись Чтение/Запись Запись маски
Адрес подчиненного устройства	Адрес подчиненного устройства MODBUS	1	0 до 255
Опрос (мс)	Период связи (мс)	100	0 до 3600000
Отображение области диагностики	Начальный адрес диагностики интерфейса MODBUS	-	0 до 2147483640
Чтение начального адреса данных	Начальный адрес чтения	1	1 to 65536
Чтение размера данных	Дата MODBUS	-	Зависит от использованных функций
Чтение переменной IEC	Количество прочитанных данных MODBUS	-	0 до 2147483647
Запись начального адреса данных	Начальный адрес переменных чтения (%I)	1	1 до 65536
Запись размера данных	Начальный адрес записи данных MODBUS	-	Зависит от использованных функций
Запись переменной IEC	Количество записываемых данных MODBUS	-	0 до 2147483647
Маска записи переменных IEC	Начальный адрес переменных записи (%Q)	-	0 до 2147483644

Таблица 112: Сопоставление устройств

**Примечания:**

**Таблица сопоставлений устройств:** количество настроек и значений, описанных в столбце «Параметры», может варьироваться в зависимости от типа данных и функции MODBUS.

Адрес подчиненного устройства: как правило, адрес 0 используется, когда сервер является MODBUS RTU или MODBUS RTU через TCP-шлюз, и он же передает запрос всем сетевым устройствам. Когда используется адрес 0, клиент не ждет ответа и его использование служит только для написания команд. Более того, в соответствии со

## 5. КОНФИГУРАЦИЯ

стандартом MODBUS допустимый диапазон адресов для ведомых устройств составляет от 0 до 247, а адреса с 248 по 255 зарезервированы.

Опрос: этот параметр указывает, как часто должна выполняться связь, установленная для этой связи. По окончании связи будет выждено время, равное опросу configured, после чего в кратчайшие сроки будет осуществлена новая коммуникация.

**Картографирование диагностической области:** это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦП, который можно найти в разделе Память. Настроенная диагностика отношений MODBUS описана в таблице 86.

**Размер данных для чтения и записи:** сведения о размере данных, поддерживаемых каждой функцией, описаны в примечаниях к разделу Конфигурация запросов — Параметры символического отображения.

### ВНИМАНИЕ

При доступе к памяти данных связи между устройствами с разным порядком байтов (Little-Endian и Big-Endian) может произойти инверсия данных чтения/записи. В этом случае пользователь должен скорректировать данные в приложении.

**Чтение переменной IEC:** в случае, если типом данных MODBUS является катушка или состояние входа (бит), начальный адрес переменных IEC будет в формате %IX10.1. Однако, если типом данных MODBUS является регистр временного хранения или регистр ввода (16 бит), начальный адрес переменных IEC будет иметь формат %IW. Это поле ограничено объемом памяти адресуемых входных переменных (%I) каждого ЦП, что можно увидеть в разделе «Память».

**Запись переменной IEC:** если тип данных MODBUS — Coil (bit), начальный адрес переменных IEC будет в формате %QX10.1. Однако, если типом данных MODBUS является регистр временного хранения (16 бит), начальный адрес переменных IEC будет иметь формат %QW. Это поле ограничено размером памяти адресуемых выходных переменных (%Q) каждого ЦП, что можно увидеть в разделе «Память».

**Маска записи переменных IEC:** функция регистра записи маски (FC 22) использует логику между уже записанным значением и двумя словами, которые указаны в этом поле, используя %QW(0) для маски AND и %QW(2) для операционная маска; позволяя пользователю обрабатывать слово. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе Память.

**Значение по умолчанию:** заводское значение по умолчанию не может быть установлено для полей «Область диагностики сопоставления», «Чтение переменной IEC», «Запись переменной IEC» и «Маска записи переменных IEC», поскольку создание связи может быть выполнено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), которые все еще не используются. Заводские настройки по умолчанию не могут быть установлены в полях Размер данных для чтения/записи, так как они будут различаться в зависимости от выбранного типа данных MODBUS.

### ВНИМАНИЕ

В отличие от других задач приложения, при достижении отметки в отладке MainTask, задача экземпляра MODBUS Ethernet Client или любая другая задача MODBUS перестанет выполняться в момент попытки записи в область памяти. Это происходит для того, чтобы поддерживать согласованность данных областей памяти, пока MainTask не закончит.

#### 5.7.7.3. Запуск отношений клиента MODBUS в циклической форме

Для запуска отношения клиента MODBUS в циклическом виде предлагается следующий метод, который может быть реализован простым способом в прикладной программе пользователя:

Определить максимальное время опроса отношений; Держите связь обычно отключенной;

Включите отношение в тот момент, когда требуется выполнение;

Дождитесь подтверждения об окончании выполнения отношения и в этот момент снова отключите его.

#### 5.7.8. Ethernet-сервер MODBUS

Этот протокол доступен для всех ЦП серии Nexto на его каналах Ethernet. При выборе этой опции в MasterTool IEC XE ЦП становится коммуникационным сервером MODBUS, обеспечивая соединение с клиентскими устройствами MODBUS. Этот протокол доступен только тогда, когда ЦП находится в режиме выполнения (Run Mode).

Есть два способа конфигунок этого протокола. В первом используется прямое представление (%Q), в котором переменные определяются вашим адресом. Второй — через символическое отображение, где переменные определяются вашим именем.

Процедура вставки экземпляра протокола подробно описана в Руководстве пользователя MasterTool IEC XE — MU299609.

##### 5.7.8.1. Конфигурация протокола Ethernet сервера MODBUS для символического отображения

## 5. КОНФИГУРАЦИЯ

Чтобы подтвердить этот протокол с помощью Symbolic Mappings, необходимо выполнить следующие шаги:

Настройте общие параметры протокола сервера MODBUS, такие как: TCP-порт, выбор протокола, IP-фильтры для чтения и записи (доступны по кнопке Конфигурация фильтров) и время связи (доступно по кнопке Расширенные настройки сервера).

Добавьте и подтвердите сопоставления MODBUS, указав имя переменной, тип данных, начальный адрес данных и размер данных. Описание каждой Конфигурации приведено далее в этом разделе.

### 5.7.8.1.1. Общие параметры протокола сервера MODBUS – Конфигурация с помощью символического отображения

Общие параметры, находящиеся на начальном экране протокола MODBUS (Рисунок ниже), определяются как.

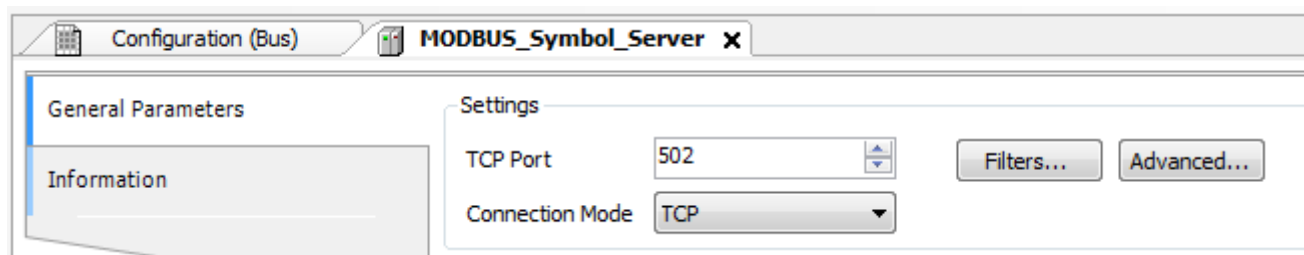


Рисунок 86: Экран конфигурации общих параметров сервера MODBUS

Конфигурация	Описание	По умолчанию	Варианты
Порт TCP	Порт TCP	502	от 2 до 65534
Режим подключения	Выбор протокола	TCP	RTU через TCP TCP

Таблица 113: Общие конфигурации сервера MODBUS

#### Примечания:

**Порт TCP:** если в один интерфейс Ethernet добавлено несколько экземпляров протокола, для каждого экземпляра необходимо выбрать разные TCP-порты. Некоторые TCP-порты, среди упомянутых выше возможностей, зарезервированы и поэтому не могут использоваться. См. раздел Таблица зарезервированных портов TCP/UDP.

Параметры, представленные на кнопке «Фильтры...», описанные в Таблице ниже, относятся к коммуникационным фильтрам TCP:

Конфигурация	Описание	Значение по умолчанию	Варианты
Адрес фильтра записи IP	Указывает диапазон IP-адресов с доступом для записи в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	0.0.0.0 до 255.255.255.255
Маска фильтра записи	Указывает маску подсети в сочетании с параметром IP-фильтра для записи.	0.0.0.0	0.0.0.0 до 255.255.255.255
Адрес фильтра чтения IP	Указывает диапазон IP-адресов с доступом для чтения в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	0.0.0.0 до 255.255.255.255
Маска фильтра чтения	Указывает маску подсети в сочетании с параметром IP-фильтра для чтения.	0.0.0.0	0.0.0.0 до 255.255.255.255

Таблица 114: IP фильтры

#### Примечания:

**Фильтры:** фильтры используются для установления диапазона IP-адресов, которые имеют доступ на запись или чтение к отношениям MODBUS, которые индивидуально проверяются. Критерии разрешения выполняются с помощью логической операции И между маской фильтра записи и IP-адресом клиента. Если результат совпадает с IP-адресом фильтра записи, клиент имеет право на запись. Например, если IP-адрес фильтра записи = 192.168.15.0 и

## 5. КОНФИГУРАЦИЯ

маска фильтра записи = 255.255.255.0, то только клиенты с IP-адресом = 192.168.15.x будут иметь право. Та же процедура применяется в параметрах фильтра чтения для определения прав чтения.

Время связи протокола сервера MODBUS, указанное с помощью кнопки «Дополнительно...» экрана «Конфигурация», разделено на: «Цикл задачи» и «Тайм-аут бездействия соединения».

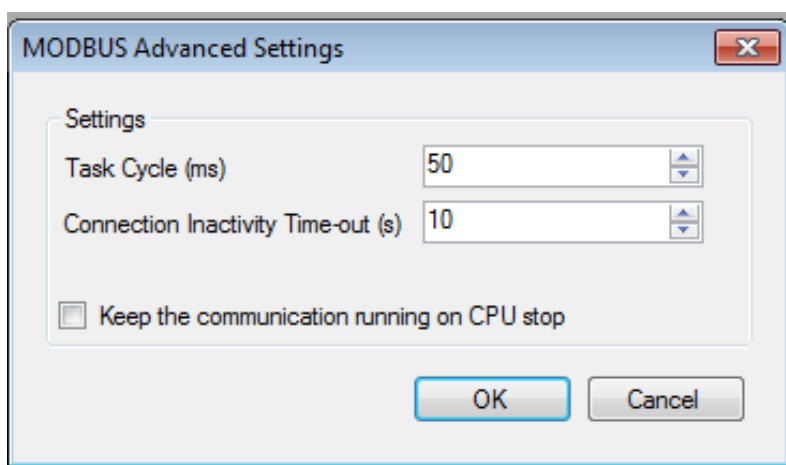


Рисунок 87: Экран конфигурации дополнительных параметров сервера MODBUS

Конфигурация	Описание	Значение по умолчанию	Варианты
Цикл задачи (мс)	Время выполнения экземпляра в цикле, без учета собственного времени выполнения	50	5 до 100
Тайм-аут бездействия соединения (с)	Максимальное время простоя между клиентом и сервером, прежде чем соединение будет закрыто сервером	10	1 до 3600
Сохраняйте обмен данными при остановке ЦП.	Разрешить ведомому символу MODBUS работать, когда ЦП находится в состоянии STOP или после точки останова.	Не отмеченное	Отмеченное или не отмеченное

Таблица 115: Расширенные конфигурации сервера MODBUS

### Примечания:

**Цикл задач:** пользователь должен быть осторожным при изменении этого параметра, так как он напрямую влияет на время ответа, объем данных для сканирования и, главным образом, на баланс ресурсов ЦП между коммуникациями и другими задачами.

**Тайм-аут бездействия соединения:** этот параметр был создан для того, чтобы избежать достижения максимального количества TCP-соединений, предполагая, что неактивные соединения остаются открытыми из-за самых разных проблем. Он указывает, как долго соединение (клиент или сервер) может оставаться открытым без использования (без обмена коммуникационными сообщениями). Если указанное время не достигнуто, соединение закрывается, освобождая вход в соединение Таблица.

### 5.7.8.1.2. Диагностика сервера MODBUS – Конфигурация с помощью символьного отображения

Диагностика и команды протокола сервера MODBUS, сконфигурированные либо символьным отображением, либо прямым представлением, хранятся в переменных типа T\_DIAG\_MODBUS\_ETH\_SERVER\_1, а отображение прямым представлением — в 4-байтовом и 8-словном формате, которые описаны в Таблице ниже (n — это значение, настроенное в поле %Q Start Address of Diagnostic Area):

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_SERVER_1.*	Размер	Описание
Диагностические биты:			

## 5. КОНФИГУРАЦИЯ

%QX(n).0	tDiag. bRunning	BIT	Запуск сервера.
%QX(n).1	tDiag. bNotRunning	BIT	Сервер не работает (см. бит bInterruptedByCommand).
%QX(n).2	tDiag. bInterruptedByCommand	BIT	Бит bNotRunning был включен, потому что сервер был прерван пользователем через командный бит.
%QX(n).3	tDiag. bConfigFailure	BIT	Прекращена диагностика.
%QX(n).4	tDiag. bRXFailure	BIT	Прекращена диагностика.
%QX(n).5	tDiag. bTXFailure	BIT	Прекращена диагностика.
%QX(n).6	tDiag. bModuleFailure	BIT	Прекращена диагностика.
%QX(n).7	tDiag. bDiag_7_reserved	BIT	Зарезервировано.
%QB(n+1)	byDiag_1_reserved	BYTE	Зарезервировано.
<b>Командные биты, автоматически перезапускаются:</b>			
%QX(n+2).0	tCommand. bStop	BIT	Остановка сервера
%QX(n+2).1	tCommand. bRestart	BIT	Перезапуск сервера
%QX(n+2).2	tCommand. bResetCounter	BIT	Сбросить диагностическую статистику (счетчики).
%QX(n+2).3	tCommand. bDiag_19_reserved	BIT	Зарезервировано.
%QX(n+2).4	tCommand. bDiag_20_reserved	BIT	Зарезервировано.
%QX(n+2).5	tCommand. bDiag_21_reserved	BIT	Зарезервировано.
%QX(n+2).6	tCommand. bDiag_22_reserved	BIT	Зарезервировано.
%QX(n+2).7	tCommand. bDiag_23_reserved	BIT	Зарезервировано.
%QB(n+3)	byDiag_3_reserved	BYTE	Зарезервировано.
<b>Статистика общения:</b>			
%QW(n+4)	tStat. wActiveConnections	WORD	Количество установленных соединений между клиентом и сервером (от 0 до 64).
%QW(n+6)	tStat. wTimeoutClosedConnections	WORD	Счетчик подключений между клиентом и сервером, прерванных после периода бездействия - тайм-аут (от 0 до 65535).
%QW(n+8)	tStat. wClientClosedConnections	WORD	Счетчик подключений прерван по запросу клиента (от 0 до 65535).

Переменная прямого представления	Диагностическая переменная T_DIAG_MODBUS_ETH_SERVER_1.*	Размер	Описание
%QW(n+10)	tStat. wRXFrames	WORD	Счетчик кадров Ethernet, полученных сервером. Кадр Ethernet может содержать более одного запроса (от 0 до 65535).
%QW(n+12)	tStat. wRXRequests	WORD	Запросы, полученные счетчиком сервера и нормально отвеченные (от 0 до 65535).
%QW(n+14)	tStat. wTXExceptionResponses	WORD	Запросы, полученные счетчиком сервера, и ответы на них с кодами исключений (от 0 до 65535). Коды исключений перечислены ниже: 1: код функции (FC) разрешен, но не поддерживается. 2: отношение не найдено в этих данных MOD-BUS. 3: недопустимое значение адреса. 128: мастер/клиент не имеет права ни читать, ни писать. 129: связь MODBUS отключена.
%QW(n+16)	tStat. wRXIllegalRequests	WORD	Счетчик нелегальных запросов (от 0 до 65535).
%QW(n+18)	tStat. wDiag_18_Reserved	WORD	Зарезервировано.

Таблица 116: Диагностика сервера MODBUS

**Примечание:**

**Счетчики:** все счетчики диагностики сервера MODBUS Ethernet обнуляются при превышении предельного значения 65535.

## 5.7.8.1.3. Конфигурация отображения – Конфигурация через символическое отображение

Отношения MODBUS Конфигурация, показанная на рисунке ниже, соответствует параметрам, описанным в таблице ниже:

Mappings

	Value Variable	Data Type	Data Start Address	Absolute Data Start Address	Data Size	Data Range
*						

Рисунок 88: Экран сопоставления данных сервера MODBUS

Конфигурация	Описание	Значение по умолчанию	Варианты
Переменная значения	Символьное имя переменной	-	Имя переменной, объявленной в программе или GVL
Тип данных	Тип данных MODBUS	-	Катушка Входной регистр хранения состояния Входной регистр
Начальный адрес данных	Начальный адрес данных MODBUS	-	1 до 65536

## 5. КОНФИГУРАЦИЯ

Абсолютный начальный адрес данных	Начальный адрес абсолютных данных Modbus как его тип	-	-
Размер данных	Размер данных MODBUS	-	1 до 65536
Диапазон данных	Настроен адрес диапазона данны	-	-

Таблица 117: Конфигурация отображения MODBUS Ethernet

### Примечания:

**Переменная значения:** это поле используется для указания символической переменной в отношении MODBUS.  
**Тип данных:** это поле используется для указания типа данных, используемых в отношении MODBUS.

**Начальный адрес данных:** начальный адрес данных отношения MODBUS.

**Абсолютный начальный адрес данных:** абсолютный начальный адрес данных MODBUS в соответствии с их типом. Например, регистр хранения с адресом 5 имеет абсолютный адрес 400005. Это поле доступно только для чтения и доступно для помощи в конфигурации клиента/главного устройства MODBUS, которое будет обмениваться данными с этим устройством. Значения зависят от базового адреса (смещения) каждого типа данных и разрешенного адреса MODBUS для каждого типа данных.

**Размер данных:** значение размера данных устанавливает максимальный объем данных, к которым отношение MODBUS может получить доступ с начального адреса. Таким образом, чтобы прочитать Продолжающийся диапазон адресов, необходимо, чтобы все адреса были объявлены в одном отношении. Это поле зависит от настроенного типа данных MODBUS.

**Диапазон данных:** это поле доступно только для чтения и сообщает о диапазоне адресов, который используется этим сопоставлением. Он формируется суммой полей Data Start Address и Data Size. Не может быть наложен диапазон с другими сопоставлениями того же типа данных.

### 5.7.8.2. Конфигурация протокола Ethernet сервера MODBUS через прямое представление (%Q)

#### ВНИМАНИЕ

В отличие от других задач приложения, при достижении отметки в отладке MainTask, задача экземпляра MODBUS Ethernet Server или любая другая задача MODBUS перестанет выполняться в момент попытки записи в область памяти. Это происходит для того, чтобы поддерживать согласованность данных областей памяти, пока MainTask не закончит

Чтобы подтвердить этот протокол с помощью прямого представления (%Q), пользователь должен выполнить следующие шаги:

Согласуйте общие параметры протокола сервера MODBUS, такие как: время связи, адрес и переменные прямого представления (%Q) для получения диагностических и управляющих отношений.

Добавьте и подтвердите отношения MODBUS, указав тип данных MODBUS, переменные прямого представления (%Q) для получения/записи данных и количество данных, которые необходимо сообщить.

Описание каждой Конфигурации приведено ниже в этом разделе.

#### 5.7.8.2.1. Общие параметры протокола сервера MODBUS – Конфигурация через прямое представление (%Q)

Общие параметры, находящиеся на главном экране протокола MODBUS Конфигурация (Рисунок ниже), определяются как:

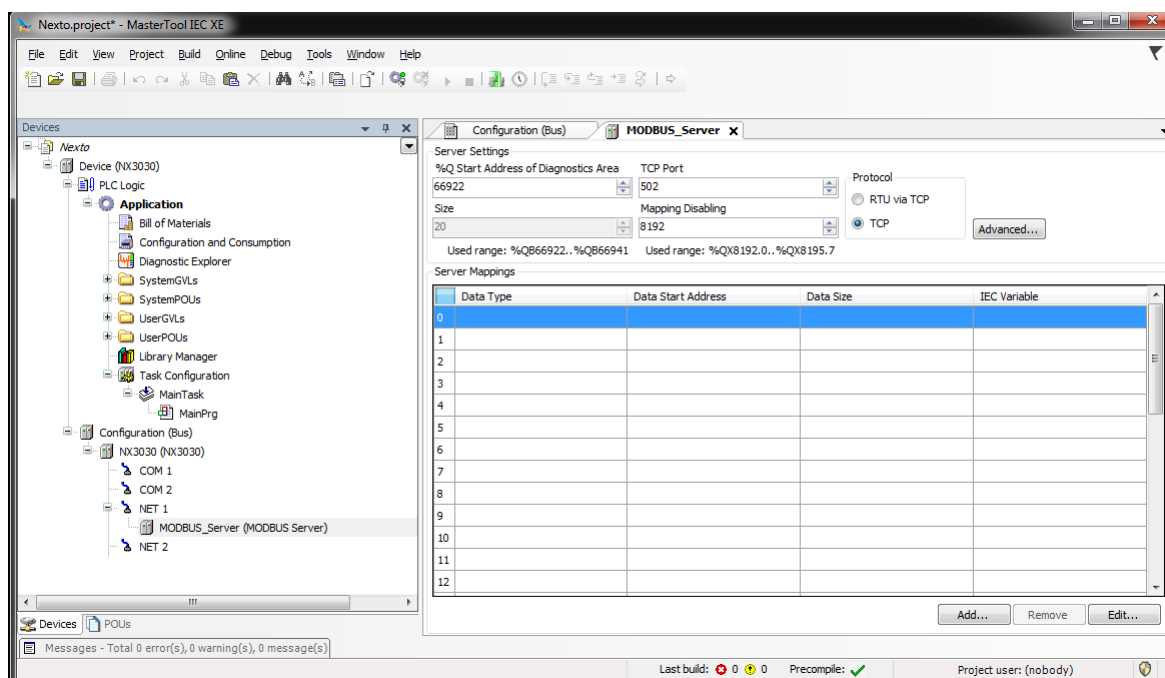


Рисунок 89: Экран настройки сервера MODBUS

ТПорт CP, протокол и переменные прямого представления (%Q) для управления взаимосвязями и диагностики:

Конфигурация	Описание	Значение по умолчанию	Варианты
%Q Начальный адрес области диагностики	Начальный адрес диагностических переменных	-	0 до 2147483628
Размер	Размер диагностики	20	Отключено для редактирования
TCP-порт	TCP-порт	502	2 до 65534
Отключение отображения	Начальный адрес, используемый для отключения отношений MODBUS	-	0 до 2147483644
Протокол	Выбор протокола	TCP	RTU через TCP TCP

Таблица 118: Настройки для контроля взаимосвязей и диагностики

#### Примечания:

**%Q Начальный адрес области диагностики:** это поле ограничено размером адресуемой памяти выходных переменных (%Q) на ЦП, который можно найти в разделе Память.

Порт TCP: если к одному интерфейсу Ethernet добавлено несколько экземпляров протокола, для каждого экземпляра необходимо выбрать разные порты TCP. Некоторые TCP-порты, среди упомянутых выше возможностей, зарезервированы и поэтому не могут использоваться. См. раздел Таблица зарезервированных портов TCP/UDP.

Отключение сопоставления: состоит из 32 битов, используемых для индивидуального отключения 32 отношений MODBUS, указанных в пространстве сопоставлений серверов. Отношение отключено, когда соответствующий бит равен 1, в противном случае отображение включено. Это поле ограничено размером адресуемой памяти выходных переменных (%Q) каждого процессора, который можно найти в разделе «Память».

Значение по умолчанию: заводское значение по умолчанию не может быть установлено для полей %Q Start Address of Diagnostics Area и Mapping Disabled, поскольку создание экземпляра протокола может быть приостановлено в любое время при разработке приложения. Программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления (%Q), еще не использованных.

Время связи протокола сервера MODBUS, указанное на кнопке Дополнительно... экрана Конфигурация, разделено на: Цикл задачи (мс) и Тайм-аут бездействия соединения (с). Дальнейшие подробности описаны в разделе «Общие параметры протокола сервера MODBUS — Конфигурация через символическое отображение».

Диагностика и команды MODBUS описаны в Таблице 116.

## 5. КОНФИГУРАЦИЯ

### 5.7.8.2.2. Сопоставление Конфигурация – Конфигурация через прямое представление (%Q)

Параметры отношений MODBUS, показанные на рисунках ниже, соответствуют параметрам, описанным в таблице ниже:

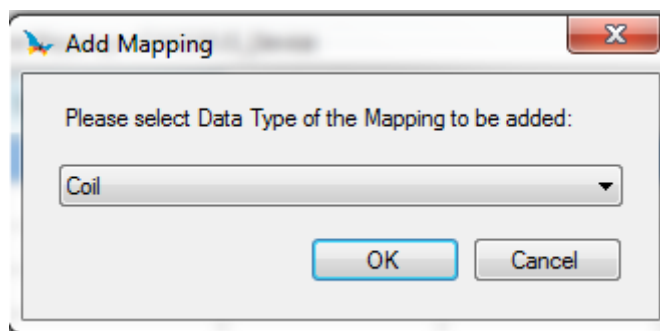


Рисунок 90: Тип данных MODBUS

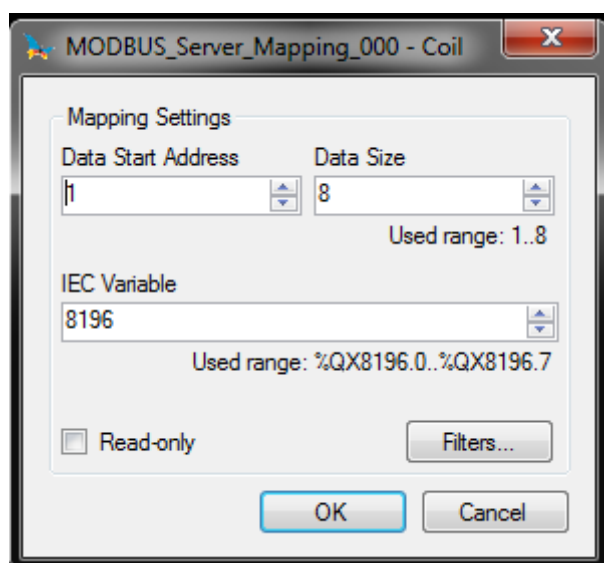


Рисунок 91: Функция сервера MODBUS

Конфигурация	Описание	По умолчанию	Варианты
Типд данных	Тип данных MODBUS	Катушка	Катушка (1 бит) Регистр хранения (16 бит) Состояние входа (1 бит) Входной регистр (16 бит)
Адрес начала данных	Начальный адрес данных MODBUS	1	1 до 65536
Размер данных	Количество данных MODBUS	8	от 1 до 65536 (регистр хранения и регистр ввода) от 8 до 65536 (состояние катушки и входа)
Переменная МЭК	Начальный адрес переменных (%Q)	-	0 до 2147483647
Только чтение	Разрешить только чтение	Неполноценный	Включено или отключено

Таблица 119: Сопоставления серверов

#### Примечания:

**Варианты:** значения, записанные в столбце Варианты, могут варьироваться в зависимости от настроенных данных MODBUS.

Размер данных: значение размера данных устанавливает максимальный объем данных, к которым отношение MODBUS может получить доступ с начального адреса. Таким образом, чтобы прочитать Продолжающийся диапазон адресов, необходимо, чтобы все адреса были объявлены в одном отношении. Это поле изменяется в

## 5. КОНФИГУРАЦИЯ

зависимости от установленного типа данных MODBUS, то есть, когда выбрано значение Coil или Input Status, поле Data Size должно быть числом, кратным 8. Также важно позаботиться о том, чтобы максимальное значение не превышало адресуемое значение. размер выходной памяти и атрибутивные значения уже не используются во время приложения

**Переменная МЭК:** в случае, если типом данных MODBUS является катушка или состояние входа (бит), начальный адрес переменных IEC будет иметь формат, например, %QX10.1. Однако, если типом данных MODBUS является регистр хранения или регистр ввода (16 бит), начальный адрес переменных IEC будет иметь формат %QW. Это поле ограничено размером памяти адресуемых выходных переменных (%Q) каждого ЦП, что

### ВНИМАНИЕ

При доступе к памяти данных связи между устройствами с разным порядком байтов (Little-Endian и Big-Endian) может произойти инверсия данных чтения/записи. В этом случае пользователь должен скорректировать данные в приложении.

можно увидеть в разделе «Память».

**Только чтение:** когда он включен, он позволяет ведущему коммуникационному устройству только считывать переменные данные. Это не позволяет писать.

Эта опция действительна только для функций записи.

По умолчанию: значение По умолчанию не может быть определено для поля Переменная IEC, так как создание экземпляра протокола может быть выполнено в любой момент разработки приложения, в результате чего программное обеспечение MasterTool IEC XE само выделяет значение из диапазона выходных переменных прямого представления ( %Q) до сих пор не используется. Значение По умолчанию не может быть определено для поля Размер данных, поскольку оно будет варьироваться в зависимости от выбранного типа данных MODBUS.

Параметры, представленные на кнопке «Фильтры...», описанные в Таблице ниже, относятся к коммуникационным фильтрам TCP:

Конфигурация	Описание	Значение по умолчанию	Варианты
Адрес фильтра записи IP	Указывает диапазон IP-адресов с доступом для записи в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	0.0.0.0 до 255.255.255.255
Маска фильтра записи	Указывает маску подсети в сочетании с параметром IP-фильтра для записи.	0.0.0.0	0.0.0.0 до 255.255.255.255
Адрес фильтра чтения IP	Указывает диапазон IP-адресов с доступом для чтения в переменных, объявленных в интерфейсе MODBUS.	0.0.0.0	0.0.0.0 до 255.255.255.255

Конфигурация	Описание	По умолчанию Value	Варианты
Маска фильтра чтения	Указывает маску подсети в сочетании с параметром IP-фильтра для чтения.	0.0.0.0	0.0.0.0 до 255.255.255.255

Таблица 120: IP фильтры

**Примечание:**

**Фильтры:** фильтры используются для установления диапазона IP-адресов, которые имеют доступ на запись или чтение к отношениям MODBUS, которые индивидуально проверяются. Критерии разрешения выполняются с помощью логической операции И между маской фильтра записи и IP-адресом клиента. Если результат совпадает с IP-адресом фильтра записи, клиент имеет право на запись. Например, если IP-адрес фильтра записи = 192.168.15.0 и маска фильтра записи = 255.255.255.0, то только клиенты с IP-адресом = 192.168.15.x будут иметь право. Та же процедура применяется в параметрах фильтра чтения для определения прав чтения.

В ранее определенных соотношениях максимальный размер данных MODBUS может быть 65536 (максимальное значение configured в поле Data Size). Однако запрос, поступающий на сервер MODBUS Ethernet, должен адресовать подгруппу этого сопоставления, и эта группа должна иметь не более размера данных, зависящего от функционального кода, который определен ниже:

Читать катушки (FC 1): 2000

Чтение состояния входа (FC 2): 2000

Чтение регистров хранения (FC 3): 125

Чтение входных регистров (FC 4): 125

Запись одиночной катушки (FC 5): 1

Запись одного регистра хранения (FC 6): 1 Force Multiple Coils (FC 15): 1968

Регистры хранения записи (FC 16): 123

Регистр записи маски (FC 22): 1 регистр хранения чтения/записи (FC 23):

- Чтение: 121

- Запись: 121

**ВНИМАНИЕ**

В отличие от других прикладных задач, при достижении метки очистки в MainTask задача экземпляра сервера Ethernet MODBUS и любая другая задача MODBUS перестанут выполняться в тот момент, когда она попытается выполнить запись в область памяти. Это происходит для того, чтобы сохранить согласованность данных областей памяти, когда основная задача не запущена.

**5.7.9. Сервер Доступа к коммуникационным данным открытой платформы**

Связь с процессорами серии Nexto возможна с использованием технологии OPC DA (Доступ к коммуникационным данным открытой платформы). Эта открытая коммуникационная платформа была разработана как стандарт промышленных коммуникаций. Основанный на клиент-серверной архитектуре, он предлагает ряд преимуществ при разработке проектов и связи с системами автоматизации.

Очень распространенной аналогией для описания технологии OPC DA является принтер. При правильном подключении компьютеру требуется драйвер для взаимодействия с оборудованием. Точно так же OPC помогает с интерфейсом между системой контроля и полевыми данными на ПЛК.

Когда дело доходит до разработки проекта, для установления связи и обмена информацией между системами чрезвычайно просто использовать технологию OPC DA. Используя другие драйверы, на основе адресов необходимо создавать Таблицы, чтобы связать теги из системы наблюдения с переменными из программируемого контроллера. Когда области данных изменяются в ходе проекта, необходимо переназначить переменные и создать новые таблицы с отношениями между информацией на ПЛК с системой диспетчерского управления и сбора данных (SCADA).

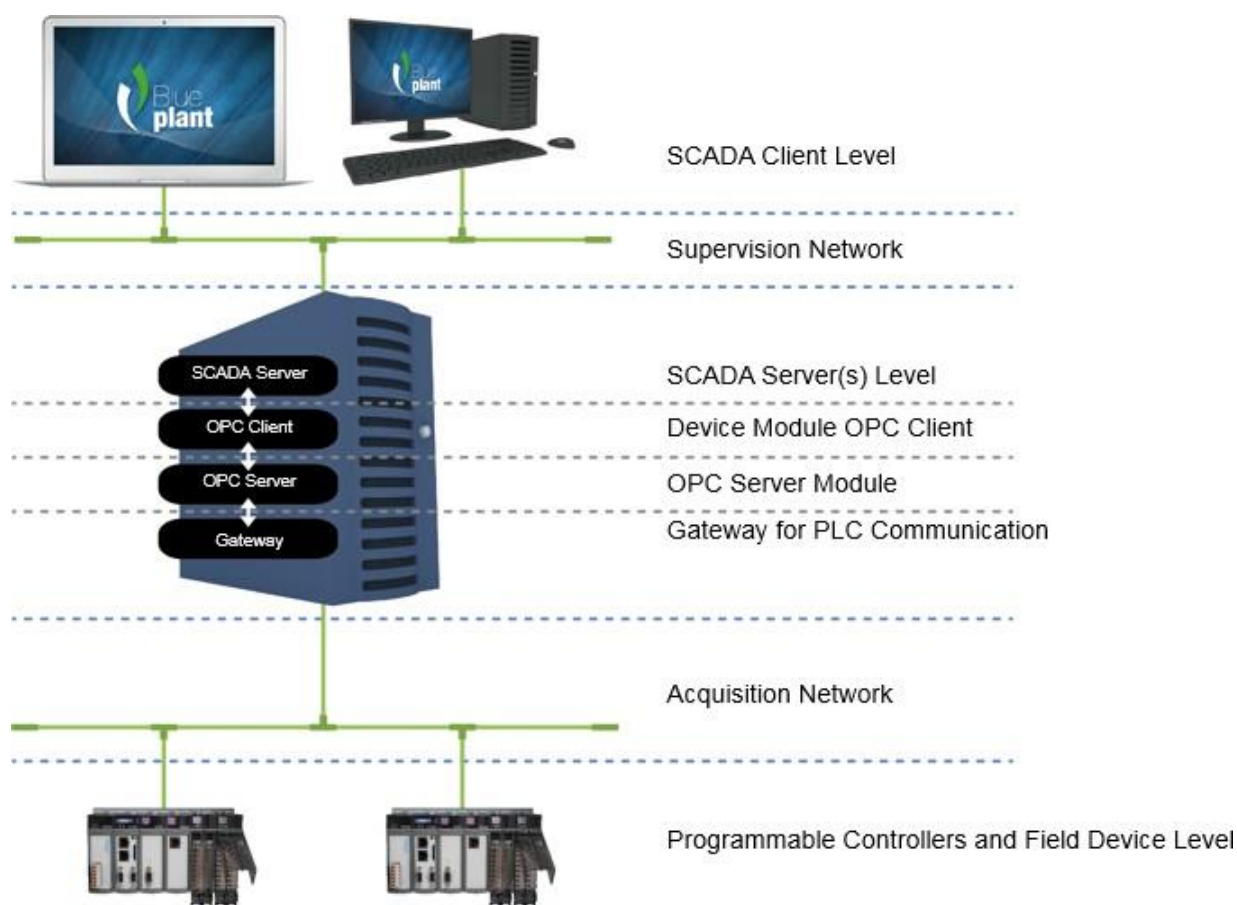


Рисунок 92: Архитектура Доступа к коммуникационным данным открытой платформы

На рисунке выше показана архитектура для связи системы SCADA и ПЛК в проектах автоматизации. Все роли, присутствующие в сообщении, явно указаны на этом рисунке независимо от оборудования, на котором оно выполняется, поскольку они могут выполняться на одном и том же оборудовании или на разных. Каждая из ролей этой архитектуры описана в Таблице ниже.

Роль	Описание
Уровень программируемых контроллеров и полевых устройств	В полевых устройствах и ПЛК хранится информация о рабочем состоянии и управлении предприятием. Система SCADA получает доступ к информации об этих устройствах и сохраняет ее на сервере SCADA, чтобы клиенты SCADA могли обращаться к ней во время работы предприятия.
Сеть приобретения	Сеть сбора данных — это место, куда передаются запросы на данные, собранные полевыми устройствами, для запроса данных, собранных с полевых устройств.
Шлюз для связи с ПЛК	Шлюз обеспечивает связь между сервером OPC DA и ПЛК серии Nexto. Шлюз в той же подсети ПЛК всегда необходим, как описано в главе «Настройки связи» руководства пользователя MasterTool IEC XE — MU299609.
Модуль сервера OPC	Сервер OPC DA — это модуль, отвечающий за получение запросов OPC DA и преобразование их в связь с полевыми устройствами.

Роль	Описание
Модуль устройства OPC-клиент	Модуль клиентского устройства OPC отвечает за запросы к серверу OPC DA с использованием протокола OPC DA. Собранные данные сохраняются в базе данных SCADA Server.
Уровень сервера SCADA	Сервер SCADA отвечает за подключение к различным коммуникационным устройствам и хранение собранных ими данных в базе данных, чтобы клиенты SCADA могли обращаться к ним.
Сеть наблюдения	Сеть наблюдения — это сеть, через которую клиенты SCADA подключаются к серверам SCADA. В топологии, в которой нет нескольких Клиентов или где Сервер и Клиент установлены на одном и том же оборудовании, такой сети не существует.
Уровень клиента SCADA	Клиенты SCADA несут ответственность за запрос к серверам SCADA необходимых данных для отображения на экране, где выполняется работа установки. После этого можно выполнять чтение и запись данных, хранящихся в базе данных сервера SCADA.

Таблица 121: Описание ролей в архитектуре сервера OPC DA

Связь между тегами в системе контроля и данными процесса в переменных контроллера полностью прозрачна. Это означает, что при изменении областей данных в ходе разработки проекта нет необходимости переделывать отношения между информацией в ПЛК и SCADA, просто используйте новую переменную, предоставленную ПЛК в системах, которые запрашивают эти данные.

Использование OPC обеспечивает большую производительность и возможность подключения к системам SCADA. Это способствует сокращению времени разработки приложений и стоимости обслуживания. Это даже позволяет вводить новые данные о связи в упрощенной форме и с большей гибкостью и функциональной совместимостью между системой автоматизации благодаря тому, что это открытый стандарт.

Установка сервера OPC DA выполняется вместе с установкой MasterTool IEC XE, а его настройки выполняются внутри инструмента. Стоит отметить, что OPC доступен только с локальным интерфейсом Ethernet процессоров Nexto. Модули расширения Ethernet не поддерживают эту функцию.

#### 5.7.9.1. Создание проекта для связи OPC DA

В отличие от связи с такими драйверами, как MODBUS и PROFIBUS DP, для установки связи OPC DA необходимо только правильно установить узел и указать, какие переменные будут использоваться в связи. Есть два способа указать, какие переменные проекта будут доступны на сервере OPC DA. В обоих случаях необходимо добавить в приложение объект Конфигурация символа, если его нет. Чтобы добавить его, щелкните правой кнопкой мыши объект Приложение и выберите параметр

#### ВНИМАНИЕ

Переменные, отображаемые в объектах IoConfig\_Globals, IoConfig\_Application\_Mappings и IoConfig\_Global\_Mappings, используются внутри для управления вводом/выводом и не должны использоваться пользователем.

#### ВНИМАНИЕ

В дополнение к переменным, объявленным в программных модулях языка SFC, также показаны некоторые неявно созданные переменные. Для каждого созданного шага создается переменная типа IecSfc.SFCStepType, в которой можно отслеживать состояния шага, а именно, активен он или нет, и время, в течение которого он активен, как в норме IEC 61131-1. Для каждого перехода создается переменная типа BOOL, которая определяет, является ли переход истинным или ложным. Эти переменные отображаются в объекте Symbol Configuration, которым может быть предоставлен доступ к OPC Client.

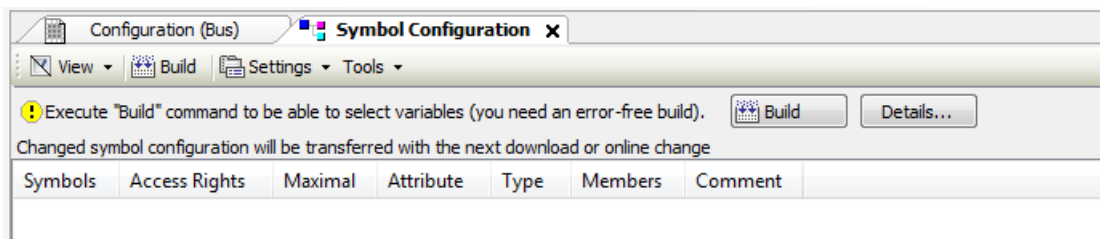


Рисунок 93: Объект конфигурации символа

В приведенной ниже Таблице представлены описания полей экрана объекта Конфигурация символа.

Сфера	Описание
<b>Символы</b>	Идентификатор переменной, который будет предоставлен серверу OPC DA.
<b>Права доступа</b>	Указывает, какой возможный уровень прав доступа находится в объявленном символе. Когда он не используется, этот столбец остается пустым, а уровень прав доступа максимальный. В противном случае уровень прав доступа можно изменить, щелкнув это поле. Возможные варианты: Только чтение Только запись Чтение и запись
<b>Максимальный</b>	Указывает максимальный уровень прав доступа, который можно присвоить переменной. Символы имеют те же значения, что и в правах доступа. Изменить его невозможно, и на это указывает наличие или отсутствие атрибута «символ».
<b>Атрибут</b>	Указывает, используется ли атрибут 'symbol' при объявлении переменной. Если этот столбец не используется, он остается пустым. Для случаев, в которых используется атрибут, поведение следующее: атрибут «символ» := «чтение» в столбце отображается атрибут «символ» := «запись» в столбце отображается атрибут 'symbol' := 'readwrite' столбец показывает
<b>Тип</b>	Тип данных объявленной переменной.
<b>Члены</b>	Если типом данных является Struct, в этом столбце активна кнопка. Нажатие на кнопку позволит выбрать, какие элементы этой структуры будут предоставлены серверу OPC DA.
<b>Комментарий</b>	Комментарий к переменной, вставляемый в POU или GVL, где была объявлена переменная. Чтобы отобразиться здесь как комментарий к переменной, комментарий должен быть введен за одну строку до переменной в редакторе в текстовом режиме или в столбце комментариев в табличном режиме.

Таблица 122: Описание полей экрана объекта конфигурации символа

При изменении настроек проекта, таких как добавление или удаление переменных, необходимо запустить команду Build, чтобы обновить список переменных. Эту команду необходимо выполнять до тех пор, пока сообщение на Рисунок 93 не исчезнет. После этого все доступные переменные в проекте, независимо от того, объявлены ли они в POU, GVL или диагностике, будут показаны здесь и могут быть выбраны. Выбранные переменные будут доступны на сервере OPC DA для доступа клиентов.

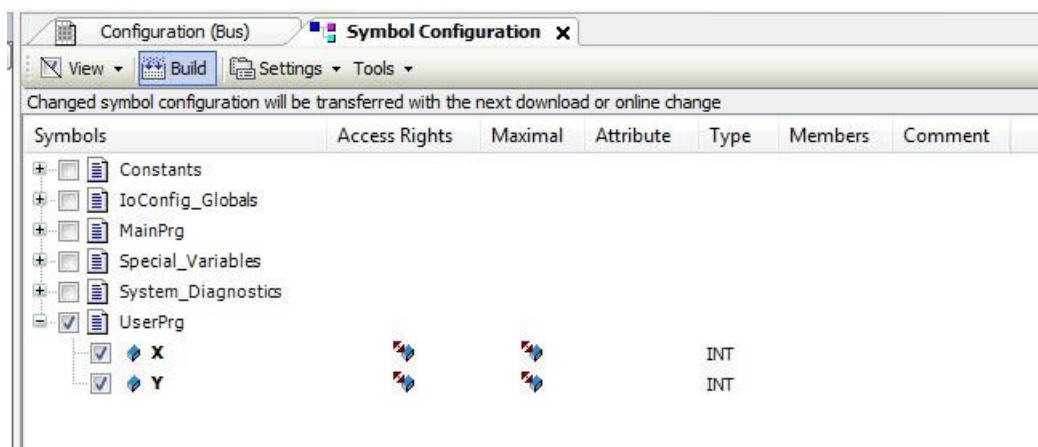


Рисунок 94: Конфигурация выбора переменных в символе

После этой процедуры проект должен быть загружен в ПЛК, чтобы переменные были доступны для связи с сервером OPC DA. Если экран объекта Конфигурация символа открыт и любая из выбранных переменных, POU или GVL изменена, ее имя будет выделено красным цветом. Это может произойти, когда переменная удалена или значение атрибута изменено.

Также возможно установить, какие переменные будут доступны на сервере OPC DA, с помощью атрибута, вставленного непосредственно в POU или GVL, где объявлены переменные. Когда в объявлении переменной присутствует атрибут 'symbol', и это может быть перед определением имени POU или GVL, или каждой переменной по отдельности, эти переменные отправляются непосредственно в объект Конфигурация символа с символом в атрибуте столбец. В этом случае необходимо перед загрузкой проекта в ЦП выполнить команду Build из объекта Конфигурация символа.

Допустимые синтаксисы для использования атрибута:

attribute 'symbol' := 'none' — когда значение атрибута равно 'none', переменные не будут доступны для сервера OPC DA и не будут отображаться на экране Конфигурация символа объекта.

attribute 'symbol' := 'read' - когда значение атрибута 'read', переменные будут доступны для OPC DA Server с правами доступа только для чтения.

атрибут 'symbol' := 'write' - когда значением атрибута является 'write', переменные будут доступны для сервера OPC DA с правом доступа только для записи.

атрибут 'symbol' := 'readwrite' — когда значением атрибута является 'readwrite', переменные будут доступны для сервера OPC DA с правами доступа на чтение и запись.

В следующем примере объявления переменных настройки переменных A и B позволяют серверу OPC DA обращаться к ним с доступом для чтения и записи. Однако к переменной C нельзя получить доступ, а к переменной D можно получить доступ только с правами доступа для чтения.

```
{attribute 'symbol' := 'readwrite'}
PROGRAM UserPrg

VAR

A: INT;

B: INT;

{attribute 'symbol' := 'none'}
C: INT;
```

Когда определяется переменная с типом, отличным от базовых типов, использование атрибута должно выполняться внутри объявления этого DUT, а не только в контексте, в котором создается переменная. Например, в случае экземпляра DUT внутри POU или GVL, у которого есть атрибут, это не повлияет на поведение элементов этого экземпляра DUT. Необходимо будет применить тот же уровень прав доступа к декларации DUT.

**ВНИМАНИЕ**

Конфигурации символов, которые будут предоставлены серверу OPC DA, хранятся внутри проекта ПЛК. Изменяя эти конфигурации, необходимо загрузить приложение в ПЛК, чтобы можно было получить доступ к этим переменным.

**ВНИМАНИЕ**

Когда переменная удаляется из проекта и загружается в ПЛК, снимая отметку с нее в объекте Symbol Configuration, переменная больше не может быть прочитана с помощью OPC-клиента. Если переменная снова добавляется в проект с тем же именем и тем же контекстом и вставляется в объект «Конфигурация символа», необходимо будет перезагрузить клиент OPC, чтобы обновить ссылку на адрес переменной, которая будет создана в другой памяти.

## 5.7.9.2. Настройка ПЛК на сервере OPC DA

Конфигурация ПЛК выполняется внутри MasterTool IEC XE с помощью опции, доступной в Online. Необходимо запустить MasterTool IEC XE от имени администратора.

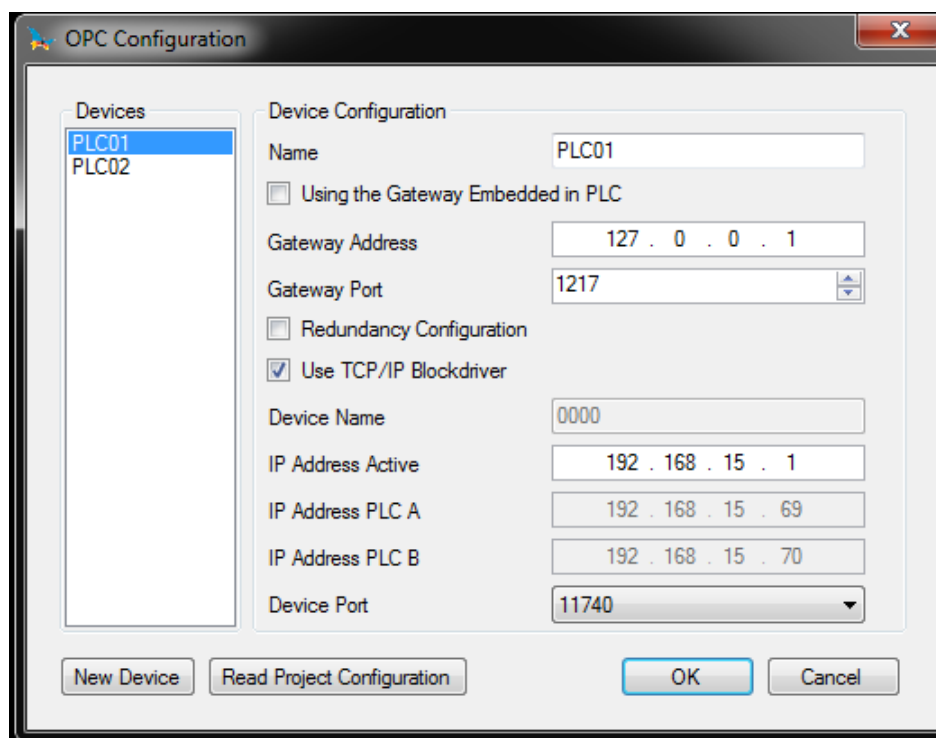


Рисунок 95: Настройки сервера OPC DA

Конфигурация шлюза — это тот же набор в шлюзе, который используется для связи между MasterTool IEC XE и ПЛК и описан в разделе «Настройки связи» в Руководстве пользователя MasterTool IEC XE — MU299609. Если используется Конфигурация localhost, Адрес шлюза должен быть заполнен 127.0.0.1. Эта Конфигурация необходима, поскольку сервер OPC DA использует тот же коммуникационный шлюз и тот же протокол, что и для связи между ПЛК и MasterTool IEC XE.

Существует вариант «Использование шлюза, встроенного в ПЛК», который можно выбрать, если необходимо использовать шлюз, встроенный в ПЛК. Эту опцию можно использовать для оптимизации связи, поскольку она предотвращает избыточный трафик через конкретную станцию, когда к одному и тому же ПЛК подключено более одной станции с OPC-клиентом.

Для настройки ПЛК существует два возможных типа Конфигурации, в зависимости от выбора флажка Use TCP/IP Blockdriver. Если опция не выбрана, поле Имя устройства должно быть заполнено именем ПЛК. Это имя отображается ПЛК, выбранным в качестве активного на экране настроек связи.

Другой вариант — использовать IP-адрес Ethernet-интерфейсов. В этом поле необходимо указать тот же адрес, что и на экранах Конфигурация. Кроме того, при использовании этого метода номер порта должен быть установлен на 11740. Подтверждение сохранит конфигурации сервера OPC DA.

Конфигурация устройства	Описание	Настройки по умолчанию	Варианты
Наименование	Описание ПЛК в файле конфигурации сервера OPC DA. Это поле может иметь любое имя, но в организационных целях рекомендуется использовать имя проекта, которое загружено в ПЛК.	'PLC01'	Это поле представляет собой STRING и принимает буквенно-цифровые (буквы и цифры) символы и символ «_». Запрещено инициировать STRING цифрами или символом «_». Он позволяет использовать до 49 символов.
Адрес шлюза	Адрес IP компьютера, на котором установлен сервер OPC DA, для случаев, когда все ПЛК находятся в одной подсети. Если какой-то ПЛК находится в другой подсети, необходимо указать шлюз, используемый в этой подсети.	127.0.0.1	0.0.0.0 до 255.255.255.255
Порт шлюза	TCP-порт для соединения со шлюзом.	1217	2 до 65534
Наименование устройства		'0000'	Это поле является STRING и может принимать любые символы, как это сделано в имени ПЛК Конфигурация на вкладке Настройки связи устройства. Он позволяет использовать до 49 символов.
Активный адрес IP	Адрес IP ПЛК. Доступно, только если установлен флажок Использовать блочный драйвер TCP/IP. Он используется только тогда, когда настройка не является избыточной.	192.168.15.1	0.0.0.0 до 255.255.255.255
Адрес IP PLC A	IP-адрес ПЛК А. Активируется, только если конфигурация избыточна. Это основной адрес ПЛК, с которым сервер будет связываться в случае отсутствия сбоя.	192.168.15.69	0.0.0.0 до 255.255.255.255
Адрес IP PLC B	IP-адрес ПЛК В. Активируется только при избыточной конфигурации. Это вторичный адрес ПЛК, с которым сервер будет связываться в случае сбоя.	192.168.15.70	0.0.0.0 до 255.255.255.255

## 5. КОНФИГУРАЦИЯ

Порт устройства	TCP-порт. Доступно только при установленном флажке Use TCP/IP Blockdriver.	11740	11740 или 11739
-----------------	--	-------	-----------------

Таблица 123: Конфигурация параметров каждого ПЛК для сервера OPC DA

Когда необходимо настроить новый ПЛК на сервере OPC DA, просто нажмите кнопку «Новое устройство», и Конфигурация будет создана. При доступе к экрану настройки отобразится список всех ПЛК, уже настроенных на сервере OPC DA. Существующие конфигурации можно редактировать, выбрав ПЛК в списке устройств и отредактировав параметры. Настройки ПЛК, которые больше не используются, можно удалить. Максимальное количество ПЛК, сконфигурированных на сервере OPC DA, равно 16.

Если используемая архитектура автоматизации указывает, что сервер OPC DA должен быть запущен на компьютере, который не выполняет связь с ПЛК через MasterTool IEC XE, инструмент должен быть установлен на этом компьютере, чтобы разрешить настройку сервера OPC DA так же, как это делается в других ситуациях

### ВНИМАНИЕ

Чтобы сохранить конфигурацию сервера OPC DA, MasterTool IEC XE должен быть запущен с правами администратора в операционной системе. В зависимости от версии ОС эту привилегию необходимо выполнить в момент запуска программы: щелкните правой кнопкой мыши значок MasterTool IEC XE и выберите «Запуск от

### ВНИМАНИЕ

Настройки ПЛК на сервере OPC DA не сохраняются в проекте, созданном в MasterTool IEC XE. По этой причине он может быть выполнен с открытым или закрытым проектом. Настройки хранятся в файле конфигурации, где установлен сервер OPC DA. При изменении настроек не требуется загружать приложение на ПЛК, но в зависимости от клиента OPC может потребоваться переподключение к серверу или загрузка настроек для корректного обновления данных.

#### 5.7.9.2.1. Импорт конфигурации проекта

С помощью кнопки чтения конфигурации проекта, как показано на рисунке 95, вы можете назначить Конфигурацию открытого проекта редактируемой Конфигурации ПЛК. Чтобы эта опция работала правильно, должен быть открыт проект и должен быть установлен активный путь, как описано в разделе «Настройки связи», представленном в Руководстве пользователя MasterTool IEC XE — MU299609. Если какое-либо из этих условий не выполняется, появится сообщение об ошибке, и никакие данные не будут изменены.

Когда вышеуказанные условия выполняются, настройки ПЛК получают параметры открытого проекта. Информация об IP-адресе и порте шлюза согласовывается, как описано в разделе «Настройки связи», в соответствии с активным путем. Однако настройки IP-адреса считываются из настроек интерфейса NET 1 Ethernet. Порт для подключения к ПЛК в этом случае всегда назначается как 11740.

#### 5.7.9.3. Статус связи OPC DA и переменные качества

Для каждого ПЛК, созданного на сервере OPC DA, генерируются переменные состояния с именами \_CommState и \_CommStateOK. Переменная \_CommState указывает на связь между OPC и состоянием ПЛК. Это состояние может интерпретироваться клиентами OPC в соответствии с приведенной ниже таблицей.

Состояние	Значение	Описание
STATE_TERMINATE	-1	Если связь между сервером OPC DA и клиентом OPC прерывается, возвращается это значение. Когда одновременно подключено более одного клиента OPC, этот возврат произойдет при отключении последнего подключенного клиента. Помимо того, что это состояние находится в переменной, его значение нельзя визуализировать, потому что оно меняется только тогда, когда связь с клиентом пропадает.

## 5. КОНФИГУРАЦИЯ

<b>STATE_PLC_NOT_CONNECTED</b>	0	ПЛК, настроенный на сервере OPC DA, не подключен. Это может произойти, если Конфигурация неверна (неправильный IP-адрес ПЛК и/или шлюза) или ПЛК в данный момент недоступен.
<b>STATE_PLC_CONNECTED</b>	1	ПЛК, сконфигурированный на сервере OPC DA, подключен. Это переходное состояние во время соединения.

Состояние	Значение	Описание
<b>STATE_NO_SYMBOLS</b>	2	В ПЛК, сконфигурированном на сервере OPC DA, нет доступных символов (переменных). Это может произойти, когда отсутствуют символы или в ПЛК не загружен проект.
<b>STATE_SYMBOLS_LOADED</b>	3	Завершен процесс чтения символов (переменных) из ПЛК, сконфигурированного на сервере OPC DA. Это переходное состояние во время соединения.
<b>STATE_RUNNING</b>	4	После считывания символов (переменных) сервер OPC DA запускает периодическое обновление значений доступных символов в каждом сконфигурированном ПЛК.
<b>STATE_DISCONNECT</b>	5	Произошло разъединение с ПЛК, сконфигурированным на сервере OPC DA.
<b>STATE_NO_КОНФИГУРАЦИЯ</b>	6	Если Конфигурация OPC (хранящаяся в файле OPCServer.ini) имеет неправильный синтаксис, значение переменной будет таким. Обычно такое поведение не наблюдается для Master-Tool IEC XE, поддерживающего эту Конфигурацию в силе.

Таблица 124: Описание состояний связи между сервером OPC DA и ПЛК

**\_CommStateOK** — это переменная типа Bool, которая указывает, работает ли связь между сервером OPC DA и ПЛК. Когда значение TRUE, это указывает на то, что связь работает правильно. Если значение равно FALSE, по какой-то причине связь с ПЛК невозможна.

В дополнение к мониторингу состояния связи клиент OPC может получить доступ к информации о качестве связи. Биты качества образуют байт. Они разделены на три группы битов: Качество, Подстатус и Предел. Биты распределяются следующим образом: QQSSSSL, где QQ — это биты качества, биты подстатуса SSSS и биты ограничения LL. В этом случае биты QQ являются старшими в байте, а биты LL — наименее значащими.

QQ	Значение битов	Определение	Описание
0	00SSSSL	плохо	Считанное значение не может быть использовано из-за проблем с соединением. Можно отслеживать значение <b>_CommState</b> и диагностировать проблему.
1	01SSSSL	Неуверенный	Качество не может быть определено и может быть представлено в поле Substatus.
2	10SSSSL	Не указано	Это значение зарезервировано и не используется стандартом OPC.
3	11SSSSL	Хорошо	Качество хорошее, можно использовать прочитанное значение

Таблица 125: Описание значения качества OPC

В таблице 125 представлены возможные значения качества. Сервер OPC DA возвращает только значения хорошего и плохого качества. Клиент OPC может сохранять качество как неопределенное из-за сбоев, при которых он не может установить соединение с сервером. В случае мониторинга 8 битов качества непосредственно с сервера

## 5. КОНФИГУРАЦИЯ

---

OPC DA, поля Substatus и Limit должны быть нулевыми, а хорошее качество будет представлено как значение 192, а плохое качество будет иметь значение 0.

### 5.7.9.4. Ограничения связи с сервером OPC DA

В приведенной ниже Таблице представлены ограничения Конфигурации сервера OPC DA.

Максимальное количество переменных, взаимодействующих с одним ПЛК	-
Максимальное количество ПЛК в сервере OPC DA	16
Максимальное количество одновременных подключений сервера OPC DA в одном ПЛК	8

Таблица 126: Ограничения связи с сервером OPC DA

**Примечание:**

**Максимальное количество переменных, взаимодействующих с одним ПЛК:** Конфигурация не ограничена. Максимально возможное количество переменных зависит от вычислительной мощности устройства.

Для связи между сервером OPC DA и ПЛК используется тот же протокол, что и для связи MasterTool IEC XE с ПЛК. Этот протокол доступен только для интерфейсов Ethernet процессоров серии Nexto, установить такой тип связи с модулями расширения Ethernet невозможно.

Когда связь между сервером OPC DA и ПЛК установлена, эти два элемента начинают серию транзакций, направленных на определение адресов каждой объявленной переменной, оптимизируя связь в режиме чтения данных. Кроме того, на этом этапе также решаются коммуникационные группы, используемые некоторыми Клиентами, для оптимизации коммуникации. Этот начальный процесс требует некоторого времени и зависит от количества отображаемых переменных и вычислительной мощности устройства.

Для связи между сервером OPC DA и ПЛК используется тот же протокол, что и для связи MasterTool IEC XE с ПЛК. Этот протокол доступен только для интерфейсов Ethernet процессоров серии Nexto, установить такой тип связи с модулями расширения Ethernet невозможно.

**ВНИМАНИЕ**

Максимальное количество одновременных подключений сервера OPC DA к одному ПЛК совпадает с количеством подключений, выполненных с помощью MasterTool IEC XE. т.е. сумма подключений OPC DA Server и MasterTool IEC XE не должна превышать максимальное количество, указанное в таблице 126.

Когда связь между сервером OPC DA и ПЛК установлена, эти два элемента начинают серию транзакций, направленных на определение адресов каждой объявленной переменной, оптимизируя связь в режиме чтения данных. Кроме того, на этом этапе также решаются коммуникационные группы, используемые некоторыми Клиентами, для оптимизации коммуникации. Этот начальный процесс требует некоторого времени и зависит от количества отображаемых переменных и вычислительной мощности устройства.

**5.7.9.5. Доступ к данным через клиент OPC DA**

После Конфигурации сервера OPC DA доступные данные на всех ПЛК могут быть доступны через клиента OPC. В Конфигурации клиента OPC должно быть выбрано имя сервера OPC DA. В данном случае это CoDeSys.OPC.DA. На рисунке ниже показан выбор сервера в клиентском драйвере программного обеспечения BluePlant SCADA.

**ВНИМАНИЕ**

Так же, как и в MasterTool IEC XE, некоторые инструменты должны выполняться с правами администратора в операционной системе для правильного функционирования клиента OPC. В зависимости от версии ОС эта привилегия должна быть активирована в момент запуска программы. Для этого щелкните правой кнопкой мыши значок MasterTool IEC XE и выберите «Запуск от имени администратора».

## 5. КОНФИГУРАЦИЯ

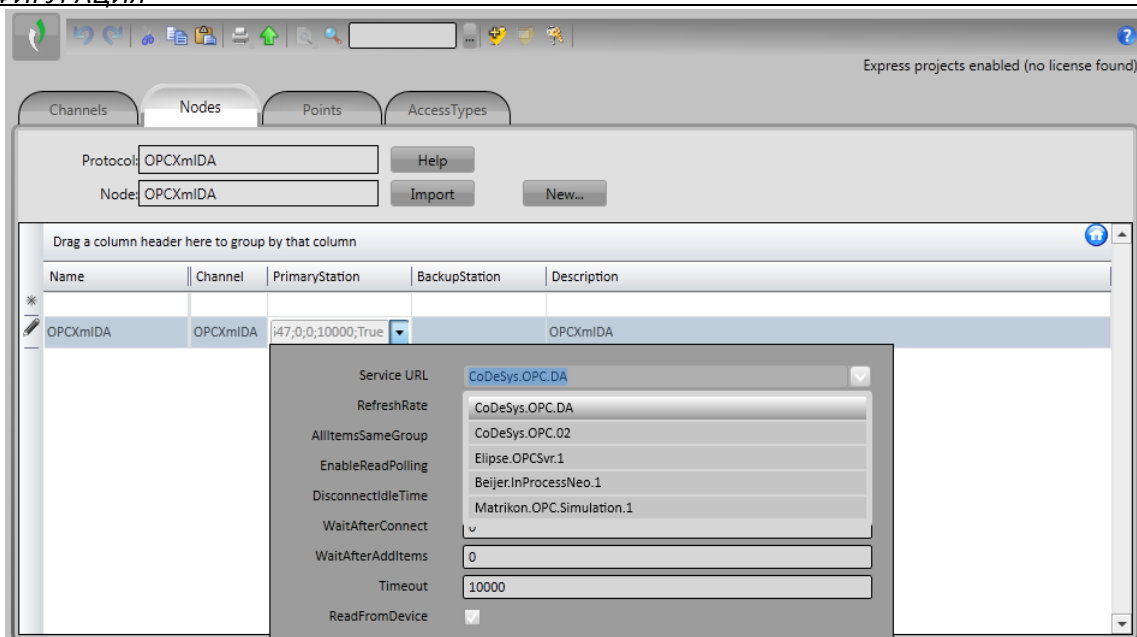


Рисунок 96: Выбор сервера OPC DA в клиентской конфигурации

В случаях, когда сервер расположен удаленно, может потребоваться добавить сетевой путь или IP-адрес компьютера, на котором установлен сервер. В этих случаях есть два варианта конфигурации. Во-первых, это прямое указание на Рисунок, поскольку это необходимо для включения службы COM/DCOM Windows. Однако проще использовать инструмент туннелирования, который абстрагирует параметры COM/DCOM и обеспечивает более безопасную связь между клиентом и сервером. Для получения дополнительной информации об этом типе инструмента обратитесь к NAP151 - Tunneller OPC.

После того, как клиент подключится к серверу, можно использовать команды импорта тегов. Эти команды обращаются к информации, объявленной в ПЛК, возвращая список со всеми доступными в нем символами.

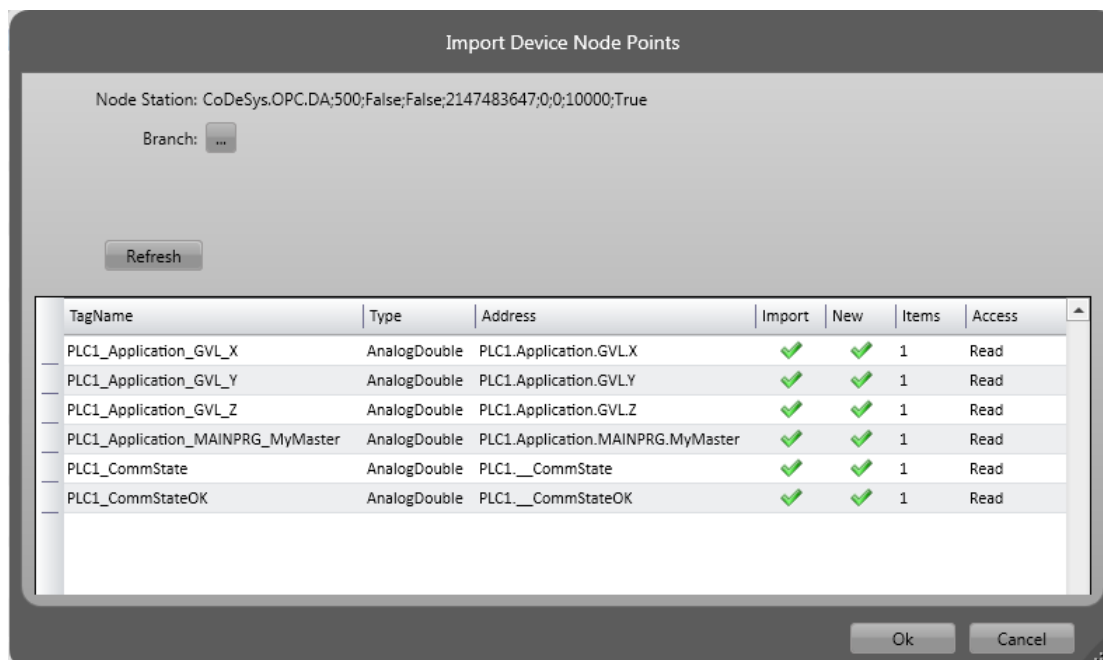


Рисунок 97: Список символов, к которому обращается клиент OPC

Список выбранных переменных будет включен в список связи с клиентом и может использоваться, например, на экране системы SCADA.

**ВНИМАНИЕ**

Режим моделирования программного обеспечения MasterTool IEC XE можно использовать для тестирования связи OPC. Информация о том, как его настроить, представлена в разделе «Тестирование связи OPC с использованием симулятора» руководства пользователя MasterTool IEC XE — MU1299600.

**5.7.10. Сервер OPC UA**

Протокол OPC UA является развитием семейства OPC. Независимо от платформы, он призван стать новым стандартом, используемым в промышленных коммуникациях.

Основанный на архитектуре клиент/сервер, протокол OPC UA предлагает многочисленные преимущества при разработке дизайна и средств связи с системами автоматизации.

Когда дело доходит до разработки проекта, настройка связи и обмена информацией между системами чрезвычайно проста с использованием технологии OPC UA. Используя другие адресные драйверы, необходимо создать Таблицы, чтобы связать теги системы контроля и программируемые переменные контроллера. Когда области данных изменяются во время разработки проекта, необходимо переделать сопоставления и новые таблицы с отношениями между информацией ПЛК и системой SCADA.

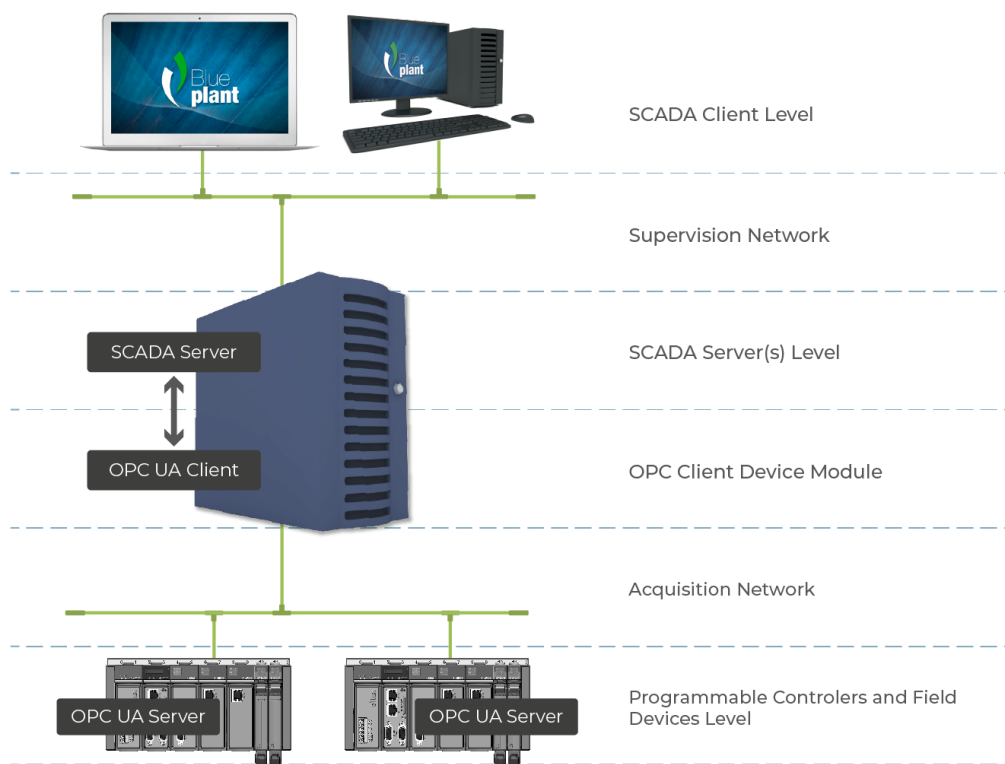


Рисунок 98: Архитектура OPC UA

На приведенном выше рисунке представлена типичная архитектура для связи системы SCADA и ПЛК в системе автоматизации. На этом рисунке явно указаны все роли, присутствующие в общении, независимо от того, где они работают, они могут быть на одном и том же оборудовании или на другом оборудовании. Каждая из ролей этой архитектуры описана в Таблице ниже.

Роль	Описание
Уровень программируемых контроллеров и полевых устройств	В полевых устройствах и ПЛК хранится информация о рабочем состоянии и управлении предприятием. Система SCADA получает доступ к информации об этих устройствах и сохраняет ее на сервере SCADA, чтобы клиенты SCADA могли обращаться к ней во время работы установки.
Серверные модули OPC UA	Сервер OPC UA — это внутренний модуль ПЛК, отвечающий за получение запросов OPC UA и их преобразование для связи с полевыми устройствами.
Сеть приобретения	Сеть сбора данных — это сеть, по которой перемещаются сообщения OPC UA для запроса данных, собранных с ПЛК и полевых устройств.
Модуль клиентского устройства OPC	Модуль клиента OPC UA, являющийся частью сервера SCADA, отвечает за выполнение запросов к серверам OPC UA с использованием протокола OPC UA. Собранные им данные хранятся в базе данных SCADA Server.
Уровень сервера SCADA	Сервер SCADA отвечает за подключение к различным коммуникационным устройствам и хранение собранных ими данных в базе данных, чтобы клиенты SCADA могли обращаться к ним.
Сеть наблюдения	Супервизорная сеть — это сеть, по которой клиенты SCADA подключаются к серверам SCADA, часто с использованием собственного системного протокола SCADA. В топологии, в которой не используются несколько Клиентов или Сервер и Клиент установлены в одном оборудовании, такой сети нет, и в этом случае это оборудование должно напрямую использовать протокол OPC UA для связи с ПЛК.
Уровень клиента SCADA	Клиенты SCADA несут ответственность за запрос к серверам SCADA необходимых данных для отображения на экране, где выполняется работа установки. После этого можно выполнять чтение и запись данных, хранящихся в базе данных сервера SCADA.

Таблица 127: Описание роли на архитектуре сервера OPC UA

При использовании протокола OPC UA взаимосвязь между тегами систем супервизора и данными процесса в переменных контроллера полностью прозрачна. Это означает, что если области данных изменяются во время разработки проекта, нет необходимости повторно устанавливать отношения между информацией ПЛК и SCADA. Просто используйте новую переменную, предоставленную ПЛК, в системах, запрашивающих эти данные.

Использование OPC UA обеспечивает большую производительность и возможность подключения к системам SCADA. Это способствует сокращению времени разработки приложений и затрат на обслуживание. Это также позволяет вводить новые данные в коммуникацию упрощенным способом с большей гибкостью и функциональной совместимостью между системами автоматизации, поскольку это открытый стандарт.

Стоит отметить, что OPC UA доступен только на локальных интерфейсах Ethernet процессоров Nexto. Модули расширения Ethernet не поддерживают эту функцию.

#### 5.7.10.1. Создание проекта для связи OPC UA

Шаги по созданию проекта с OPC UA очень похожи на шаги, описанные в разделе Создание проекта для связи OPC DA. Как и в случае с протоколом OPC DA, конфигурация протокола OPC UA основана на конфигурации Symbol. Чтобы включить OPC UA, просто включите параметр Support OPC UA Features в конфигурации, как показано на рисунке ниже.

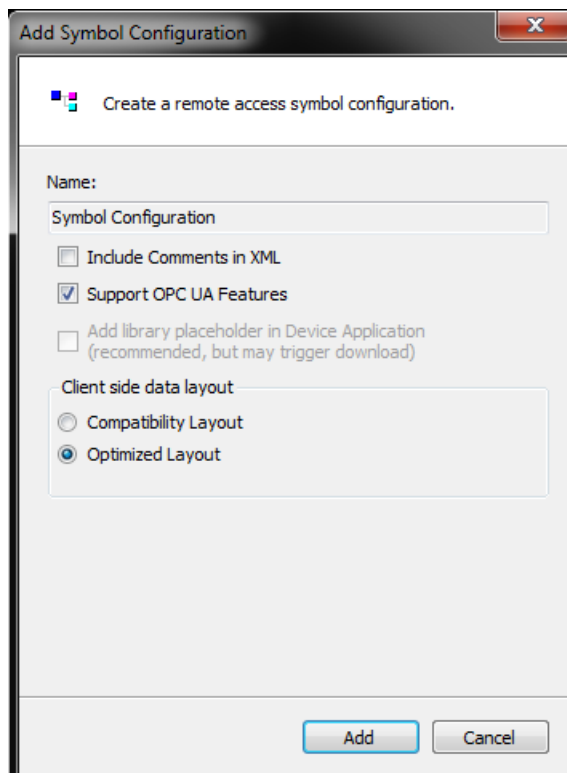


Рисунок 99: Символ конфигурации объекта

**ВНИМАНИЕ**

При включении поддержки протокола OPC UA поддержка протокола OPC DA по-прежнему включена. Вы можете одновременно включить связь OPC UA и OPC DA, чтобы сообщать о переменных, настроенных в объекте конфигурации символа или через атрибуты.

Другой способ получить доступ к этой конфигурации после того, как уже создан проект с объектом конфигурации символа, предоставляется путем доступа к меню «Настройки» на вкладке конфигурации символа. Просто выберите параметр «Поддержка функций OPC UA», чтобы включить поддержку протокола OPC UA, как показано на рисунке ниже.

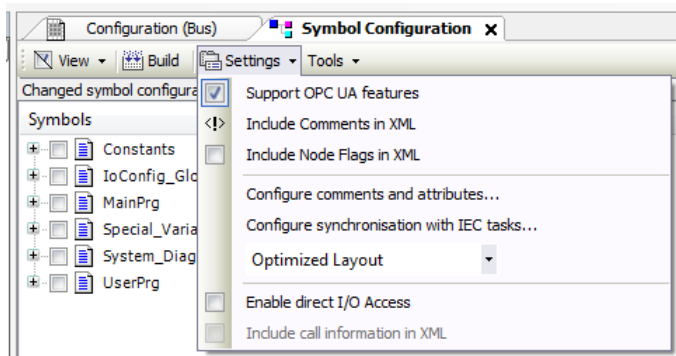


Рисунок 100: Включение OPC UA в конфигурации символа объекта

После данной процедуры проект можно загрузить в ПЛК, и выбранные переменные будут доступны для связи с сервером OPC UA.

**5.7.10.2. Типы поддерживаемых переменных**

В этом разделе определяются типы переменных, поддерживающих обмен данными по протоколу OPC UA, когда они объявлены в GVL или ROU и выбраны в объекте Конфигурация символа (см. предыдущий раздел).

Поддерживаются следующие типы простых переменных:

BOOL SINT  
USINT / БАЙТ INT

## 5. КОНФИГУРАЦИЯ

---

UINT / СЛОВО DINT  
UDINT/DWORD LINT  
ULINT/LWORD РЕАЛЬНОЕ  
LREAL STRING TIME LTIME

Вы также можете использовать структурированные типы (STRUCT или функциональные блоки), созданные из предыдущих простых типов. Наконец, также можно создавать массивы простых или структурированных типов.

### 5.7.10.3. Ограничение количества подключенных клиентов на сервере OPC UA

Максимальное количество клиентов OPC UA, одновременно подключенных к ПЛК, равно 8 (восемь).

### 5.7.10.4. Ограничение коммуникационных переменных на сервере OPC UA

Конфигурация не ограничена. Максимально возможное количество переменных зависит от вычислительной мощности устройства.

Когда устанавливается связь между сервером OPC UA и ПЛК, эти два элемента иницируют серию транзакций, направленных на определение адреса каждой объявленной переменной, оптимизируя связь в режиме чтения данных. Кроме того, на этом этапе также разрешаются классификации групп связи, используемых некоторыми Клиентами, с целью оптимизации коммуникации. Этот начальный процесс занимает некоторое время и зависит от количества отображаемых переменных и вычислительной мощности устройства.

### 5.7.10.5. Настройки шифрования

При желании пользователь может настроить шифрование для обмена данными OPC UA с использованием профиля Basic256SHA256 для безопасного соединения (кибербезопасность).

Чтобы настроить шифрование на сервере OPC UA, необходимо создать для него сертификат, выполнив следующие действия в программаторе MasterTool:

1. Определите активный путь для связи с контроллером (вход в систему не требуется);
  2. В меню «Вид» выберите «Экран безопасности»;
  3. Щелкните вкладку «Устройства» в левой части этого экрана;
  4. Щелкните значок, чтобы выполнить обновление;
  5. Щелкните значок Устройство, под которым откроется несколько сертификатов (Собственные сертификаты, Доверенные сертификаты, Недоверенные сертификаты, Сертификаты на карантине);
  6. Щелкните значок для создания сертификата и выберите следующие параметры:  
Длина ключа (бит): 3072  
Срок действия (дней): 365 (при желании можно изменить)
  7. Подождите, пока сертификат будет рассчитан и передан в контроллер (это может занять несколько минут);
  8. Перезагрузите контроллер.
  9. На клиенте OPC UA выполните необходимые процедуры для подключения к серверу OPC UA и создания сертификата с профилем Basic256Sha256 (подробности см. в руководстве по конкретному клиенту OPC UA);
  10. Вернитесь в MasterTool, щелкните значок экрана безопасности, чтобы выполнить обновление;
  11. На экране «Безопасность» выберите папку «Сертификаты на карантине» в разделе «Устройство». На правой панели вы должны увидеть сертификат, запрошенный клиентом OPC UA;
  12. Перетащите этот сертификат в папку «Доверенные сертификаты»;
- Продолжите настройку в клиенте OPC UA (подробности см. в руководстве по конкретному клиенту OPC UA).

Чтобы удалить ранее настроенное на контроллере шифрование, необходимо сделать следующее:

1. Определите активный путь для связи с контроллером (вход в систему не требуется);
2. В меню «Вид» выберите «Экран безопасности»;
3. Нажмите «Устройства» в левой части этого экрана;
4. Щелкните значок, чтобы выполнить обновление;
5. Щелкните значок Устройство, под которым откроется несколько сертификатов (Собственные сертификаты, Доверенные сертификаты, Недоверенные сертификаты, Сертификаты на карантине);
6. Щелкните папку «Собственные сертификаты» и в правой панели выберите сертификат (OPC UA Server);
7. Щелкните значок, чтобы удалить этот проект и сертификат драйвера;
8. Сбросьте (выключите и снова включите) контроллер.

### 5.7.10.6. Основные параметры связи, настроенные в клиенте OPC UA

Некоторые параметры связи OPC UA согласовываются на клиенте OPC UA и согласовываются с сервером OPC UA во время установления соединения между ними. В следующих подразделах описываются основные параметры связи OPC UA, их значение и выбор для них подходящих значений.

В клиенте OPC UA можно сгруппировать переменные сервера в разные подписки. Каждая подписка представляет собой набор переменных, о которых сообщается в одном коммуникационном пакете (PublishResponse), отправляемом с сервера клиенту. Выбор переменных, которые будут составлять каждую подписку, производится в клиенте OPC UA.

## 5. КОНФИГУРАЦИЯ

Некоторые из описанных ниже параметров связи должны быть определены для сервера в целом, другие для каждой подписки, третьи для каждой переменной, составляющей подписку.

### ВНИМАНИЕ

Группировка переменных в несколько подписок интересна для оптимизации производительности обработки и использования полосы пропускания Ethernet. Такие аспекты оптимизации более подробно проанализированы в примечаниях по применению NAR165, где предлагаются некоторые правила для составления подписок. В этом примечании по применению также более подробно обсуждаются пакеты Publish Response протокола OPC UA.

#### 5.7.10.6.1. Конечная точка URL

Этот параметр определяет IP-адрес и TCP-порт сервера, например:

opc.tcp://192.168.17.2:4840

В этом примере IP-адрес контроллера — 192.168.17.2. Порт TCP всегда должен быть 4840.

#### 5.7.10.6.2. Интервал публикации (мс) и интервал выборки (мс)

Параметр Publishing Interval (единица измерения: миллисекунды) должен быть установлен для каждой подписки.

Параметр Sampling Interval должен быть установлен для каждой переменной (единица измерения: миллисекунды). Однако во многих клиентах OPC UA

Параметр Sampling Interval может быть определен для подписки, он одинаков для всех переменных, сгруппированных в подписке.

Только переменные подписки, значения которых были изменены, сообщаются клиенту через пакет связи Publish Response. Параметр Publishing Interval определяет минимальный интервал между последовательными пакетами Publish Response одной и той же подписки, чтобы ограничить потребление обработки и пропускную способность связи Ethernet.

Чтобы выяснить, какие переменные подписки изменились и должны быть сообщены клиенту в следующем пакете ответа на публикацию, сервер должен выполнить сравнения, и такие (выборки) выполняются тем же самым с интервалом выборки. Рекомендуется, чтобы значение интервала выборки варьировалось от 50 % до 100 % значения интервала публикации, так как процесс сравнения, выполняемый в каждом интервале выборки, требует относительно большого объема обработки.

Можно сказать, что сумма между Publishing Interval и Sampling Interval представляет собой максимальную задержку между изменением значения на сервере и передачей пакета Publish Response, сообщающего об этом изменении. Половина этой суммы представляет собой среднюю задержку между изменением значения на сервере и передачей пакета ответа на публикацию, сообщающего об этом изменении.

#### 5.7.10.6.3. Счетчик жизненного цикла и счетчик поддержания активности

Эти два параметра должны быть согласованы для каждой подписки.

Назначение этих двух параметров — создать механизм деактивации подписки по инициативе сервера, в случае, если он длительное время не получает коммуникационные пакеты PublishRequest от клиента для этой подписки. Пакеты PublishRequest должны быть получены сервером, чтобы он мог транслировать пакеты Publish Response, содержащие переменные подписки, значения которых изменились.

Если сервер не получает пакеты PublishRequest в течение времени, превышающего значение Lifetime Count, умноженное на Publishing Interval, сервер деактивирует подписку, которую при желании клиент должен создать заново в будущем.

В ситуациях, когда переменные подписки не изменяются, может пройти много времени без успешной передачи PublishResponse и, следовательно, PublishRequests, что приведет к нежелательной деактивации подписки. Чтобы этого не происходило, был создан параметр Keep-Alive Count. Если данные подписки не изменяются в течение времени, равного счетчику активности, умноженному на интервал публикации, сервер отправит небольшой пустой пакет ответа на публикацию, указывающий, что никакая переменная не изменилась. Этот пустой ответ на публикацию разрешает клиенту немедленно отправить следующий запрос на публикацию.

Значение Keep-Alive Count должно быть меньше значения Lifetime Count, чтобы предотвратить нежелательную деактивацию подписки. Предполагается, что значение Lifetime Count должно быть как минимум в 3 раза больше, чем Keep-Alive Count.

#### 5.7.10.6.4. Размер очереди и удаление самых старых

Эти параметры должны поддерживаться со следующими фиксированными значениями, которые обычно являются значениями по умолчанию для клиентов: Размер очереди: 1

Отбрасывать самые старые: включить

Согласно стандарту OPC UA эти параметры можно определить для каждой переменной. Однако многие клиенты позволяют вам определять общие значения для всех переменных, настроенных в подписке.

Размер очереди должен быть сохранен со значением 1, поскольку в этой реализации сервера OPC UA нет

## 5. КОНФИГУРАЦИЯ

---

поддержки событий, поэтому нет необходимости определять очередь. Увеличение значения размера очереди может означать увеличение пропускной способности связи и обработки ЦП, и этого следует избегать.

Discard Oldest должен поддерживаться со значением enable, чтобы пакет Publish Response всегда сообщал о самом последнем изменении значения, обнаруженном для каждой переменной.

### 5.7.10.6.5. Тип фильтра и тип зоны нечувствительности

Эти параметры должны поддерживаться со следующими фиксированными значениями, которые обычно являются значениями по умолчанию в клиентах:

Тип фильтра: DataChangeFilter Тип зоны нечувствительности: нет

Согласно стандарту OPC UA эти параметры можно определить для каждой переменной. Однако многие клиенты позволяют вам определять общие значения для всех переменных, настроенных в подписке.

Параметр Тип фильтра должен иметь значение DataChangeFilter, указывающее, что изменения значений в переменных должны привести к их передаче в пакете ответа на публикацию.

Для типа зоны нечувствительности следует оставить значение «нет», потому что для аналоговых переменных зоны нечувствительности не реализованы. Таким образом, любое изменение аналоговой переменной, даже минимальное, вызывает ее передачу в пакете Publish Response.

Чтобы уменьшить вычислительную мощность и пропускную способность связи Ethernet, вы можете самостоятельно развернуть зоны нечувствительности следующим образом:

Не включайте аналоговую переменную в подписку;

Вместо этого включите в подписку вспомогательную переменную, связанную с аналоговой переменной;

Копируйте аналоговую переменную во вспомогательную переменную только тогда, когда зона нечувствительности, управляемая пользователем, экстраполируется.

### 5.7.10.6.6. PublishingEnabled, MaxNotificationsPerPublish и Priority

Рекомендуется, чтобы следующие параметры сохранялись со следующими значениями, которые обычно являются значениями по умолчанию в клиентах:

PublishingEnabled: true MaxNotificationsPerPublish: 0

Приоритет: 0

Эти параметры должны быть настроены для каждой подписки.

PublishingEnable должно иметь значение «true», чтобы о переменных подписки сообщалось в случае изменения значения.

MaxNotificationsPerPublish указывает, сколько переменных, значение которых изменилось, может быть включено в один и тот же пакет ответа на публикацию. Специальное значение «0» указывает на то, что для этого нет ограничений, и рекомендуется использовать это значение, чтобы обо всех измененных переменных сообщалось в одном и том же пакете Publish Response.

Приоритет указывает относительный приоритет этой подписки по сравнению с другими. Если в какой-то момент сервер должен отправить несколько пакетов Publish Response разных подписок, он отдаст приоритет тому, у которого наивысшее значение приоритета. Если все подписки имеют одинаковый приоритет, пакеты Publish Response будут передаваться в фиксированной последовательности.

### 5.7.10.7. Доступ к данным через клиент OPC UA

## 5. КОНФИГУРАЦИЯ

После Конфигурации сервера OPC UA к данным, доступным во всех ПЛК, можно получить доступ через клиент OPC UA. В Конфигурации клиента OPC UA должен быть выбран адрес правильного сервера OPC UA. В этом случае адрес `opc.tcp://ip-адрес-устройства:4840`. На рисунке ниже показан выбор сервера в драйвере клиентского программного обеспечения SCADA BluePlant.

### ВНИМАНИЕ

Как и MasterTool IEC XE, некоторые инструменты необходимо запускать с правами администратора в операционной системе для правильной работы клиента OPC UA. В зависимости от версии операционной системы это право необходимо авторизовать при запуске программы. Для этой операции щелкните правой кнопкой мыши исполняемый файл инструмента и выберите параметр «Запуск от имени администратора».

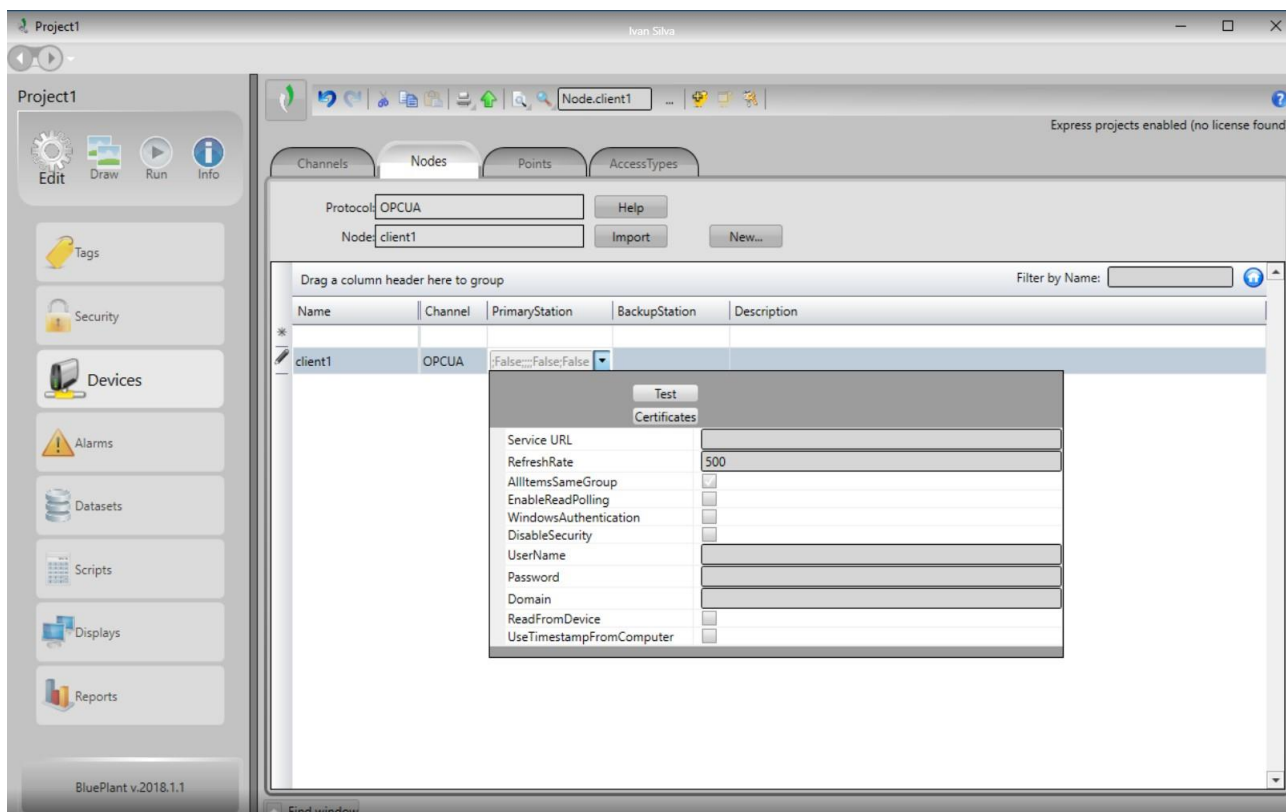


Рисунок 101: Выбор сервера OPC UA в конфигурации клиента

После подключения клиента к серверу можно использовать команды импорта TAG. Эти команды запрашивают информацию, объявленную в ПЛК, и возвращают список со всеми символами, доступными в ПЛК.

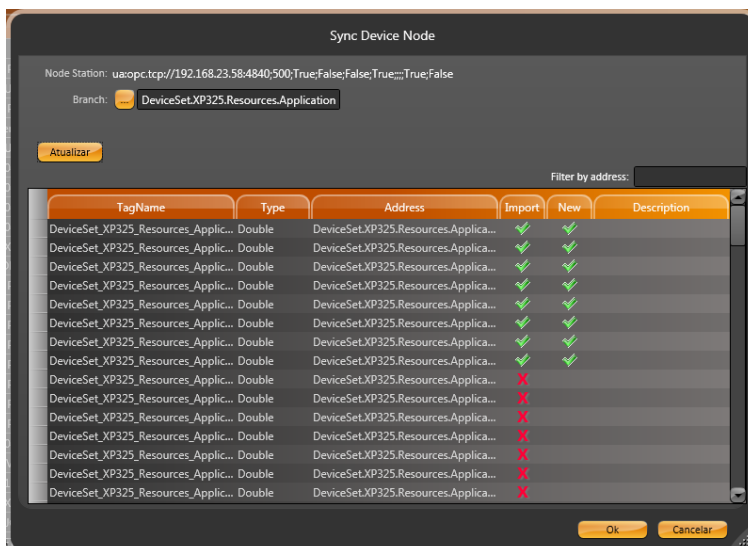


Рисунок 102: Список символов, просматриваемых OPC UA

## 5. КОНФИГУРАЦИЯ

Список выбранных переменных будет включен в список коммуникаций Клиента и может быть использован, например, в экранах SCADA-системы.

### 5.7.11. Мастер EtherCAT

EtherCAT (технология автоматизации управления Ethernet) — это протокол архитектуры ведущий-ведомый с высокой производительностью для детерминированного Ethernet, который обеспечивает производительность в режиме реального времени, поскольку он обновляет 1000 распределенных вводов-выводов за 30 мкс или 100 осей серводвигателей каждые 100 мкс с использованием кабелей витой пары или оптическое волокно. Кроме того, он поддерживает гибкую топологию, позволяющую использовать линейные, древовидные и/или звездообразные соединения.

Фрейм Ethernet может обрабатываться в режиме реального времени, а не приниматься, интерпретироваться и копироваться как данные процесса при каждом соединении. FMMU (модуль управления памятью полевой шины) в каждом подчиненном узле считывает адресованные ему данные одновременно с пересылкой телеграммы на следующее устройство. Аналогичным образом вводимые данные вставляются по мере прохождения телеграммы. Из-за этого кадры задерживаются всего на несколько наносекунд. Доступ к терминалам Ethernet может осуществляться в любом порядке, поскольку последовательность данных не зависит от физического порядка. Он может выполнять широковебательную, многоадресную и межведомственную связь.

Протокол EtherCAT обеспечивает точную синхронизацию, которая требуется, например, в приложениях, где несколько осей одновременно выполняют скоординированные движения, это можно сделать с помощью точной настройки распределенных часов. Также существует возможность подключения устройств, которые, в отличие от синхронной связи, имеют повышенную степень устойчивости внутри системы связи.

Конфигурация модулей EtherCAT изначально определяется файлами описания устройства используемых ведущего и ведомого устройств и может быть изменена пользователем в диалоговых окнах редактора конфигурации. Тем не менее, для обычных приложений и при желании как можно более простой манипуляции крупномасштабные Конфигурации можно автоматизировать, выбрав режим Autoconfig в EtherCAT Master — General.

Обратите внимание на возможность изменения параметров конфигурации Master и Slave также в рабочем режиме, через экземпляры Master и Slave, в зависимости от доступности рассматриваемого устройства.

#### 5.7.11.1. Установка и подключение устройств EtherCAT

Чтобы иметь возможность вставлять и конфигунок устройств EtherCAT как объектов в дереве устройств, должны быть установлены ведомые устройства.

Мастер-устройство автоматически устанавливается при установке MasterTool IEC XE. Мастер EtherCAT определяет, какие ведомые устройства могут быть вставлены.

Для установки ведомых устройств необходимо открыть репозиторий устройств, использовать фильтр EtherCAT XML Device Description Configuration File (\*.xml) и выбрать файлы описания устройства (EtherCAT XML Device Description / ESI EtherCAT Slave Information), поставляемые с оборудованием. Описание Slave доступно в виде файлов XML (тип файла: \*.xml).

Мастер EtherCAT можно добавить в дерево устройств с помощью команды «Добавить устройство» через контекстное меню разъемов CPU NET.

Под мастером можно добавить одно или несколько подчиненных устройств, выбрав мастер EtherCAT и выполнив команду «Добавить устройство» (контекстное меню мастера EtherCAT) или выполнив команду «Сканировать устройства».

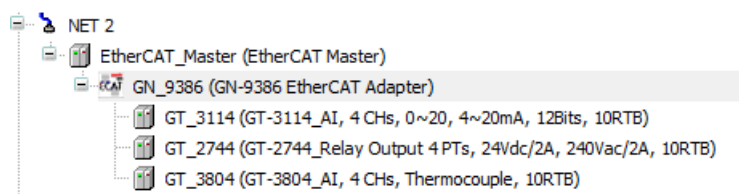


Рисунок 103: Пример конфигурации EtherCAT

#### ВНИМАНИЕ

- Допускается только один экземпляр EtherCAT Master на проект.
- Доступно только на разъемах NET ПЛК.
- Его нельзя использовать, когда сети настроены как резервные.
- Его нельзя использовать, если Project имеет избыточность кластера.
- Другие драйверы не могут быть установлены в тот же порт NET, что и мастер EtherCAT.
- Он поддерживает максимум 128 ведомых устройств EtherCAT на проект..

## 5. КОНФИГУРАЦИЯ

### 5.7.11.1.1. Скан устройства EtherCAT

Команда «Сканировать устройства», доступная в контекстном меню EtherCAT Master, запускает поиск подчиненных устройств, физически установленных в сети EtherCAT подключенного в данный момент ПЛК. Это означает, что с помощью этой команды можно обнаружить и визуализировать аппаратные компоненты в окне, представленном на рисунке ниже, что позволяет пользователю отображать их непосредственно в дереве устройств проекта do projecto.

Примечательно, что при выборе команды «Сканировать устройства» соединение с ПЛК будет автоматически установлено до начала поиска и прервано по окончании поиска. Итак, для первого выполнения этой команды необходимо разрешить соединение Gateway и загрузить в ПЛК программу с EtherCAT Master configured. В случае будущих добавлений ведомых устройств для запуска этой команды необходимо, чтобы сеть EtherCAT была остановлена. Для этого установите в TRUE бит bStopBus, присутствующий в переменных диагностики EtherCAT Master.

При выполнении команды в поле Scanned Devices появится список всех устройств и модулей, найденных при последнем сканировании. Чтобы добавить их в проект, достаточно нажать на кнопку «Копировать все устройства в проект». Также можно выполнить сравнение устройств, найденных в поиске, с устройствами в проекте, установив флажок Показать отличия от проекта.

Если вы добавите в проект ведущий модуль EtherCAT и воспользуетесь командой «Сканировать устройства», вы получите список всех доступных ведомых устройств EtherCAT. Записи, выделенные жирным шрифтом, будут показаны, если имеется более одного устройства с одинаковым описанием. Двойным щелчком по входу откроется список, и можно будет выбрать нужное устройство.

После внесения изменений в Конфигурацию сети EtherCAT необходимо сделать новую загрузку проекта, чтобы изменения вступили в силу.

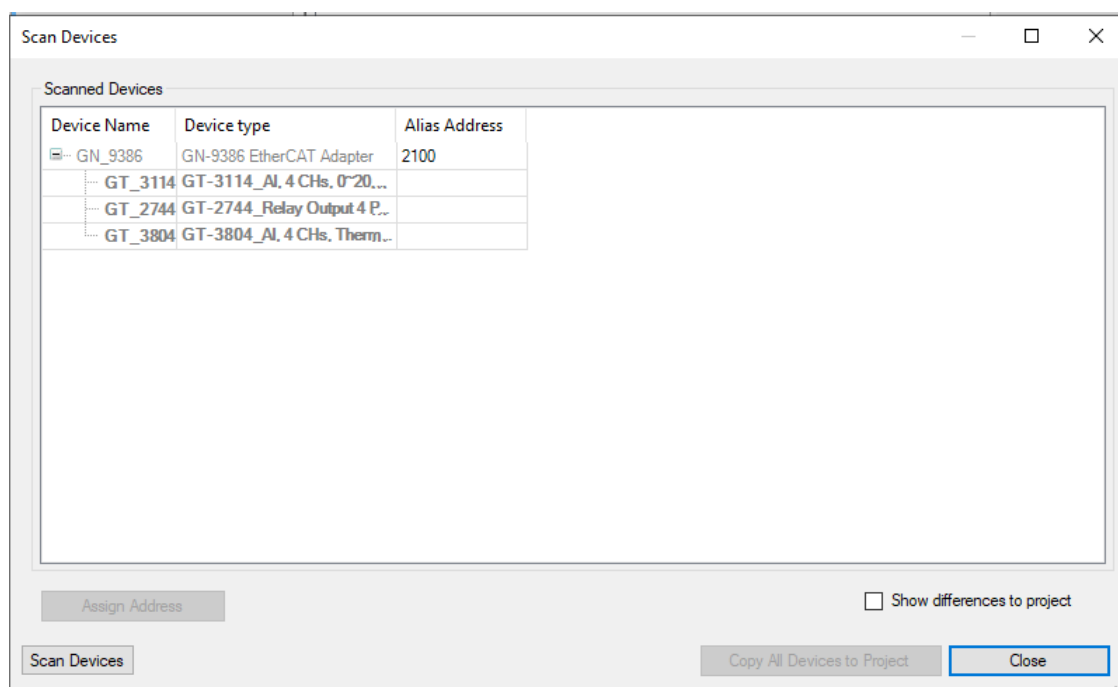


Рисунок 104: Диалоговое окно поиска устройств EtherCAT

### 5.7.11.2. Основные настройки EtherCAT

Ниже перечислены варианты выполнения главной конфигурации EtherCAT, такие как определены в файле описания устройства.

#### 5.7.11.2.1. Мастер EtherCAT — Общие

Ниже приведены общие параметры, находящиеся на начальном экране EtherCAT Master Конфигурация, согласно рисунку ниже.

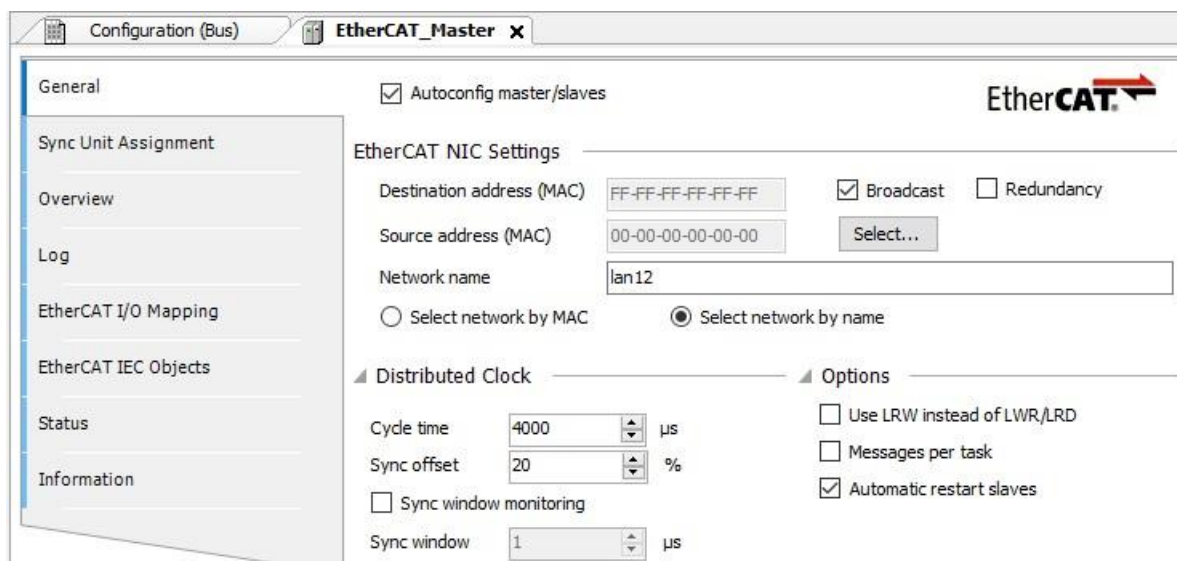


Рисунок 105: Диалог конфигурации EtherCAT Master

Конфигурация устройства	Описание	Заводские настройки по умолчанию	Возможные значения
Мастер/ведомые автоконфигурации	Включите автоматическую Конфигурацию Master и Slave.	Отмечено	Отмечено Не отмечено
Время цикла [мкс]	Устанавливает период времени, в течение которого новая телеграмма данных должна быть отправлена на шину.	4000	от 2000 до 1000000
Смещение синхронизации [%]	Отрегулируйте смещение прерывания синхронизации EtherCAT Slave относительно цикла ПЛК.	20	от -50 до 50
Мониторинг окна синхронизации	Если эта опция включена, она позволяет контролировать синхронизацию Slave.	Отмечено	Отмечено Не отмечено
Окно синхронизации [мкс]	Время Мониторинга Окна Синхронизации.	1	от 1 до 32768
Используйте LRW вместо LWR/LRD	Включение комбинированных команд чтения и записи.	Отмечено	Отмечено Не отмечено
Сообщений на задачу	Если эта функция включена, команды чтения и записи, имеющие дело с входными и выходными сообщениями, могут выполняться в разных задачах.	Отмечено	Отмечено Не отмечено
Ведомые устройства с автоматическим перезапуском	Перезапустите устройства, когда связь будет прервана.	Отмечено	Отмечено Не отмечено

Таблица 128: Конфигурация EtherCAT Master

**Примечания:**

## 5. КОНФИГУРАЦИЯ

Autoconfig master/slave: Если эта опция включена, большая часть Конфигурации Master и Slave будет выполняться автоматически, на основе файлов описания и неявных вычислений. В этом случае диалог FMMU/Sync будет недоступен. Если этот флажок не установлен, пользователю будут доступны варианты адреса входа

### ВНИМАНИЕ

Режим Autoconfig включен по умолчанию и обычно его достаточно и настоятельно рекомендуется для стандартных приложений. Если он отключен, все определения конфигурации должны быть сделаны вручную, и, следовательно, требуются некоторые специальные знания. Чтобы настроить связь между подчиненными, параметр Autoconfig должен быть отключен.

изображения и адреса выхода изображения.

**Время цикла:** Период времени, после которого на шину должна быть отправлена новая телеграмма данных. Если функция Distributed Clock включена, значение этого параметра будет передано на часы Slave. Таким образом достигается точная синхронизация обмена данными, что особенно важно в случаях, когда распределенный процесс требует одновременных действий. Таким образом, можно получить очень точную временную базу с дрожанием, значительно меньшим микросекунды, для всей сети.

**Смещение синхронизации:** это значение позволяет настроить смещение прерывания синхронизации подчиненного устройства EtherCAT относительно цикла ПЛК. Обычно цикл задач ПЛК начинается на 20 % позже, чем прерывание синхронизации ведомых устройств. Это означает, что задача ПЛК может быть задержана на 80 % времени цикла, и ни одно сообщение не будет потеряно.

**Окно синхронизации:** если синхронизация всех ведомых устройств находится в пределах этого временного окна, для диагностики EtherCAT Master bDistributed-ClockInSync будет установлено значение TRUE, в противном случае — значение FALSE. При использовании распределенных часов настоятельно рекомендуется использовать выделенную задачу с высоким приоритетом в качестве задачи цикла шины главного устройства EtherCAT. Для этого необходимо использовать Project Profiles, который позволяет создавать новые задачи, затем создать Циклическую задачу с приоритетом 0 (задача реального времени) и связать ее с главной задачей Bus cycle на вкладке EtherCAT Master — Input/Output Mapping. мастера EtherCAT. Пользователь также может изменить значение переменной wDCInSyncWindow, настроив максимальное допустимое дрожание синхронизации между ведущим и ведомыми устройствами.

**Использовать LRW вместо LWR/LRD:** активация этой опции включает связь между подчиненными, поскольку вместо использования отдельных команд чтения (LRD) и записи (LWR) будут использоваться комбинированные команды чтения/записи (LRW).

Автоматический перезапуск ведомых устройств: при включении этой опции ведущий перезапустит ведомые устройства, как только связь будет прервана.

### 5.7.11.2.2. EtherCAT Master — назначение устройства синхронизации

На этой вкладке редактора конфигурации главного устройства EtherCAT отображаются все ведомые устройства, введенные ниже определенного ведущего устройства, с назначением устройств синхронизации.

В модулях синхронизации EtherCAT несколько ведомых устройств объединяются в группы и подразделяются на более мелкие блоки. Для каждой группы можно контролировать счетчик заданий для лучшего и более точного обнаружения ошибок. Как только ведомое устройство отсутствует в группе блоков синхронизации, другие ведомые устройства в группе также отображаются как отсутствующие. Обнаружение происходит сразу же на следующем цикле шины, потому что счетчик заданий проверяется Продолжающийся. С помощью диагностики устройства отсутствующая группа может быть устранена как можно быстрее.

Незатронутые группы остаются работоспособными без какого-либо вмешательства.

### 5.7.11.2.3. Мастер EtherCAT — Обзор

На этой вкладке редактора Конфигурация мастера EtherCAT представлен обзор состояний всех ведомых устройств, которые введены ниже этого ведущего устройства и имеют адрес. Модули не отображаются.

### 5.7.11.2.4. EtherCAT Master — сопоставление входов/выходов

Эта вкладка редактора конфигурации EtherCAT Master предлагает возможность изменить задачу, которая будет использоваться для обновления шины.

### 5.7.11.2.5. EtherCAT Master — объекты IEC

На этой вкладке редактора EtherCAT Master Configuration перечислены объекты, которые разрешают доступ к устройству из приложения IEC.

В онлайн-режиме это используется для мониторинга.

### 5.7.11.2.6. EtherCAT Master — вкладки состояния/информации

На вкладке «Состояние» редактора конфигурации EtherCAT Master содержится информация о состоянии (например, «Работает», «Остановлен») и диагностические сообщения, относящиеся к устройству и внутренней шинной системе. Вкладка «Информация», присутствующая в редакторе EtherCAT Master Configuration, показывает, если она доступна, следующую общую информацию о модуле: Имя, Производитель, Тип, Номер версии, Категория, Номер заказа, Описание, Изображение.

### 5.7.11.3. Конфигурация EtherCAT Slave

Ниже перечислены основные варианты конфигурации ведомого устройства EtherCAT, определенные в файле описания устройства.

#### 5.7.11.3.1. Ведомое устройство EtherCAT — Общее

Ниже представлены общие параметры, найденные на начальном экране Конфигурация EtherCAT Slave. Это поле доступно только в том случае, если режим автонастройки (главный) не включен.

Configuration (Bus) GN\_9483 X

**General**

Expert Process Data

Process Data

Startup Parameters

Log

EtherCAT IEC Objects

Status

Information

**Address**

AutoInc address 0

EtherCAT address 1001

**Additional**

☒ Expert settings

☐ Optional

**Distributed Clock**

Select DC

☐ Enable 4000 Sync unit cycle (μs)

**Sync0**

☐ Enable Sync 0

☐ Sync unit cycle 0 Cycle time (μs)

☐ User-defined 0 Shift time (μs)

**Sync1**

☐ Enable Sync 1

☐ Sync unit cycle 0 Cycle time (μs)

☐ User-defined 0 Shift time (μs)

**Startup Checking**

☒ Check vendor ID

☒ Check product ID

☒ Check revision number

☐ Download expected slot configuration

**DC Cyclic Unit Control: Assign to Local μC**

☐ Cyclic unit ☐ Latch unit 0 ☐ Latch unit 1

**Watchdog**

☐ Set multiplier (Reg. 16#400) 2498

☐ Set PDI watchdog (Reg. 16#410) 1000 = 100.00 ms

☐ Set SM watchdog (Reg. 16#420) 1000 = 100.00 ms

**Identification**

☒ Disabled

☐ Configured station alias (ADO 0x0012) Value 1001

☐ Explicit device identification (ADO 0x0134)

☐ Data Word (2 Bytes) ADO (hex) 16#0

Рисунок 106: Диалоговое окно конфигурации подчиненного устройства EtherCAT

Конфигурация устройства	Описание	Значение по умолчанию	Варианты
Адрес AutoInc	Автоинкрементный адрес (16 бит), определяемый положением ведомого устройства в сети.	-	-65535 до 0
EtherCAT-адрес	Конечный адрес подчиненного устройства, назначаемый ведущим во время запуска. Этот адрес не зависит от положения в сети.	-	1 до 65535
Экспертные настройки	Включить дополнительные настройки ведомого устройства. Варианты.	Не отмечено	Отмечено Не отмечено

## 5. КОНФИГУРАЦИЯ

Конфигурация устройства	Описание	Значение по умолчанию	Варианты
Необязательный	Объявите ведомое устройство как необязательное.	Не отмечено	Отмечено Не отмечено
Выберите округ Колумбия	Показать все Конфигурации распределенных часов, предоставленные устройством. Файл описания.	-	-
Включить распределенные часы	Включите распределенные часы Конфигурация Варианты.	Не отмечено	Отмечено Не отмечено
Единичный цикл синхронизации [мкс]	Показать время цикла, установленное в мастере.	100000	2000 до 1000000
Включить (синхронизация 0)	Включите блок синхронизации Sync 0 Конфигурации.	Не отмечено	Отмечено Не отмечено
Единичный цикл синхронизации (Sync 0)	При выборе этой опции время цикла будет определяться произведением коэффициента и цикла синхронизации.	Не отмечено	Отмечено Не отмечено
Определяется пользователем (синхронизация 0)	Если выбран этот параметр, желаемое время в микросекундах можно установить непосредственно в поле «Время цикла (мкс)».	Не отмечено	Отмечено Не отмечено
Время цикла [мкс] (синхронизация 0)	Показать текущее установленное время цикла.	100000	1 до 2147483647
Время сдвига [мкс] (синхронизация 0)	Время между событиями синхронизации и временем «Действителен вывод» или «Защелка ввода».	0	-2147483648 2147483647
Включить (синхронизация 1)	Включите блок синхронизации Sync 1 Конфигурации.	Не отмечено	Отмечено Не отмечено
Единичный цикл синхронизации (Sync 1)	При выборе этой опции время цикла будет определяться произведением коэффициента и цикла синхронизации.	Не отмечено	Отмечено Не отмечено
Определяется пользователем (синхронизация 1)	Если выбран этот параметр, желаемое время в микросекундах можно установить непосредственно в поле «Время цикла (мкс)».	Не отмечено	Отмечено Не отмечено
Время цикла [мкс] (синхронизация 1)	Показать текущее установленное время цикла.	100000	1 до 2147483647
Время сдвига [мкс] (синхронизация 1)	Время между событиями синхронизации и временем «Действителен вывод» или «Защелка ввода».	0	-2147483648 2147483647
Проверить идентификатор поставщика	Если флажок не установлен, проверка идентификатора поставщика будет отключена.	Не отмечено	Отмечено Не отмечено
Проверить идентификатор продукта	Если флажок не установлен, проверка идентификатора продукта будет отключена.	Не отмечено	Отмечено Не отмечено
Доступ SDO	Установите временную привязку для проверки тайм-аута доступа SDO.	-	0 до 100000

Конфигурация устройства	Описание	Значение по умолчанию	Варианты
<b>I -&gt; P</b>	Установите контрольное время для проверки тайм-аута переключения из режима Init в Pre-Operation.	-	0 до 100000
<b>P -&gt; S/S -&gt; O</b>	Установите контрольное время для проверки тайм-аута переключения из предоперационного в безопасный режим и из безопасного режима в рабочий режим.	-	0 до 100000
<b>Циклический блок</b>	Установите единственный цикл на локальный микропроцессор.	Не отмечено	Отмечено Не отмечено
<b>Блок-защелка 0</b>	Установите блок-защелку 0 на локальный микропроцессор.	Не отмечено	Отмечено Не отмечено
<b>Блок защелки 1</b>	Установите блок-защелку 1 на локальный микропроцессор.	Не отмечено	Отмечено Не отмечено

Таблица 129: Конфигурации ведомых устройств EtherCAT

**Примечания:**

**Адрес AutoInc:** Этот адрес используется только во время запуска, когда ведущий назначает адреса EtherCAT подчиненным. Когда по этому вопросу первая телеграмма проходит через ведомые устройства, каждое быстро читаемое ведомое устройство увеличивает свой адрес AutoInc на

1. Ведомый с адресом 0 в конечном итоге получит данные.

Необязательно: Если ведомое устройство объявлено как необязательное, сообщение об ошибке не будет создано, если устройство не существует в шинной системе. Таким образом, адрес псевдонима станции должен быть определен и записан в EEPROM. Эта опция доступна только в том случае, если опция Autoconfig Master/Slave в настройках EtherCAT Master активирована и если эта функция поддерживается EtherCAT Slave.

Включить распределенные часы: если функция распределенных часов включена, время цикла обмена данными, отображаемое в поле Цикл единицы синхронизации (мкс), будет определяться временем основного цикла. Таким образом, главные часы могут синхронизировать обмен данными внутри сети. Настройки для работы с блоком(ами) синхронизации зависят от ведомого устройства.

Включить синхронизацию 0: Если эта опция активирована, используется модуль синхронизации Sync0. Блок синхронизации описывает набор данных процесса, которыми обмениваются синхронно.

Sync Unit Cycle (Sync 0): Если эта опция активирована, время цикла главного устройства, умноженное на выбранный коэффициент, будет использоваться как время цикла синхронизации для подчиненного устройства. В поле «Время цикла» (мкс) отображается текущее установленное время цикла.

Shift Time: Shift Time описывает время между событиями синхронизации (Sync0, Sync1) и выходом Valid или Input Latch.

раз. Значение WriТаблица, если ведомое устройство поддерживает сдвиг Output Valid или Input Latch.

Enable Sync 1: Если выбран этот параметр, используется блок синхронизации Sync1. Блок синхронизации представляет собой набор данных процесса, которыми обмениваются синхронно.

Sync Unit Cycle (Sync1): Если эта опция активирована, время основного цикла, умноженное на выбранный коэффициент, будет использоваться как время цикла синхронизации для ведомого устройства. В поле «Время цикла» (мкс) отображается текущее установленное время цикла.

Проверить идентификатор поставщика и идентификатор продукта: По умолчанию при запуске системы идентификатор поставщика и/или идентификатор продукта будут проверяться на соответствие текущим настройкам configured. При обнаружении несоответствия шина будет остановлена, и дальнейшие действия выполняться не будут. Это позволяет избежать загрузки ошибочной Конфигурации. Эта опция предназначена для отключения проверки в случае необходимости.

Доступ к SDO: По умолчанию не установлен тайм-аут для действия отправки списка SDO при запуске системы. Однако, если необходимо проверить, превышает ли это действие определенное время, оно должно быть определено (в микросекундах) в этом поле.

I -> P: По умолчанию не установлен тайм-аут для перехода состояния из Init в Pre-Operational. Однако, если необходимо проверить, превышает ли это действие определенное время, оно должно быть определено (в микросекундах) в этом поле.

## 5. КОНФИГУРАЦИЯ

P -> S / S -> O: По умолчанию не установлен тайм-аут для перехода состояния из Pre-Operational в Safe-Operational и из Safe-Operation в Operational. Однако, если необходимо проверить, превышает ли это действие определенное время, оно должно быть определено (в микросекундах) в этом поле.

Управление блоком циклов постоянного тока: выберите желаемую(ые) опцию(и) относительно функций распределенных часов, чтобы определить, какие из них должны быть назначены локальному микропроцессору. Управление осуществляется в регистре 0x980 подчиненного устройства EtherCAT. Возможные настройки: Циклический блок, Блок-фиксатор 0, Блок-фиксатор 1.

Включить: Если параметр Необязательный не активирован, этот параметр можно активировать, если он явно поддерживается устройством Описание ведомого устройства. Это позволяет напрямую назначать адрес-псевдоним, чтобы получить адрес ведомого устройства независимо от его положения на шине. Если опция Необязательно активирована, этот флажок отключен.

### 5.7.11.3.2. Ведомое устройство EtherCAT — данные процесса

На вкладке Process Data редактора конфигулятора EtherCAT Slave отображаются данные процесса ввода и вывода подчиненного устройства, каждое из которых определяется именем, типом и индексом в файле описания устройства, как показано на рисунке ниже.

Выбранный вход (для чтения) и выход (для записи) устройства доступны в диалоговом окне EtherCAT Slave — сопоставление входов/выходов в качестве входов и выходов ПЛК, на которые могут быть сопоставлены переменные проекта.

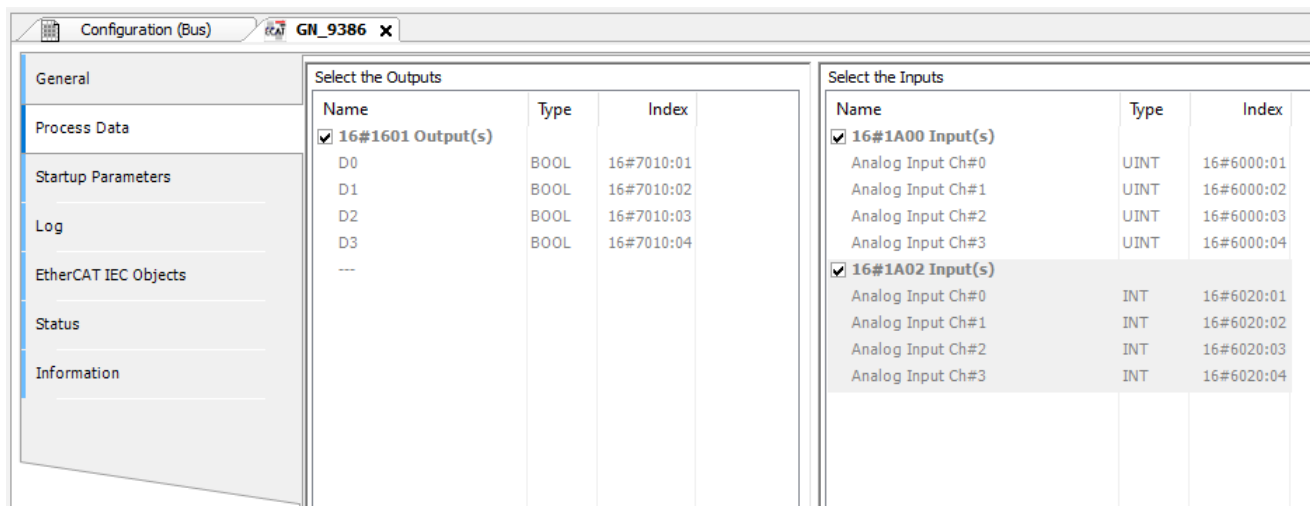


Рисунок 107: Диалоговое окно обработки данных

Диалоговое окно Expert Process Data будет доступно в редакторе конфигурации EtherCAT Slave, только если активирована опция Enable Expert Settings. Он предоставляет другое, более подробное представление о данных процесса, дополняя то, что представлено на вкладке «Данные процесса». Кроме того, в этом диалоговом окне можно активировать загрузку назначения PDO и конфигурации PDO.

#### ВНИМАНИЕ

Если ведомое устройство не принимает конфигурацию PDO, оно останется в предрабочем состоянии, и обмен данными в реальном времени будет невозможен.

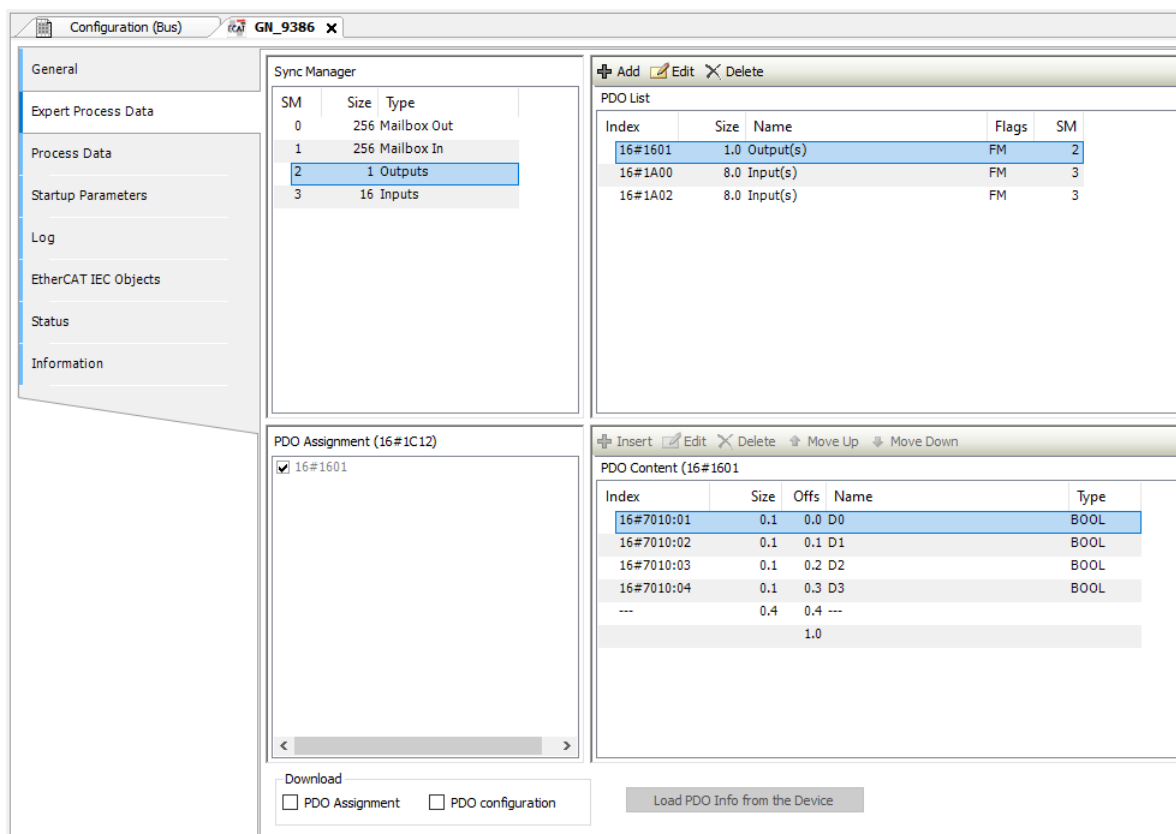


Рисунок 108: Диалоговое окно данных экспертного процесса

Этот диалог разделен на четыре раздела и два варианта:

**Sync Manager:** Список Sync Manager с размером данных и типом PDO.

**Назначение PDO:** список PDO, назначенных выбранному диспетчеру синхронизации. Флажок активирует PDO и создаются каналы ввода/вывода. Это похоже на простую PDO Конфигурация окон. Здесь могут быть включены или отключены только PDO.

**Список PDO:** Список всех PDO, определенных в файле описания устройства. Отдельные PDO можно удалить, отредактировать или добавить, выполнив соответствующую команду из контекстного меню.

**PDO Content:** Отображает содержимое PDO, выбранного в разделе выше. Записи можно удалять, редактировать или добавлять, выполнив соответствующую команду из контекстного меню.

**Назначение PDO:** если активировано, команда записи CoE будет добавлена к индексу 0x1CXX для записи конфигурации PDO 0x16XX или 0x1A00.

**Конфигурация PDO:** если активировано, будет добавлено несколько команд записи CoE для записи отображения PDO на ведомое устройство.

#### ВНИМАНИЕ

Если ведомое устройство не поддерживает конфигурацию PDO, загрузка может привести к ошибке ведомого устройства. Эта функция должна использоваться только экспертами.

## 5.7.11.3.3. Ведомое устройство EtherCAT — редактирование списка PDO

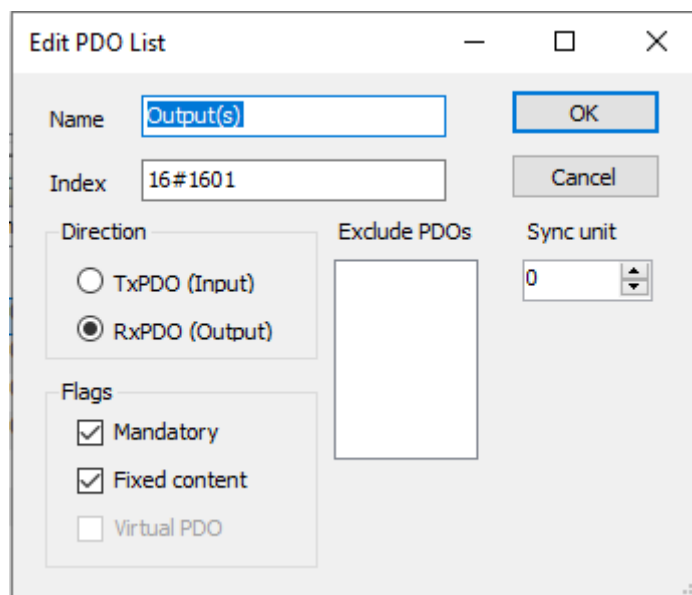


Рисунок 109: Диалоговое окно редактирования списка PDO

Это диалоговое окно открывается через контекстное меню из области списка PDO, представленного на рисунке 108. Ниже приведены некоторые пояснения по Конфигурация Варианты, представленные в этом диалоговом окне.

Имя: Имя входа PDO.

Индекс: индекс редактируемого PDO.

TxPDO (вход): если активировано, PDO будет передаваться от ведущего к ведомому. RxPDO (выход): если активировано, PDO будет передан от ведомого к ведущему. Обязательно: PDO необходим и не может быть снят в области назначения PDO.

Фиксированное содержимое: содержимое PDO является фиксированным и не может быть изменено. Невозможно добавлять записи в содержимое PDO.

панель.

Виртуальный PDO: зарезервирован для использования в будущем.

Исключить PDO: можно определить список PDO, которые могут или не могут быть выбраны вместе с редактируемым PDO в области назначения PDO или на вкладке Process Data. Если объект PDO отмечен в этом списке, его нельзя выбрать, он становится серым в области назначения PDO, когда выбран объект PDO в редакции.

SyncUnit: идентификатор PDO Sync Manager, которому должен быть назначен.

## 5.7.11.3.4. Ведомое устройство EtherCAT — параметры запуска

Во вкладке Startup Parameters можно определить параметры для устройства, которые будут передаваться SDO (Service Data Objects) или IDN при запуске системы. Варианты, доступные на этой вкладке, а также возможности доступа различаются в зависимости от используемого подчиненного устройства EtherCAT и представлены в файле описания устройства.

## 5.7.11.3.5. Ведомое устройство EtherCAT — сопоставление ввода-вывода

На этой вкладке редактора конфигурации EtherCAT Slave можно назначить переменные проекта входам или выходам EtherCAT. Таким образом, пользовательское приложение может управлять переменными EtherCAT Slave.

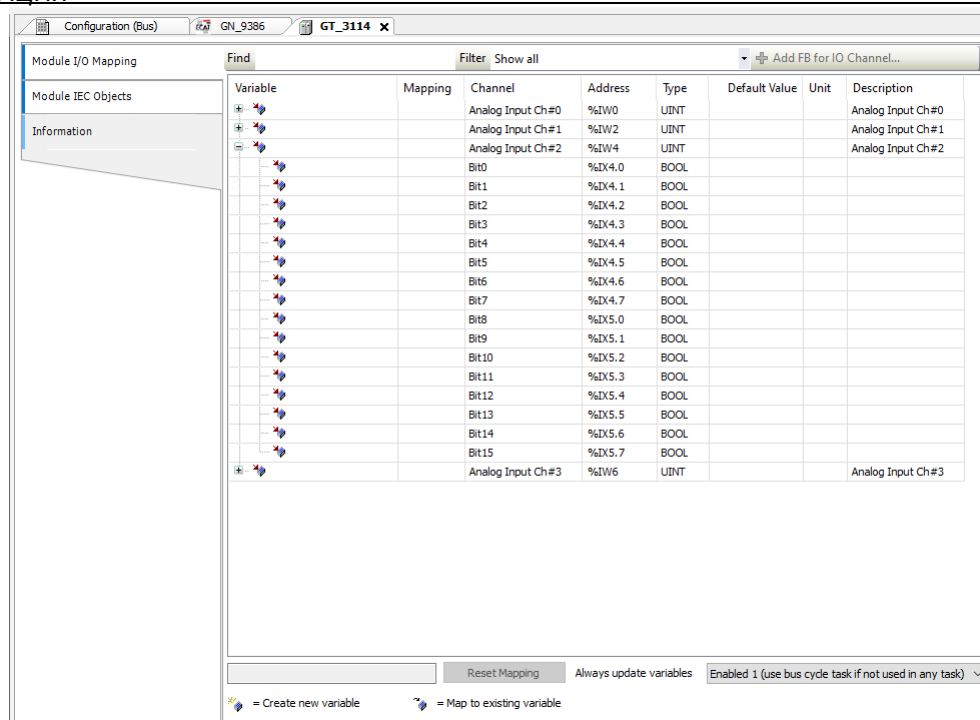


Рисунок 110: Диалоговое окно сопоставления подчиненных входов/выходов

#### 5.7.11.3.6. Ведомое устройство EtherCAT — статус и информация

На вкладке «Статус» ведомого устройства EtherCAT отображается информация о состоянии (например, «Работает», «Остановлен») и диагностические сообщения для конкретного устройства, а также на используемой плате и внутренней шинной системе.

Вкладка Информация, представленная в редакторе Конфигурация EtherCAT Slave, показывает, если она доступна, следующую общую информацию о модуле: Имя, Производитель, Тип, Версия, Категории, Номер заказа, Описание, Изображение.

### 5.7.12. EtherNet/IP

EtherNet/IP — это протокол архитектуры ведущий-ведомый, состоящий из сканера EtherNet/IP (ведущего) и одного или нескольких адаптеров EtherNet/IP (ведомых).

Протокол Ethernet/IP основан на CIP (Общем промышленном протоколе), который имеет две основные цели: Передача ориентированных на управление данных, связанных с устройствами ввода-вывода, и другой управляемой системной информации, такой как параметры конфигурации и диагностика. Первый выполняется с помощью неявных сообщений, а второй — с помощью явных сообщений.

Их система выполнения может действовать как сканер или адаптер. NET-интерфейс каждого ЦП поддерживает только один экземпляр EtherNet/IP, и его нельзя разместить на модуле расширения Ethernet.

Экземпляр адаптера EtherNet/IP поддерживает неограниченное количество модулей или байтов ввода/вывода. В эти модули можно добавлять переменные типов: BYTE, BOOL, WORD, DWORD, LWORD, USINT, UINT, UDINT, ULINT, SINT, INT, DINT,

LINT, REAL и LREAL. Диапазон MainTask устройства, работающего как адаптер, должен быть меньше или равен RPI.

#### ВНИМАНИЕ

EtherNet/IP нельзя использовать вместе с Ethernet NIC-Teaming, а не с резервированием полукластеров.

#### ВНИМАНИЕ

Для EtherNet/IP требуется циклическая основная задача, а это означает, что ее нельзя использовать с какой-либо конфигурацией, которая устанавливает основную задачу как независимую. Например, проект базового профиля не поддерживает EtherNet/IP.

**ВНИМАНИЕ**

Чтобы избежать проблем со связью, сканер EtherNet/IP может иметь адаптеры, настроенные только в одной и той же подсети.

## 5.7.12.1. EtherNet/IP-интерфейс

Чтобы добавить сканер или адаптер EtherNet/IP, необходимо добавить интерфейс EtherNet/IP под интерфейсом NET. Это можно сделать через команду Add Device. Под этим интерфейсом EtherNet/IP можно добавить сканер или адаптер.

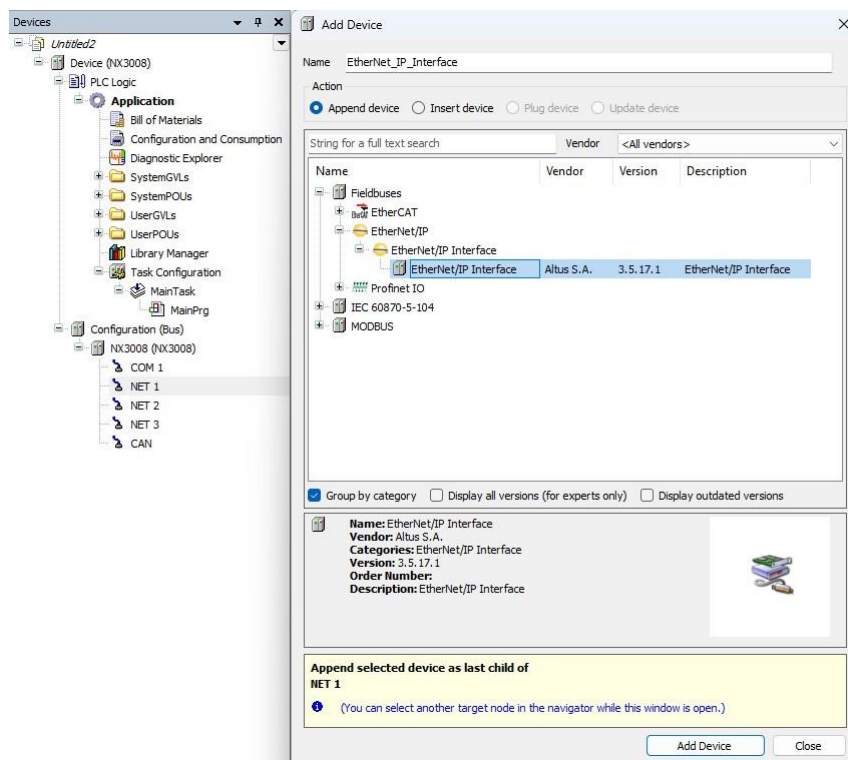


Рисунок 111: Добавление интерфейса EtherNet/IP

## 5. КОНФИГУРАЦИЯ

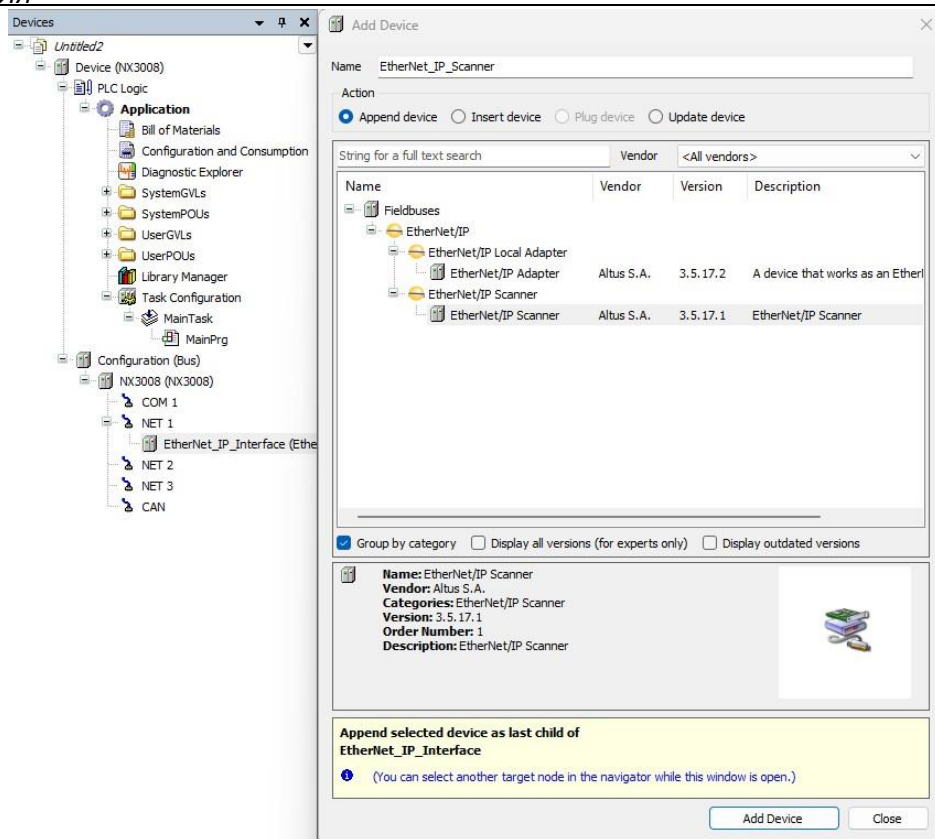


Рисунок 112: Добавление адаптера EtherNet/IP или сканера

### 5.7.12.2. Конфигурация сканера EtherNet/IP

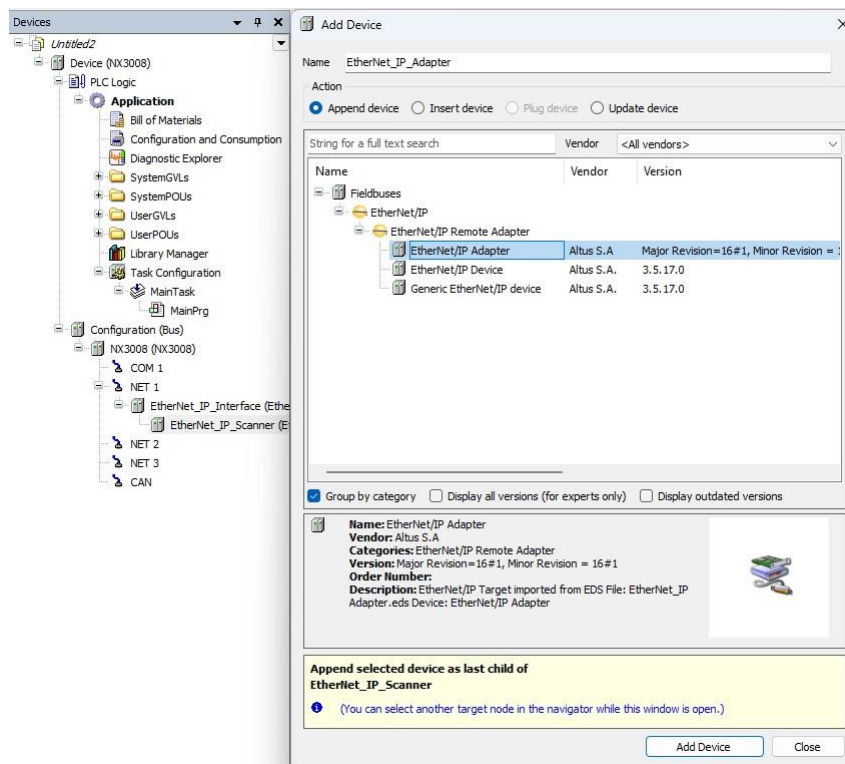


Рисунок 113: Добавление адаптера EtherNet/IP под сканер

#### 5.7.12.2.1. Общие

5. КОНФИГУРАЦИЯ

После открытия адаптера, заявленного под сканером, его можно подключить по мере необходимости. Первая вкладка — «Общие», на ней можно указать «Адрес IP» и «Электронный ключ». Эти параметры должны быть отмечены или сняты, если используемый адаптер установлен на MasterTool. В противном случае, если используемый адаптер относится к типу Generic, то поля Vendor ID, Device Тип, Product code, Major version и Minor version должны быть заполнены правильной информацией, а флажки отмечены столько, сколько необходимо.

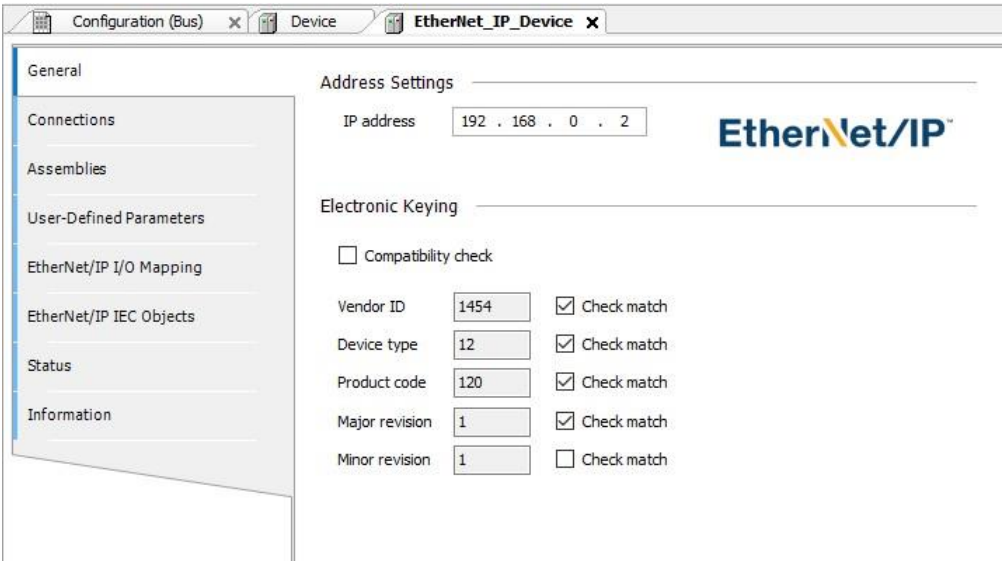


Рисунок 114: Вкладка «Общие» EtherNet/IP

5.7.12.2.2. Соединения

В верхней части вкладки «Подключения» отображается список всех подключений configured. Когда в файле EDS есть соединение с монопольным владельцем, оно вставляется автоматически при добавлении адаптера. Данные Конфигурации для этих подключений можно изменить в нижней части экрана.

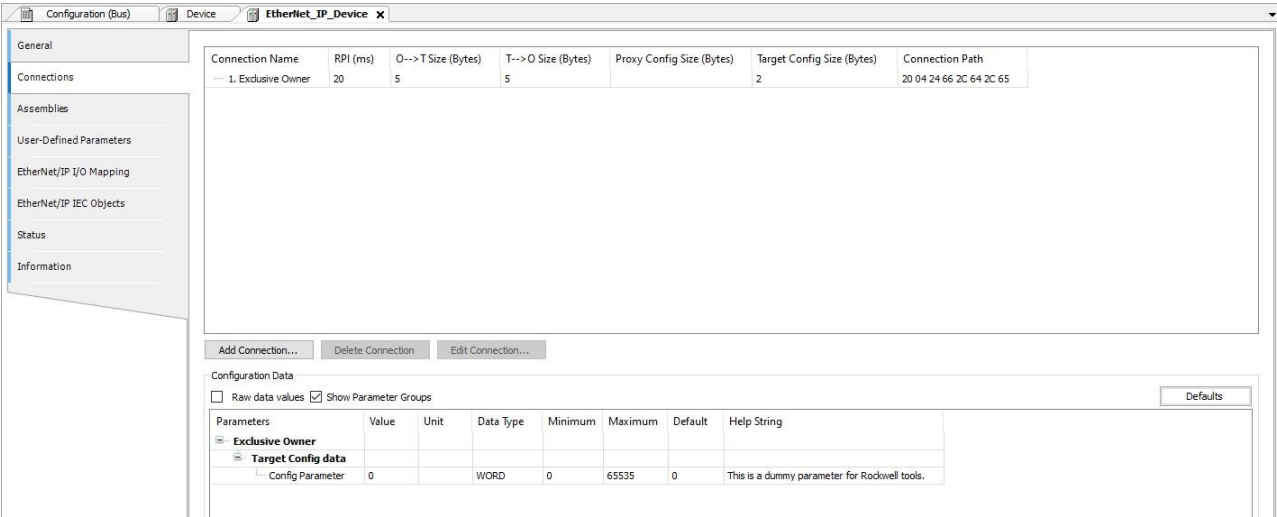


Рисунок 115: Вкладка «Соединение EtherNet/IP»

Примечания:

- Для подключения двух или более сканеров EtherNet/IP к одному и тому же удаленному адаптеру:
- 1. Только один из Сканеров может установить соединение с монопольным владельцем.
  - 2. Для сканеров должно быть установлено одинаковое значение RPI(мс).
- Данные конфигурации определяются в файле EDS. Данные передаются на удаленный адаптер при открытии соединения.

Конфигурация	Описание	Значение по умолчанию	Варианты
--------------	----------	-----------------------	----------

## 5. КОНФИГУРАЦИЯ

RPI (мс)	Request Packet Interval: интервал обмена входными и выходными данными.	10 мс	Кратность интервала MainTask
О -> Размер Т (байты)	Размер данных производителя от сканера до адаптера (О - > Т)	Зависит от ЭЦП адаптера	0 - 65527
Т -> О Размер (байты)	Размер потребительских данных от адаптера до сканера (Т -> О)	Зависит от ЭЦП адаптера	0 - 65531
Размер конфигурации №1 (байты)	Размер данных конфигурации 1.	Зависит от ЭЦП адаптера	-
Размер конфигурации № 2 (байты)	Размер данных конфигурации 2.	Зависит от ЭЦП адаптера	-
Путь подключения	Адрес объектов Конфигурация - входных объектов - выходных объектов.	Зависит от ЭЦП адаптера	-

Таблица 130: Параметры подключения EtherNet/IP

Для добавления новых подключений есть кнопка Добавить подключение..., которая открывает окно Новое подключение. В этом окне можно настроить новый тип подключения из предопределенных на ЭЦП адаптера или универсальное подключение с нуля.

Рисунок 116: Окно нового подключения EtherNet/IP

## 5.7.12.2.3. Сборки

В верхней части вкладки «Сборки» отображается список всех настроенных соединений. Когда соединение выбрано, связанные входы и выходы отображаются в нижней части вкладки.

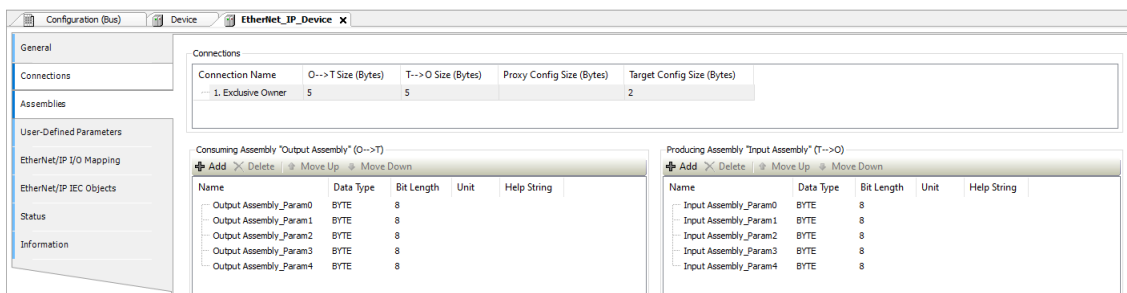


Рисунок 117: Сборки EtherNet/IP

Выходная сборка и входная сборка:

Конфигурация	Описание
Добавлять	Открывает диалоговое окно «Добавить ввод/вывод».
Удалить	Удаляет все выбранные входы/выходы.
Вверх	Перемещает выбранный вход/выход в списке.
Вниз	Порядок в списке определяет порядок в сопоставлении ввода/вывода.

Таблица 131: Вкладка сборок EtherNet/IP

Диалоговое окно «Добавить вводы/выводы»:

Конфигурация	Описание
Имя	Имя входа/выхода, которое нужно вставить.
Строка справки	
Тип данных	Тип вставляемого входа/выхода. Этот тип также определяет его битовую длину.
Длина бита	Данное значение не может быть дополнено

Таблица 132: Окно EtherNet/IP «Добавить вход/выход»

## 5.7.12.2.4. Сопоставление ввода/вывода EtherNet/IP

На вкладке «Отображение ввода/вывода» в столбце «Переменная» отображается имя автоматически сгенерированного экземпляра адаптера в разделе «Объекты IEC». Таким образом, приложение может получить доступ к экземпляру. Здесь переменные проекта сопоставляются с входами и выходами адаптера.

## 5.7.12.3. Конфигурация адаптера EtherNet/IP

Для адаптера EtherNet/IP требуются модули Ethernet/IP. Модули будут предоставлять сопоставления ввода/вывода, которыми пользовательское приложение может управлять через адреса %I или %Q в соответствии с его Конфигурацией.

## 5. КОНФИГУРАЦИЯ

Новые адаптеры можно установить на MasterTool с помощью файлов EDS. Конфигурация Варианты может отличаться в зависимости от файла описания устройства добавленного адаптера.

### 5.7.12.3.1. Общая информация

Первая вкладка адаптера EtherNet/IP — это вкладка «Общие». Здесь вы можете установить параметры электронного ключа, используемого в сканере, для проверки совместимости. В этой вкладке также можно установить ЭЦП устройства непосредственно в репозиторий устройства MasterTool или экспортировать его.

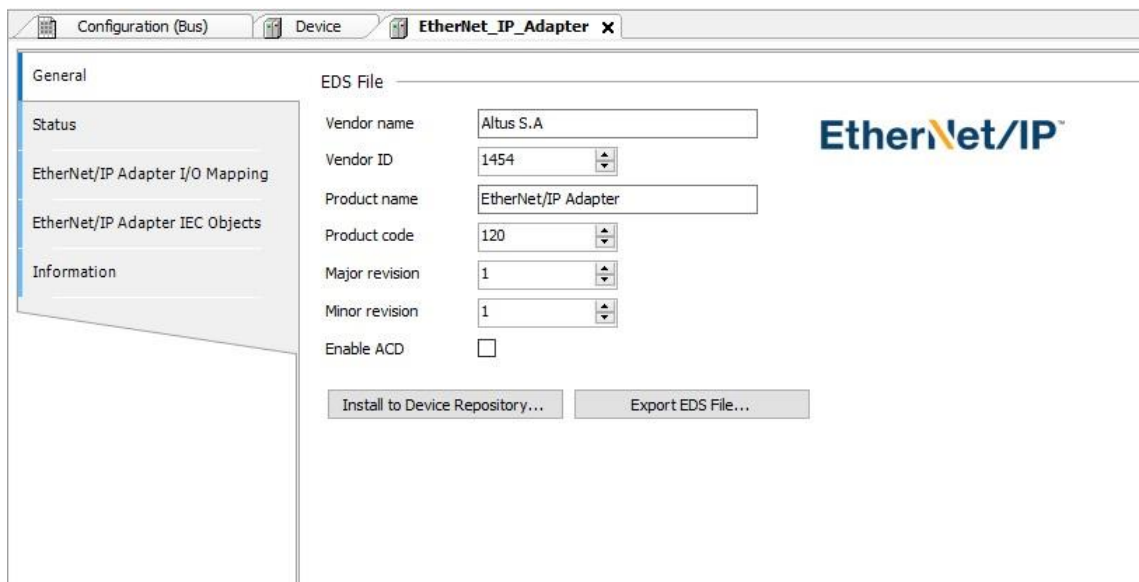


Рисунок 118: Вкладка «Общие» EtherNet/IP

### 5.7.12.3.2. Адаптер EtherNet/IP: сопоставление входов/выходов

На вкладке EtherNet/IP-входов/выходов можно указать, какую задачу цикла шины будет выполнять адаптер.

### 5.7.12.4. Конфигурация модуля EtherNet/IP

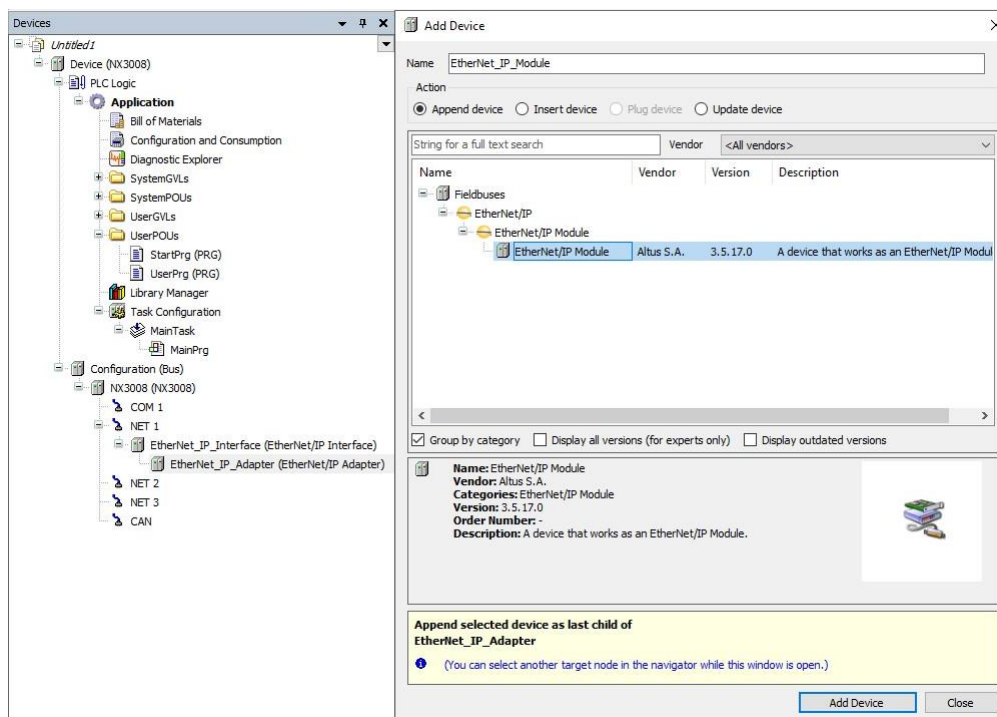


Рисунок 119: Добавление модуля EtherNet/IP под адаптер

#### 5.7.12.4.1. Сборки

Параметры вкладки «Общие» модуля подчиняются тем же правилам, что и в таблицах 131 и 132.

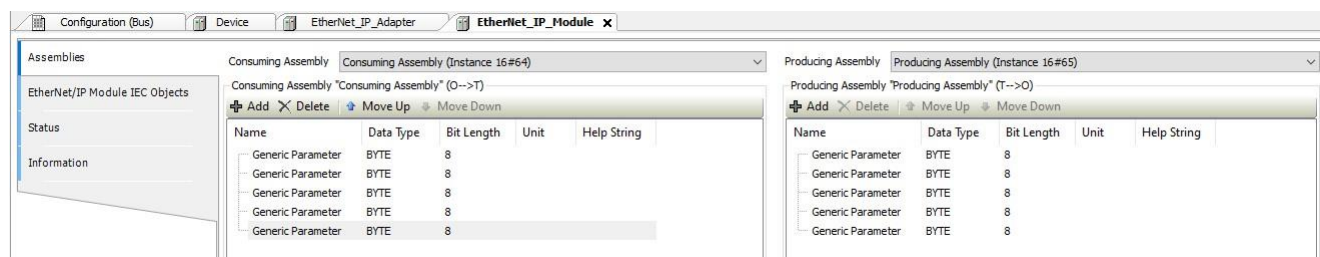


Рисунок 120: Вкладка "Сборки модулей EtherNet/IP"

#### 5.7.12.4.2. Модуль EtherNet/IP: сопоставление ввода-вывода

На вкладке I/O Mapping в столбце Variable отображается имя автоматически сгенерированного экземпляра адаптера. Таким образом, пользовательское приложение может получить доступ к экземпляру.

#### 5.7.13. Менеджер CANopen

CANopen — это протокол, основанный на шине CAN, который обеспечивает быстрое обновление ввода-вывода (около 5 мс для сети 1000 кбит/с с несколькими ведомыми устройствами) с простой инфраструктурой физической шины витой пары.

CANopen Manager (мастер) отвечает за управление ведомыми устройствами, управление их рабочим состоянием и обмен данными ввода/вывода и другими служебными данными. По умолчанию действия протокола диспетчера CANopen (цикл шины) выполняются в контексте MainTask, сохраняя его синхронность с выполнением кода приложения.

Конфигурация сети CANopen выполняется с поддержкой файлов EDS, которые описывают данные ввода-вывода и сервисные объекты (PDO и SDO) ведомого устройства и должны быть предоставлены производителем устройства. Кроме того, прикладная библиотека под названием CiA405 снабжена функциональными блоками, которые позволяют выполнять несколько специальных действий, таких как изменение состояния подчиненного устройства (NMT), получение аварийного объекта, запрос состояния подчиненного устройства и выполнение команд чтения/записи SDO. Полное описание библиотеки CiA405 можно найти в интерактивной справке (F1) MasterTool IEC XE.

Следует уделить особое внимание физической длине шины и выбранной скорости передачи данных. Следующая таблица показывает максимальную длину шины, которую можно безопасно использовать с заданной скоростью передачи данных:

#### ВНИМАНИЕ

- Допускается только один экземпляр CANopen Manager для каждого проекта.
- Хотя спецификация CANopen позволяет использовать до 127 узлов (включая диспетчер), приложения с NX3008 не должны превышать 64 ведомых устройства.

Скорость передачи данных	Максимальная длина шины
1000кбит/с	25 м
500 кбит/с	100 м
250 кбит/с	250 м
125 кбит/с	500 м
100 кбит/с	700 м
< 50 кбит/с	1000 м

Таблица 133: Скорость передачи данных X Длина шины

#### 5.7.13.1. Установка и подключение устройств CANopen

Конфигурация сети CANopen использует ту же стандартную процедуру конфигурации других полевых шин в MasterTool IEC XE.

## 5. КОНФИГУРАЦИЯ

Чтобы добавить CANopen Manager, щелкните правой кнопкой мыши интерфейс CAN и выберите «Добавить устройство». Расширьте элементы, пока не найдете устройство CANopen\_Manager, и нажмите кнопку «Добавить устройство». Устройство CANopen Manager появится под интерфейсом CAN, как показано на следующем рисунке:

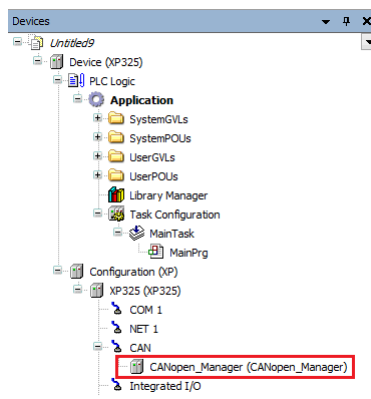


Рисунок 121: Добавление менеджера CANopen

Чтобы добавить ведомое устройство CANopen, сначала необходимо установить его в хранилище устройств. Для этого перейдите в Инструменты -> Хранилище устройств и установите файл EDS устройства.

После этого щелкните правой кнопкой мыши устройство CANopen\_Manager и нажмите «Добавить устройство». Найдите нужные устройства и нажмите кнопку «Добавить устройство», как показано на следующем рисунке:

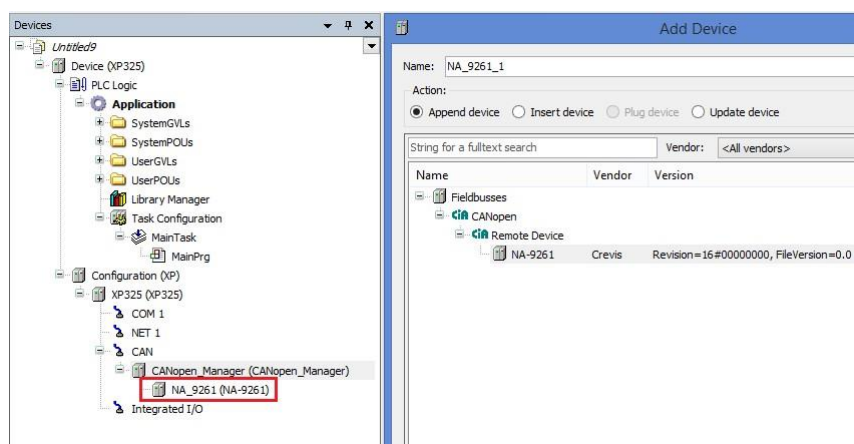


Рисунок 122: Добавление подчиненного устройства CANopen

### 5.7.13.2. Конфигурация менеджера CANopen

CANopen Manager поставляется с готовой к использованию конфигурацией (значения по умолчанию). Как правило, достаточно просто установить правильную скорость передачи данных и адрес ведомого устройства, чтобы сеть работала.

Основные параметры CANopen менеджера находятся на вкладке Общие:

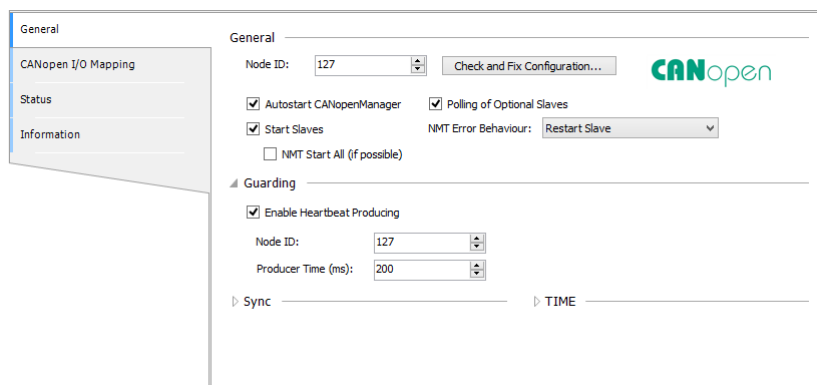


Рисунок 123: Основные параметры CANopen Manager

Подробное описание общих параметров CANopen Manager можно найти в разделе Device Editors -> CANopen интерактивной справки MasterTool IEC XE (F1).

Кроме того, вкладка Отображение ввода/вывода CANopen позволяет изменить задачу цикла шины:

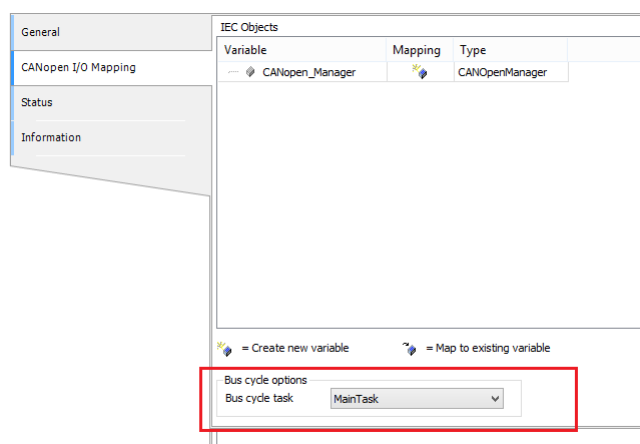


Рисунок 124: Настройка задачи цикла шины CANopen Manager

По умолчанию, задача цикла шины предназначена для использования MainTask. Это рекомендуемая настройка для большинства приложений. Изменение этого параметра требуется только в очень конкретном сценарии, который требует реализации критического по времени цикла управления с использованием ввода-вывода CANopen (скажем, 5 мс), который не может быть выполнен в MainTask из-за тяжелого кода приложения.

### 5.7.13.3. Конфигурация ведомого устройства CANopen

Конфигурация удаленных устройств CANopen (ведомых) разделена на первых четырех вкладках, показанных на следующем рисунке:

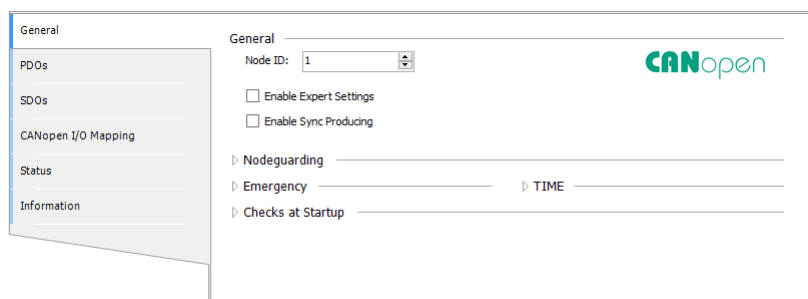


Рисунок 125: Параметры ведомого устройства CANopen

Вкладка «Общие» содержит адрес ведомого устройства (идентификатор узла), параметры защиты узла и аварийного объекта. Вкладка PDO содержит Конфигурацию данных процесса (данные ввода-вывода), которые будут обмениваться.

Вкладка SDO содержит объекты SDO, которые могут быть выбраны для доступа с помощью функционального блока чтения/записи SDO, предоставляемого библиотекой CiA405.

Подробное описание параметров ведомого устройства CANopen можно найти в разделе «Редакторы устройств -> CANopen интерактивной справки Master-Tool IEC XE (F1).

## 5.8. Коммуникационная производительность

### 5.8.1. Сервер MODBUS

Устройства MODBUS, настраиваемые в ЦП Nexto, работают в фоновом режиме с приоритетом ниже пользовательского приложения и циклически. Таким образом, их производительность варьируется в зависимости от оставшегося времени с учетом разницы между интервалом и временем, которое требуется приложению для запуска. Например, устройство MODBUS в приложении, которое запускается каждые 100 мс, с временем работы 50 мс, будет иметь более низкую производительность, чем приложение, работающее с интервалом от 50 до 200 мс. Это происходит потому, что в последнем случае у ЦП будет больше времени между каждым циклом MainTask для выполнения задач с более низким приоритетом.

## 5. КОНФИГУРАЦИЯ

Также необходимо учитывать количество циклов, которое требуется устройству, ведомому устройству или серверу для ответа на запрос. Чтобы обработать и передать ответ, ведомому устройству MODBUS RTU потребуется два цикла (время цикла задачи MODBUS), тогда как задача сервера MODBUS Ethernet занимает только один цикл. Но это минимальное время между получением запроса и ответом. Если запрос отправляется сразу после выполнения задачи, время цикла MODBUS может быть в 2 или 3 раза больше времени цикла для ведомого устройства MODBUS и в 1-2 раза больше времени цикла для сервера MODBUS.

В этом случае: Максимальное время ответа = 3 \* (время цикла) + (время выполнения задач) + (время межкадровых символов) + (задержка отправки).

Например, для задачи сервера MODBUS Ethernet с циклом 50 мс, приложения, которое выполняется в течение 60 мс каждые 100 мс, сервер может выполнять только один цикл между каждым циклом приложения. С другой стороны, при использовании того же приложения, работающего в течение 60 мс, но с интервалом в 500 мс, MODBUS работает лучше, потому что пока приложение не запущено, оно будет запускаться каждые 50 мс и только каждый цикл MainTask его будет дольше бегать. В этих случаях наихудшей производительностью будет сумма времени выполнения пользовательского приложения со временем цикла задачи MODBUS.

Для ведущего и клиентского устройств принцип работы точно такой же, но с учетом времени опроса отношения MODBUS, а не времени цикла задачи MODBUS. В этих случаях наихудшая производительность связи будет достигнута после времени опроса вместе с временем выполнения пользовательского приложения.

Важно подчеркнуть, что количество запущенных устройств MODBUS также меняет его производительность. В пользовательском приложении со временем выполнения 60 мс и интервалом 100 мс у ЦП остается 40 мс для выполнения всех задач с более низким приоритетом. Следовательно, ЦП только с одним сервером Ethernet MODBUS будет иметь более высокую производительность, чем ЦП, использующий четыре таких устройства.

### 5.8.1.1. Локальные интерфейсы ЦП

Для устройства MODBUS Ethernet Server мы можем утверждать, что устройство способно отвечать на  $x$  запросов в секунду. Или, другими словами, Сервер способен передавать  $n$  байтов в секунду, в зависимости от размера каждой заявки. Чем меньше время цикла задачи сервера MODBUS, тем выше влияние количества соединений на его скорость ответов. Однако для времени цикла менее 20 мс это влияние не является линейным, и для информации необходимо просмотреть Таблицу ниже.

Таблица ниже иллюстрирует количество запросов, на которые сервер MODBUS, вставленный в локальный интерфейс Ethernet, способен ответить, в зависимости от времени цикла, настроенного для задачи MODBUS, и количества активных соединений:

Количество активных подключений	Отвеченных запросов в секунду с рабочим циклом MODBUS в 5 мс	Отвеченных запросов в секунду с рабочим циклом MODBUS в 10 мс	Отвеченных запросов в секунду с рабочим циклом MODBUS в 20 мс
1 подключение	185	99	50
2 подключения	367	197	100
4 подключения	760	395	200
7 подключений	1354	695	350
10 подключений	1933	976	500

Таблица 134: Скорость связи сервера MODBUS на локальном интерфейсе

#### ВНИМАНИЕ

Коммуникационные характеристики, упомянутые в этом разделе, являются лишь примерами использования ЦП только с одним устройством MODBUS TCP Server, без какой-либо логики, которая должна выполняться внутри приложения, что может задержать связь. Поэтому эти показатели необходимо принять за максимальные

Для времени цикла, равного или превышающего 20 мс, увеличение частоты ответов является линейным и может быть рассчитано с использованием уравнения:

$$N = C \times (1/T)$$

Где:

$N$  — среднее количество ответов в секунду;  $C$  — количество активных подключений;

$T$  — интервал задачи MODBUS в секундах.

В качестве примера сервера MODBUS с одним активным соединением и временем цикла 50 мс мы получаем:  $C = 1$ ;  $T = 0,05$  с;

$$N = 1 \times (1/(0,05))$$

$$N = 20$$

То есть в данной Конфигурации Сервер MODBUS отвечает, в среднем, на 20 заявок в секунду.

## 5. КОНФИГУРАЦИЯ

Если полученное значение умножить на количество байт в каждой заявке, то получим скорость передачи  $n$  байт в секунду.

### 5.8.1.2. Удаленные интерфейсы

Производительность сервера MODBUS устройства в одном удаленном интерфейсе Ethernet аналогична производительности в локальных интерфейсах ЦП.

Однако из-за времени обмена данными между ЦП и модулями максимальная производительность ограничена. Только для одного активного соединения количество ответов ограничено максимум 18 ответами в секунду. При более активных подключениях количество ответов будет увеличиваться линейно, точно так же, как и в случае с локальными интерфейсами, но ограничено максимальным значением 90 ответов в секунду. Итак, для удаленного интерфейса Ethernet у нас будут следующие формы для расчета его производительности:

Для  $T \leq 55$  мс используется:

$$N = C \times (18,18 - (18,18 / (0,055 \times 1000)))$$

А для  $T \geq 55$  мс используется:

$$N = C \times (Z - (Z / (T \times 1000)))$$

Где  $N$  — среднее количество ответов в секунду,  $C$  — количество активных подключений, а  $T$  — время цикла задачи MODBUS (в секундах).

Пользователь должен обратить внимание на то, что максимальная производительность устройства MODBUS Server в одном удаленном интерфейсе Ethernet составляет 90 ответов на заявки в секунду.

### 5.8.2. Сервер OPC-UA

В примечаниях по применению NAPI65 более подробно анализируется производительность связи OPC UA, в том числе рассматривается потребление пропускной способности связи Ethernet. В этом примечании по применению также обсуждаются принципы работы протокола OPC UA.

## 5.9. Производительность системы

В случаях, когда приложение имеет только одну пользовательскую задачу MainTask, отвечающую за выполнение одной программной единицы типа «Программа», называемой MainPrg (как в случае с одним профилем), ПЛК потребляет определенное количество времени для обработки задачи. В то время мы называем это временем выполнения.

В приложении среднее время выполнения приложения можно узнать с помощью MasterTool IEC XE в элементе «Устройство» его дерева устройств следующим образом:

Логика ПЛК->Приложение->Конфигурация задач на вкладке Монитор, столбец Среднее время цикла.

Пользователь должен обращать внимание на время цикла, чтобы оно не превышало 80% интервала, установленного в пользовательской задаче MainTask. Например, в приложении, где интервал составляет 100 мс, соответствующее время цикла составляет до 80 мс. Это связано с тем, что ЦП требуется время для выполнения других задач, таких как обработка связи, обработка дисплея и карты памяти, и эти задачи выполняются в пределах диапазона (оставшиеся 20% времени цикла).

#### ВНИМАНИЕ

При очень большом времени цикла (обычно более 300 мс) даже при соблюдении значения 80 % может произойти сокращение времени отклика дисплея и диагностической клавиши. В случае несоблюдения 80-процентного значения и времени выполнения пользовательской задачи ближе или превышает интервал, установленный для основной задачи, экран и кнопка диагностики не могут реагировать, поскольку их приоритет в системе ниже, чем у пользовательских задач. Если в ЦП загружено приложение с ошибками, может потребоваться его перезапуск без загрузки этого приложения, как показано в разделе «Системный журнал».

#### ВНИМАНИЕ

Системные журналы ЦП серии Nexto, начиная с версии прошивки 1.4.0.33, теперь перезагружаются в случае сброса ЦП или перезагрузки системы выполнения, то есть вы можете просмотреть более старые журналы при возникновении одного из этих условий.

### 5.9.1. Время сканирования входа/выхода

Для проекта, который использует цифровые модули ввода/вывода, будучи вставленными в шину и объявленными в проекте, время MainTask будет увеличиваться в соответствии с количеством модулей. Таблица ниже иллюстрирует среднее время, которое добавляется к MainTask:

Объявленные модули в шине	Добавленное время во время цикла MainTask (мкс)
5	300

---

10	700
20	1000

Таблица 135: Время сканирования входа/выхода

В проектах, использующих удаленный ввод/вывод, например, с использованием ведущего модуля NX5001 PROFIBUS-DP, необходимо обращаться к руководству по соответствующему модулю для получения информации о производительности и влиянии модуля на выполнение пользовательских задач.

#### 5.9.2. Карта памяти

Передача данных с использованием карты памяти выполняется ЦП в фоновом режиме, так как приоритет отдается выполнению пользовательского приложения и обработке связи. Таким образом, передача файлов на карту может страдать дополнительным значительным временем, зависящим от Cycle Time пользовательского приложения.

Время, необходимое для чтения/записи файлов на карту, будет напрямую зависеть от времени цикла пользовательского приложения, поскольку это приложение имеет приоритет при выполнении.

Дополнительную информацию об использовании карты памяти см. в разделе «Карта памяти».

### 5.10. Часы реального времени

ЦП серии Nexto имеют внутренние часы, которые можно использовать с помощью библиотеки NextoStandard.lib. Эта библиотека автоматически загружается при создании нового проекта (процедуру вставки библиотеки см. в разделе «Библиотеки»). На рисунке ниже показано, как включить блоки в проект:

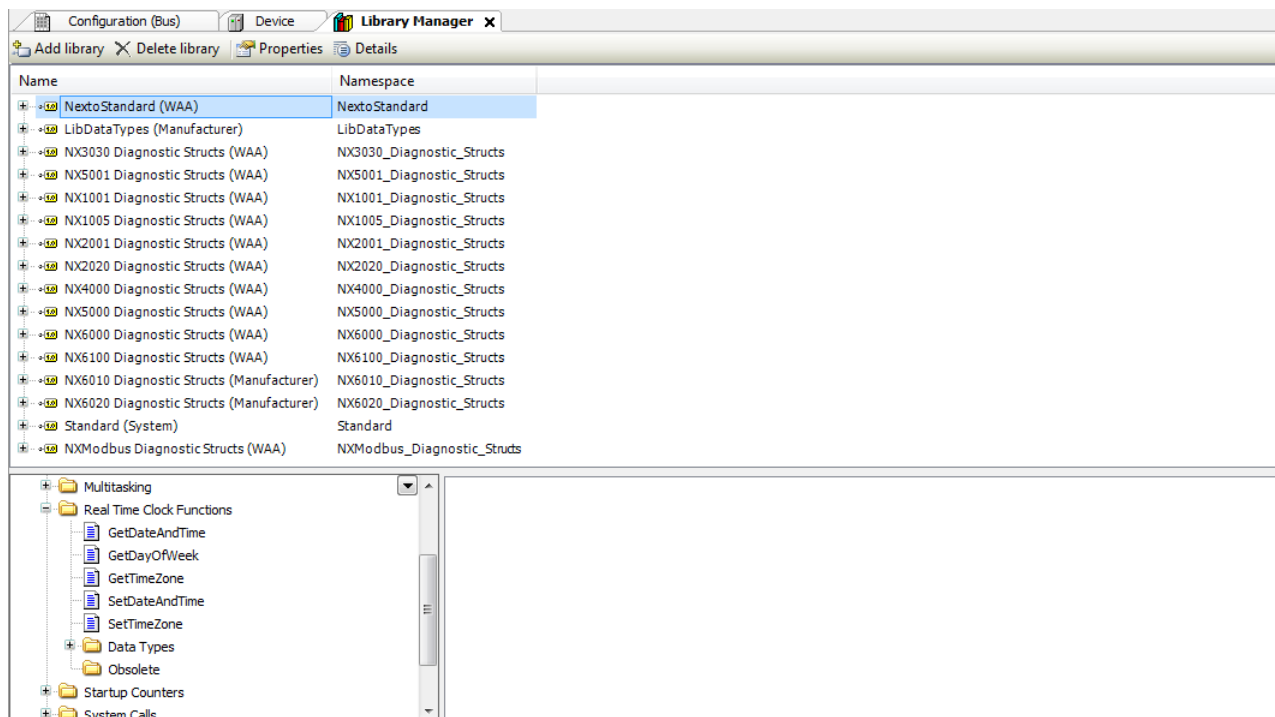


Рисунок 126: Блоки чтения и записи часов

#### ВНИМАНИЕ

Функциональные блоки чтения и записи RTC, ранее доступные в 2.00 MasterTool IEC XE или более ранней версии, устаревают с 2.00 или более поздней версии, следующие блоки больше не используются: *NextoGetDateAndTime*, *NextoGetDateAndTimeMs*, *NextoGetTimeZone*, *NextoSetDateAndTime*, *NextoSetDateAndTimeMs* and *NextoSetTimeZone*.

### 5.10.1. Функциональные блоки для чтения и записи RTC

Среди других функциональных блоков есть очень важные, используемые для чтения часов (*GetDateAndTime*, *GetDayOfWeek* и *GetTimeZone*) и для настройки новых данных даты и времени (*SetDateAndTime* и *SetTimeZone*). Эти функции всегда используют местное время, то есть учитывают значение, определенное часовым поясом.

Ход разбирательства по поводу этих двух блоков описан ниже.

#### 5.10.1.1. Функциональные блоки для чтения RTC

Считывание часов может быть выполнено с помощью следующих функций:

##### 5.10.1.1.1. GetDateAndTime

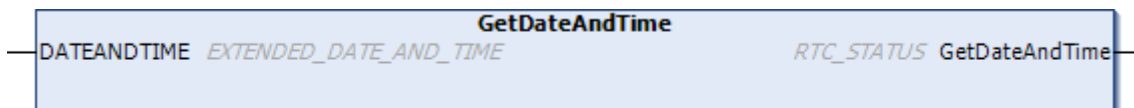


Рисунок 127: Считывание даты и времени

Внутренние параметры	Тип	Описание
<b>DATEANDTIME</b>	EXTENDED_DATE_AND_TIME	Эта переменная возвращает значение даты и часа RTC в формате, показанном в Таблице 145.

Таблица 136: Входные параметры GetDateAndTime

Внешние параметры	Тип	Описание
<b>GETDATEANDTIME</b>	RTC_STATUS	Возвращает состояние ошибки функции, см. Таблицу 147.

Таблица 137: Выходные параметры GetDateAndTime

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR

Result : RTC_STATUS;

DATEANDTIME : EXTENDED_DATE_AND_TIME;

xEnable--:--BOOL;-----
END_VAR

IF xEnable = TRUE THEN

```

#### 5.10.1.1.2. GetTimeZone

Следующая функция считывает Конфигурацию часового пояса, эта функция напрямую связана со временем в часовом поясе в службе синхронизации SNTP:

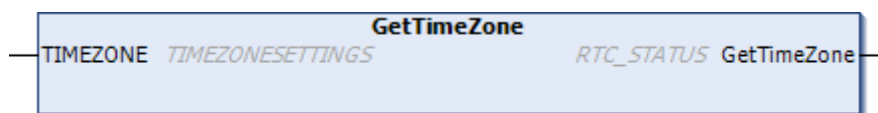


Рисунок 128: Конфигурация чтения временной зоны

Внутренние параметры	Тип	Описание
<b>TIMEZONE</b>	TIMEZONESETTINGS	Эта переменная представляет чтение конфигурации часового пояса.

Таблица 138: Входные параметры GetTimeZone

Внешние параметры	Тип	Описание
<b>GetTimeZone</b>	RTC_STATUS	Возвращает состояние ошибки функции, см. Таблицу 147.

Таблица 139: Выходные параметры GetTimeZone

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR

GetTimeZone_Status : RTC_STATUS;
TimeZone          : TIMEZONESETTINGS;

xEnable : BOOL;
END_VAR

IF xEnable = TRUE THEN

```

#### 5.10.1.1.3. GetDayOfWeek

Функция *GetDayOfWeek* используется для чтения дня недели.

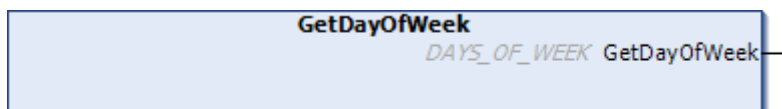


Рисунок 129: Чтение дня недели

Внешние параметры	Тип	Описание
<b>GetDayOfWeek</b>	DAYS_OF_WEEK	Возвращает день недели. См. раздел 146.

Таблица 140: Внешние параметры GetDayOfWeek

При вызове функция считывает день недели и заполняет структуру DAYS\_OF\_WEEK.

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR

DayOfWeek : DAYS_OF_WEEK;
END_VAR

```

## 5.10.1.2. Функции записи RTC

Настройки часов выполняются через функции и функциональные блоки следующим образом:

## 5.10.1.2.1. SetDateAndTime

Функция SetDateAndTime используется для записи настроек часов. Обычно точность составляет порядка сотен миллисекунд.

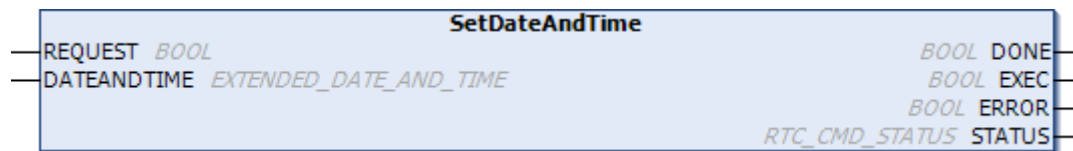


Рисунок 130: Установить дату и время

Внутренние параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, когда получает нарастающий фронт, разрешает запись часов.
<b>DATEANDTIME</b>	EXTENDED_DATE_AND_TIME	Получает значения даты и часа в миллисекундах. См. раздел 145.

Таблица 141: Входные параметры SetDateAndTime

Внешние параметры	Тип	Описание
<b>DONE</b>	BOOL	Если эта переменная имеет значение true, это означает, что действие было успешно завершено.
<b>EXEC</b>	BOOL	Если эта переменная имеет значение true, это означает, что функция обрабатывает значения.
<b>ERROR</b>	BOOL	Если эта переменная имеет значение true, это указывает на ошибку во время записи.
<b>STATUS</b>	RTC_CMD_STATUS	Возвращает ошибку, возникшую во время Конфигурации. См. Таблицу 147.

Таблица 142: Выходные параметры SetDateAndTime

Когда на входе REQUEST появляется нарастающий фронт, функциональный блок записывает новые значения DATEANDTIME в часы. Если запись выполнена успешно, вывод DONE будет равен TRUE. В противном случае выход ERROR будет равен TRUE, и ошибка появится в переменной STATUS.

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR

SetDateAndTime : SetDateAndTime;
xRequest : BOOL;

DateAndTime : EXTENDED DATE AND TIME;

```

```

xExec : BOOL;
xError : BOOL;

xStatus : RTC_STATUS;
END_VAR
-----

IF xRequest THEN
  SetDateAndTime.REQUEST:=TRUE;
  SetDateAndTime.DATEANDTIME:=DateAndTime;
  xRequest:= FALSE;

END_IF

SetDateAndTime();
SetDateAndTime.REQUEST:=FALSE;

IF SetDateAndTime.DONE THEN
  xExec:=SetDateAndTime.EXEC;

```

**ВНИМАНИЕ**

Если вы попытаетесь записать значения времени за пределами диапазона RTC, значения будут преобразованы в допустимые значения при условии, что они не превышают допустимый диапазон от 01.01.2000 до 31.12.2035. Например, если пользователь попытается записать значение 2000 мс, оно будет преобразовано в 2 секунды, напишите значение 100 секунд, оно будет преобразовано в 1 минуту и 40 секунд. Если значение типа 30 часов, оно преобразуется в 1 день и 6 часов и т. д.

## 5.10.1.2.2. SetTimeZone

Следующий функциональный блок выполняет запись настроек часового пояса:

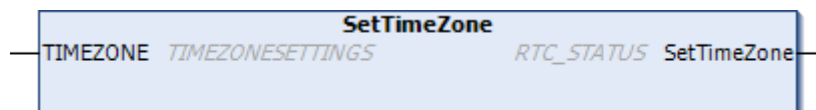


Рисунок 131: Запись настроек часового пояса

Внутренние параметры	Тип	Описание
TIMEZONE	TIMEZONESETTINGS	Структура с часовым поясом для настройки. См. Таблицу 148.

Таблица 143: Внутренние параметры SetTimeZone

Внешние параметры	Тип	Описание
SetTimeZone	RTC_STATUS	Возвращает ошибку, возникшую во время считывания/установки. См. Таблица 147.

Таблица 144: Внешние параметры SetTimeZone

При вызове функция настроит TIMEZONE с новой конфигурацией системного часового пояса. Результат настройки возвращает функция.

## 5. КОНФИГУРАЦИЯ

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR

Status : RTC_STATUS;
TimeZone : TIMEZONESETTINGS;

xWrite : BOOL;
END_VAR

//FB SetTimeZone

IF (xWrite = TRUE) THEN

Status := SetTimeZone(TimeZone);
```

### ВНИМАНИЕ

Для выполнения часов должны использоваться значения времени и даты в следующем допустимом диапазоне: с 00:00:00 часов 01.01.2000 до 31.12.2035 23:59:59 часов, в противном случае сообщается об ошибке через Выходной параметр СТАТУС. Подробнее о выходном параметре STATUS см. в разделе [RTC\\_STATUS](#).

### 5.10.2. Структура данных RTC

Функциональные блоки чтения и настройки ЦП серии Nexto RTC используют следующие структуры данных в своей конфигурации:

#### 5.10.2.1. EXTENDED\_DATE\_AND\_TIME

Эта структура используется для хранения даты RTC при использовании функциональных блоков для чтения/установки даты с точностью до миллисекунд. Это описано в Таблице ниже:

Структура	Тип	Переменная	Описание
EXTENDED_DATE_AND_TIME	BYTE	byDayOfMonth	Сохраняет день установленной даты.
	BYTE	ByMonth	Сохраняет месяц установленной даты.
	WORD	wYear	Сохраняет год установленной даты.
	BYTE	byHours	Сохраняет час установленной даты.
	BYTE	byMinutes	Сохраняет минуты установленной даты.
	BYTE	bySeconds	Сохраняет секунды установленной даты.
	WORD	wMilliseconds	Сохраняет миллисекунды установленной даты.

Таблица 145: EXTENDED\_DATE\_AND\_TIME

#### 5.10.2.2. DAYS\_OF\_WEEK

Эта структура используется для хранения дня недели:

Исчисляемый	Значение	Описание
DAYS_OF_WEEK	0	INVALID_DAY
	1	SUNDAY
	2	MONDAY
	3	TUESDAY
	4	WEDNESDAY
	5	THURSDAY
	6	FRIDAY
	7	SATURDAY

Таблица 146: Структура DAYS\_OF\_WEEK

## 5.10.2.3. RTC\_STATUS

Этот перечислитель используется для возврата типа ошибки в настройке или чтении RTC, и он описан в Таблице ниже:

Исчисляемый	Значение	Описание
RTC_STATUS	NO_ERROR (0)	Нет ошибки.
	UNKNOWN_COMMAND (1)	Неизвестная команда.
	DEVICE_BUSY (2)	Устройство занято.
	DEVICE_ERROR (3)	Устройство с ошибкой.
	ERROR_READING_OSF (4)	Ошибка чтения действительного флага даты и часа.
	ERROR_READING_RTC (5)	Ошибка в чтении даты и часа.
	ERROR_WRITING_RTC (6)	Ошибка в написании даты и часа.
	ERROR_UPDATING_SYSTEM_TIME (7)	Ошибка в обновлении системной даты и часа.

Исчисляемый	Значение	Описание
	INTERNAL_ERROR (8)	Внутренняя ошибка.
	INVALID_TIME (9)	Неверная дата и час.
	INPUT_OUT_OF_RANGE (10)	Вышли за пределы допустимой даты и часа для системы.
	SNTP_NOT_ENABLE (11)	Ошибка генерируется, когда служба SNTP не включена и делается попытка изменить часовой пояс.

Таблица 147: RTC\_STATUS

## 5.10.2.4. TIMEZONESETTINGS

Эта структура используется для хранения значения часового пояса в запросах на чтение/настройку функциональных блоков RTC и описана в Таблице ниже:

Структура	Тип	Переменная	Описание
TIMEZONESETTINGS	INT	lhour	Устанавливает час часового пояса.
	INT	iminutes	Устанавливает минуты часового пояса.

Таблица 148: TIMEZONESETTINGS

**Примечание:**

**Функциональные блоки записи и чтения даты и часа:** различные библиотеки NextoStandard, которые имеют функциональные блоки или функции, которые могут выполнять доступ чтения и записи даты и часа в системе, не указаны. Библиотека NextoStandard имеет соответствующие интерфейсы для записи и чтения системной даты и часа соответственно и для информирования правильной диагностики.

## 5.11. Память пользовательских файлов

ЦП серии Nexto имеют область памяти, предназначенную для общего хранения данных, другими словами, пользователь может хранить в памяти ЦП несколько файлов проекта любого формата. Эта область памяти зависит от используемой модели процессора (см. раздел «Память»).

Чтобы использовать эту область, пользователь должен получить доступ к проекту в программном обеспечении MasterTool IEC XE и щелкнуть дерево устройств, расположенное слева от программы. Дважды щелкните элемент «Устройство» и, после выбора ЦП на вкладке «Настройки связи», которая будет открыта, выберите вкладку «Файлы» и нажмите «Обновить» как в столбце файлов компьютера (слева), так и в столбце файлов ЦП (справа). показано на рисунке ниже.

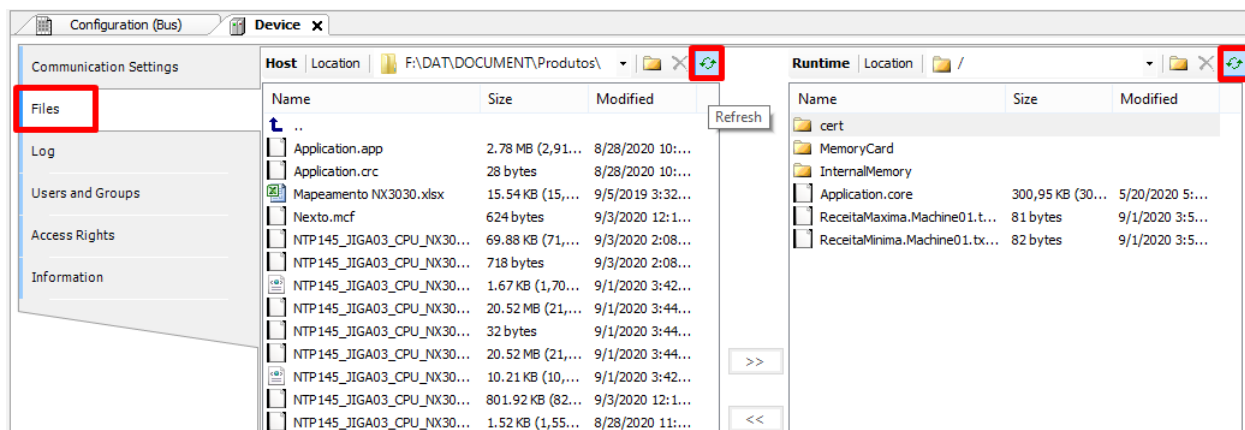


Рисунок 132: Доступ к пользовательским файлам

После обновления столбца файлов ЦП будет показан корневой каталог файлов, хранящихся в ЦП. Затем можно будет выбрать папку, в которую будут перенесены файлы. Папка «InternalMemory» — это папка «По умолчанию», которая используется для хранения файлов во внутренней памяти ЦП, поскольку невозможно передать файлы в корневой каталог. При необходимости пользователь может создавать другие папки в корневом каталоге или подпапки внутри папки «InternalMemory».

Папка «MemoryCard» — это каталог, в который монтируется карта памяти, если она вставлена в ЦП. Файлы, передаваемые на «MemoryCard», передаются непосредственно на карту памяти. По мере добавления новых функций в продукт могут появиться некоторые папки, которые пользователь должен игнорировать.

**ВНИМАНИЕ**

В случае, когда карта памяти вставлена после запуска ЦП, будут запрошены имя пользователя и пароль для выполнения доступа MasterTool IEC XE и/или передачи файлов на карту памяти или наоборот. Стандартный пользователь с правами доступа к ЦП — «Владелец», а пароль по умолчанию для этого пользователя — «Владелец».

Для того, чтобы выполнить передачу файла с микрокомпьютера на ЦП, достаточно выбрать нужный файл в левой колонке и нажать клавишу «>>», расположенную в центре экрана, как показано на рисунке ниже. Время загрузки зависит от размера файла и времени цикла (выполнения) текущего приложения ЦП и может занять несколько минут.

Пользователю не нужно находиться в режиме выполнения или подключаться к ЦП для выполнения передачи, поскольку он имеет возможность автоматически подключаться, когда пользователь выполняет передачу.

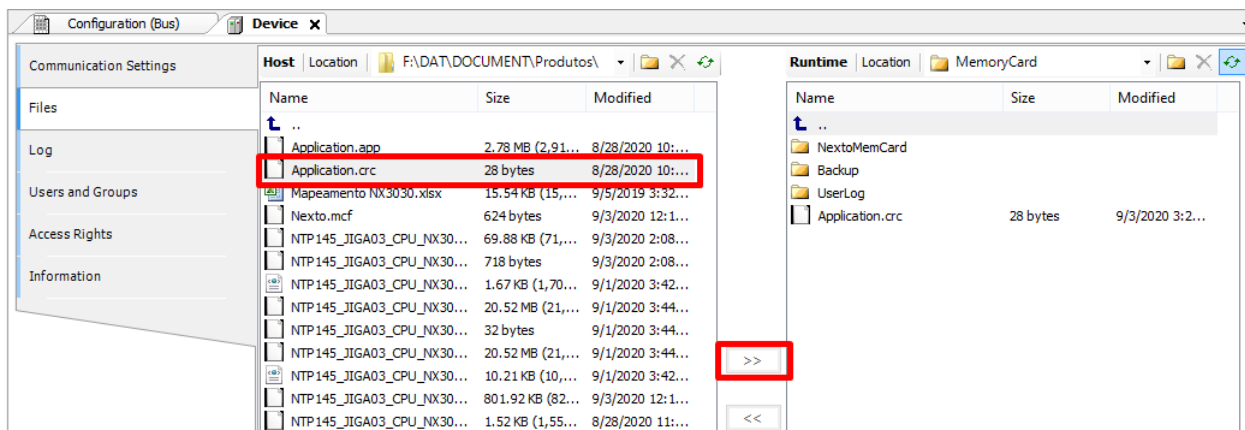





Рисунок 133: Передача файлов

**ВНИМАНИЕ**

Файлы, содержащиеся в папке проекта, созданного MasterTool IEC XE, имеют специальные имена, зарезервированные системой, поэтому их нельзя передать через вкладку «Файлы». Если пользователь хочет перенести проект в память пользователя, необходимо сжать папку, а затем загрузить сжатый файл (например, \*.zip).

В случае, если необходимо передать документы с ЦП на ПК, на котором установлено программное обеспечение MasterTool IEC XE, пользователь должен выполнить процедуру, очень похожую на описанную ранее, так как файл должен быть выбран из правого столбца и кнопка Нажата «<<<», расположенная в центре экрана.

Кроме того, у пользователя есть несколько вариантов операций в области хранения файлов, а именно:

- Новый каталог : позволяет создать новую папку в пользовательской области памяти.
- Удаление пункта : позволяет исключать файлы из папок в пользовательской области памяти.
- Обновление : позволяет обновлять на экране MasterTool IEC XE файлы в области пользовательской памяти и в вычислительной.

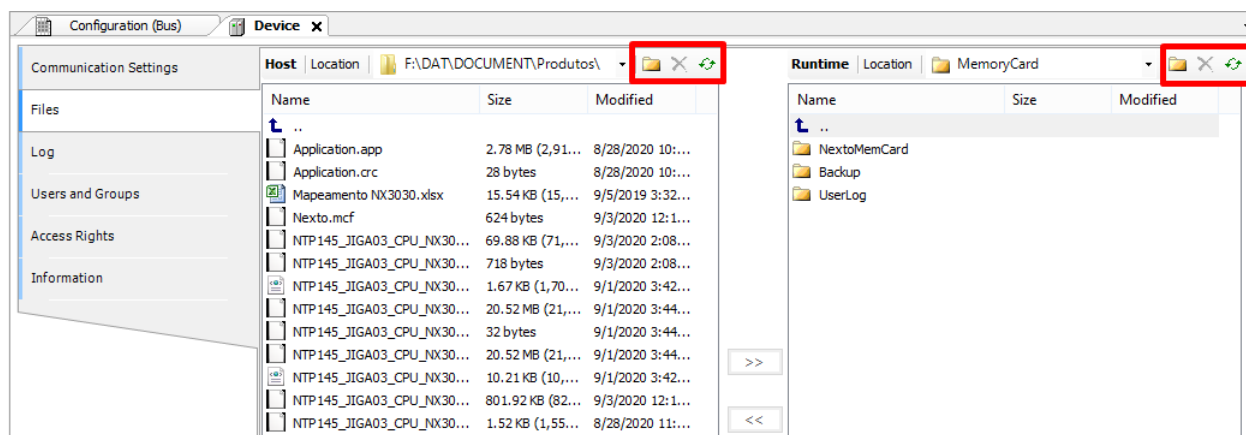


Рисунок 134: Варианты использования

**ВНИМАНИЕ**

Для процессора в режиме остановки или без приложения скорость передачи во внутреннюю память составляет примерно 150 Кбайт/с.

## 5.12. Карта памяти

Помимо прочего, процессоры серии Nexto позволяют пользователю использовать карту памяти. Он определяется в соответствии с функциями, описанными в разделе «Интерфейс карты памяти», который хранит, среди прочих файлов, проект и приложение во внутренней памяти ЦП.

Когда карта вставляется в ЦП и представляет тип файла, отличный от FAT32, она автоматически идентифицирует эти файлы и спрашивает пользователя, хочет ли он отформатировать файлы. В отрицательном случае пользователь не может использовать карту, так как она не смонтирована. Появляется сообщение о том, что формат не распознан, и наличие карты также не отображается. Если пользователь решает отформатировать файлы, ЦП требуется несколько минут для выполнения операции, в зависимости от времени цикла (выполнения) приложения, которое выполняется в ЦП. Как только карта памяти установлена, ЦП считывает ее общую информацию, что замедляет доступ к карте памяти в первые несколько минут. Эта процедура выполняется только при вставленной карте или в случае сброса ЦП.

**ВНИМАНИЕ**

Карту памяти рекомендуется форматировать непосредственно в процессоре Nexto во избежание возможных проблем при использовании, увеличения времени монтажа или даже некорректной работы.

Не рекомендуется извлекать карту памяти или обесточивать процессор во время форматирования или передачи файлов, так как это может привести к потере данных, а также к необратимым повреждениям.

### 5.12.1. Подготовка проекта

Для использования функциональности во время Конфигурации проекта в ПО MasterTool IEC XE пользователь должен включить опцию копирования проекта ЦП на карту памяти и/или карты памяти в ЦП и настроить пароли. Эти пароли будут запрошены CPU при выполнении соответствующей передачи. Информацию о Таблице см. в разделе Параметры проекта.

#### ВНИМАНИЕ

Если ЦП не имеет приложения, будет доступно меню «Карта памяти», позволяющее перенести проект с карты памяти на ЦП без какой-либо предварительной подготовки ЦП.

Для использования функции необходимо выполнить следующие шаги.

Перейдите в меню «Онлайн» и выполните команду «Создать загрузочное приложение», помня, что вы не можете войти в ЦП для выполнения этой процедуры. После выполнения этой команды в папке, в которой сохранен проект, создаются два файла. Один с расширением «приложение» и один с расширением «src».

После создания файлов в предыдущем пункте необходимо перейти к настройкам общих параметров ЦП и нажать кнопку Карта памяти... Откроется новый экран, как показано на рисунке ниже. На этом экране вы должны активировать желаемую(ые) операцию(ии) передачи и, при необходимости, установить пароль(и) только с числовыми символами. Использование пароля не требуется.

Чтобы завершить операцию установки, вы должны нажать кнопку «Найти файл...», а затем найти файл с расширением «.src», созданный на предыдущем шаге.

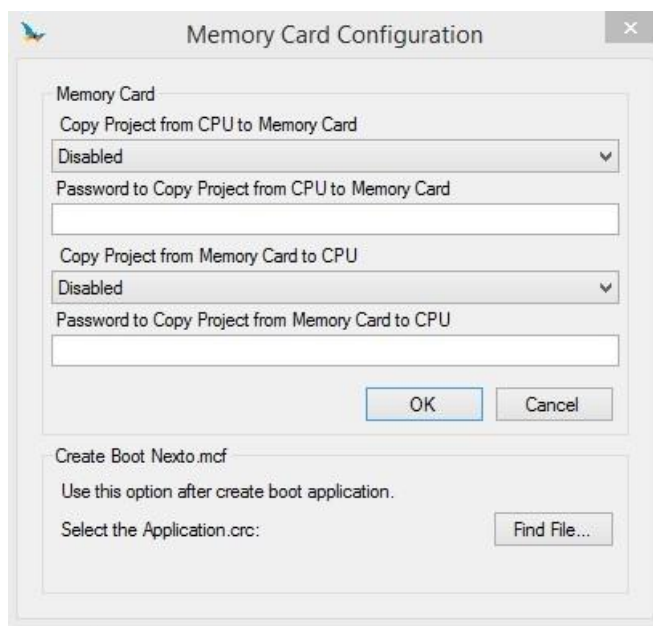


Рисунок 135: Настройки карты памяти

После выполнения этих действий MasterTool IEC XE отправит все файлы, необходимые для выполнения операций отправки и получения проектов через карту памяти. Если карта памяти смонтирована, то пароль будет записан на нее. В противном случае пароль, установленный в MasterTool, будет запрошен, если пользователь попытается перенести проект ЦП на карту памяти.

### 5.12.2. Перенос проекта

Чтобы перенести проект с ЦП на карту памяти или наоборот, пользователь, помимо включения в ПО MasterTool IEC XE возможности использования функциональности, должен будет войти в меню Карта памяти в ЦП, используя диагностический ключ, и выбрать желаемый вариант передачи.

#### ВНИМАНИЕ

Перенос проекта на карту памяти должен производиться только с помощью диагностического ключа ЦП.

После этого вам будет предложено ввести пароль, если пользователь установил его во время установки приложения. Затем при коротком нажатии на клавишу диагностики цифры увеличиваются, а при длительном нажатии подтверждаются. В подтвержденной шестой цифре ЦП будет состоять из пароля и запускать процесс.

После переноса карты памяти в ЦП, если имеется приложение RUN, оно будет оставаться в состоянии STOP по соображениям безопасности. Чтобы перевести ЦП в режим RUN, его необходимо перезагрузить.

Когда пароли как приложения, которое находится в ЦП, так и приложения, которое находится на карте памяти, одинаковы, не требуется вводить пароли в меню ЦП для выполнения переноса приложений. Дополнительные сведения об использовании клавиши диагностики см. в разделе Диагностика одним нажатием.

Чтобы извлечь карту памяти, просто нажмите и удерживайте клавишу MS и подождите, пока значок карты памяти не исчезнет с экрана состояния графического дисплея.

**ВНИМАНИЕ**

Если карта памяти извлечена без размонтирования через меню ЦП, во время передачи файла этот процесс может привести к потере данных карты, а также повреждению файлов на ней. Этот процесс может вызвать необходимость форматирования другой карты, когда она будет снова вставлена в ЦП.

**ВНИМАНИЕ**

Если в корне карты памяти есть какой-либо файл с именем «NextoMemCard» или «Backup», он будет удален для создания системных папок с таким же именем, используемых ЦП для хранения приложения проекта и архива проекта. Папки с такими именами не будут перезаписаны.

**5.12.3. Доступ MasterTool**

Доступ к карте памяти связан с тем же экраном пользовательской памяти в программном обеспечении MasterTool ИЕС ХЕ, поскольку он смонтирован в папке под названием MemoryCard. Папки NextoMemCard и Backup создаются на карте памяти каждый раз, когда последняя вставляется в ЦП. Если эти папки уже существуют, система распознает их и не будет перезаписывать папки.

В NextoMemCard на карте памяти вы найдете файлы приложения, в этом окне у вас еще есть возможность сохранить ваш проект в предпочитаемом каталоге (если вы отправили исходный код). В MasterTool в опции "Файл/Архив проекта"

/ Извлечь архив..." можно открыть в MasterTool сохраненное приложение, которое находится в выбранной ранее директории.

Папка Backup не используется пользователем.

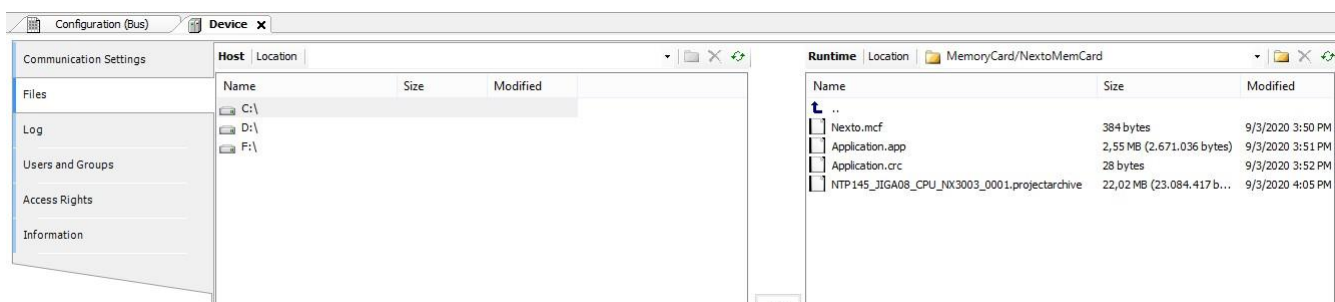


Рисунок 136: Каталог с картой памяти, вставленной вместе с проектом

**ВНИМАНИЕ**

Время передачи файлов зависит от разницы во времени интервала за вычетом среднего времени выполнения задачи(ов) в исполнении (доступное время до следующего цикла задачи), то есть чем больше эта разница для каждой задачи в приложении, тем быстрее будет быть переносом данных с карты памяти на CPU/MasterTool ИЕС ХЕ или наоборот.

Передача файлов на карту памяти будет медленнее, чем передача во внутреннюю память ЦП. Для ЦП в режиме остановки или без приложения скорость передачи близка к 100 Кбайт/с.

**5.13. Информативность процессора и Конфигурация меню**

Доступ к Информативному меню, Конфигурации ЦП Nexto и подробной диагностике доступен через уровни, а для доступа к информации меню, изменения уровня и изменения любой Конфигурации требуется долгое нажатие на кнопку диагностики и для навигации по элементам на того же уровня требуется короткое нажатие на кнопку диагностики. См. раздел One Touch Diag, чтобы проверить работу и различия между типами касания кнопки диагностики.

Таблица ниже показывает уровни меню и каждый тип экрана, доступный в ЦП, если они информативны, настраиваются или возвращают уровень.

Уровень 1	Уровень 2	Уровень 3	Тип
HARDWARE	TEMPERATURE	-	Информативный
	CONTRAST	CONTRAST LEVEL	Настраиваемый
	DATE AND TIME	-	Информативный
	BACK	-	Уровень возврата
LANGUAGES	ENGLISH	>ENGLISH	Настраиваемый
	PORTUGUES	>PORTUGUES	Настраиваемый
	BACK	-	Уровень возврата
NETWORK	NET 1 IP ADDR.		Информативный
	NET 1 MASK		Информативный
	BACK		Уровень возврата
SOFTWARE	FIRMWARE		Информативный
	BOOTLOADER		Информативный
	AUX. PROC.		Информативный
	BACK		Уровень возврата
BACK	-	-	Уровень возврата

Таблица 149: Уровни меню процессора

Как показано в Таблице 149, между доступными Вариантами для визуализации и модификации находятся основные данные, необходимые пользователю, такие как: Информация об аппаратных ресурсах:

- **ТЕМПЕРАТУРА** – внутренняя температура ЦП (пример: 36 C 97 F)
- **CONTRAST** – Настройка контрастности фронтального дисплея ЦП.
- **ДАТА И ВРЕМЯ** – Дата и время, установленные в ЦП (пример: 2001.01.31 00:00) Изменение языка меню в ЦП:
- **PORTUGUES** – меняет язык на португальский.
- **ENGLISH** – меняет язык на английский. Визуализация информации о сети, установленной в устройстве:
- **СЕТЕВОЙ 1 IP-АДРЕС.** – Адрес (пример: 192.168.0.1)
- **NET 1 MASK** – маска подсети (пример: 255.255.255.0)

Информация о версиях ПО:

- **ПРОШИВКА** – версия программного обеспечения ЦП (пример: 1.0.0.0)
- **BOOTLOADER** – версия загрузчика процессора (пример: 1.0.0.0)
- **ДОП. ПРОК.** – Версия вспомогательного процессора ЦП (пример: 1.0.0.0)

На приведенном ниже рисунке показан пример работы с меню ЦП Nexto с помощью процедуры настройки меню контрастности на экране состояния. Помимо упрощения Конфигурации, можно идентифицировать все уровни экрана и тип касания для навигации по ним, а также изменять другие параметры, такие как Язык и Карта памяти, используя ту же логику доступа. Короткое касание показывает, что контрастность увеличивается (более четкая), а при следующем касании после максимального значения она возвращается к минимальному значению (менее четкому). Длительное касание показывает подтверждение желаемого контраста и его возврат на предыдущий уровень.

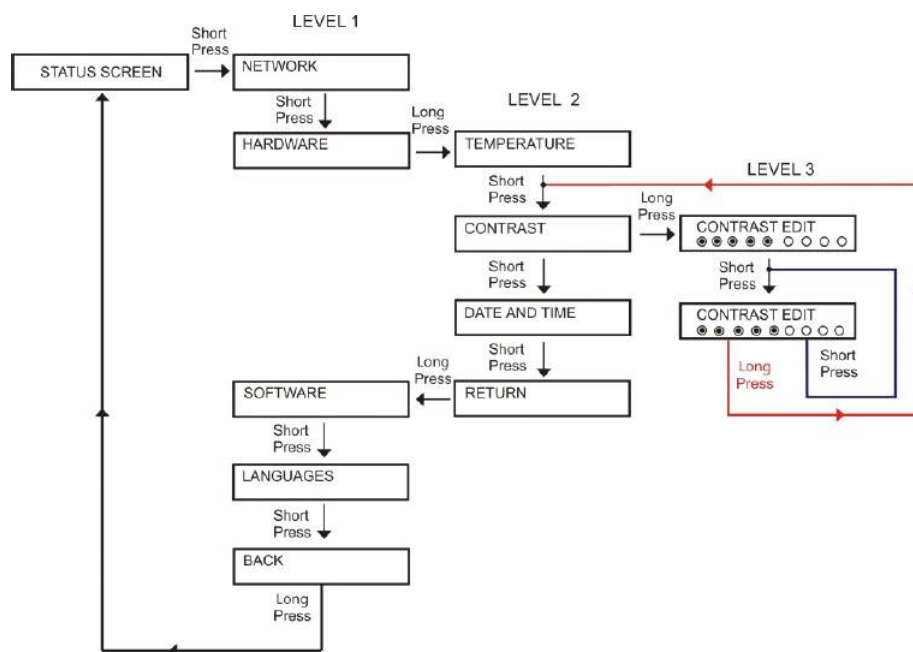


Рисунок 137: Настройка контрастности

Кроме возможности закрытия меню процессоров Nexto долгим нажатием на экранной диагностической кнопки НАЗАД

начиная с уровня 1, существуют и другие условия вывода, описанные ниже:

Короткое прикосновение в любой момент к другим модулям, существующим на шине, приводит к отключению ЦП от меню и отображению диагностики желаемого модуля.

Время простоя на любом уровне превосходит 5 с.

## 5.14. Функциональные блоки и функции

### 5.14.1. Специальные функциональные блоки для последовательных интерфейсов

Специальные функциональные блоки для последовательных интерфейсов обеспечивают локальный доступ (COM 1 и COM 2), а также доступ к удаленным последовательным портам (модули расширения). Таким образом, пользователь может создавать свои собственные протоколы и управлять последовательными портами по своему усмотрению, следуя языкам IEC 61131-3, доступным в программном обеспечении MasterTool IEC XE. Блоки доступны внутри библиотеки NextoSerial, которую необходимо добавить в проект, чтобы можно было их использовать (о выполнении процедуры вставки библиотеки см. Руководство по программированию MasterTool IEC XE – MP399609, раздел Библиотека).

Специальные функциональные блоки для последовательных интерфейсов могут занять несколько циклов (последовательных вызовов) для завершения выполнения задачи. Иногда блок можно выполнить за один цикл, но в общем случае требуется несколько циклов. Выполнение задачи, связанной с блоком, может состоять из множества шагов, некоторые из которых зависят от внешних событий, которые могут быть значительно задержаны. Функциональный блок не может реализовать подпрограммы, занимающие время ожидания этих событий, потому что это загрузит ЦП. Решением может быть создание блокирующих функциональных блоков, но это нецелесообразно, поскольку это увеличивает сложность пользовательского приложения, поскольку обычно многозадачное программирование недоступно. Следовательно, когда ожидается внешнее событие, последовательные функциональные блоки завершают работу, и управление возвращается основной программе. Обработка задачи продолжается в следующем цикле, другими словами, при следующем вызове блока.

Перед описанием специальных функциональных блоков для последовательных интерфейсов важно знать типы данных, то есть тип данных, используемый блоками.

Тип данных	Варианты	Описание
SERIAL_BAUDRATE	BAUD2400	Перечисляет все возможные скорости передачи данных (бит в секунду)
	BAUD4800	
	BAUD9600	
	BAUD19200	
	BAUD38400	
	BAUD57600	
	BAUD115200	
SERIAL_DATABITS	DATABITS_7	Перечисляет все возможные биты данных.
	DATABITS_8	
SERIAL_MODE	NORMAL_MODE	Последовательная связь Нормальный режим работы.
	EXTENDED_MODE	Расширенная последовательная связь Режим работы, в котором предоставляется информация о принятом кадре данных.
SERIAL_PARAMETERS	Определяет все возможности сигнала модема для Конфигурации:	
	BAUDRATE	Определено в SERIAL_BAUDRATE.
	DATABITS	Определено в SERIAL_DATABITS.
	STOPBITS	Определено в SERIAL_STOPBITS.
	PARITY	Определено в SERIAL_PARITY.
	UART_RX_THRESHOLD	Количество байтов, которое должно быть получено, чтобы сгенерировать новое прерывание UART. Меньшие значения делают TIMESTAMP более точным при использовании РАСШИРЕННОГО РЕЖИМА и минимизируют ошибки переполнения. Однако слишком низкие значения могут вызвать слишком много прерываний и задержку процессора.
	MODE	Определено в SERIAL_MODE.
SERIAL_PARITY	ENABLE_RX_ON_TX	При значении true весь полученный байт во время передачи будет выгружен, а не отправлен на линию RX. Используется для отключения полнодуплексного режима в интерфейсе RS-422.
	PARITY_NONE	Перечислите все возможности четности.
	PARITY_ODD	
	PARITY_EVEN	
SERIAL_PORT	COM 1	Перечислите все доступные последовательные порты (COM 10, COM 11, COM 12, COM 13, COM 14, COM 15, COM 16, COM 17, COM 18 и COM 19 – модули расширения).
SERIAL_RX_CHAR_EXTENDED	Определяет символ в очереди RX в расширенном режиме.	
	RX_CHAR	Байт данных.
	RX_ERROR	Код ошибки.

Тип данных	Варианты	Описание
	RX_TIMESTAMP	Тишина из-за предыдущего символа или из-за другого события, которое произошло до этого символа (Конфигурация последовательного порта, окончание передачи).
SERIAL_RX_QUEUE_STATUS	В нем есть несколько полей, которые предоставляют информацию о статусе/ошибке очереди RX, используемую при использовании обычного формата (без информации об ошибках и метках времени):	
	RX_FRAMING_ERRORS	Счетчик ошибок кадра: неправильное формирование символа – нет стопового бита, неправильная скорость передачи данных, в том числе – начиная с последовательного порта Конфигурация. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_PARITY_ERRORS	Счетчик ошибок четности, начиная с последовательного порта Конфигурация. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_BREAK_ERRORS	Счетчик ошибок прерывания, так как последовательный порт Конфигурация, другими словами, активная строка выше символа времени. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_FIFO_OVERRUN_ERRORS	Счетчик ошибок переполнения FIFO RX, поскольку последовательный порт Конфигурация, другими словами, ошибка в настроенном пороге FIFO RX. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_QUEUE_OVERRUN_ERRORS	Счетчик ошибок переполнения очереди RX, другими словами, превышено максимальное количество символов (1024) и происходит перезапись данных. Возвращается к нулю, когда достигает максимального значения (65535).
	RX_ANY_ERRORS	Суммируйте последние 5 счетчиков ошибок (кадр, четность, прерывание, переполнение RX FIFO, переполнение очереди RX).
	RX_REMAINING	Количество символов в очереди RX.
Список кодов критических ошибок, которые могут быть возвращены последовательным функциональным блоком. Каждый блок возвращает определенные ошибки, которые будут описаны ниже:		
	NO_ERROR	Нет ошибок

Тип данных	Варианты	Описание
------------	----------	----------

SERIAL_STATUS	ILLEGAL_*	Вернуть параметры с недопустимыми значениями или вне допустимого диапазона: - SERIAL_PORT - SERIAL_MODE - BAUDRATE - DATA_BITS - PARITY - STOP_BITS - HANDSHAKE - UART_RX_THRESHOLD - TIMEOUT - TX_BUFF_LENGTH - HANDSHAKE_METHOD - RX_BUFF_LENGTH
	PORT_BUSY	Указывает, что последовательный порт используется другим экземпляром
	HW_ERROR_UART	Обнаружена аппаратная ошибка в UART.
	HW_ERROR_REMOTE	Аппаратная ошибка при обмене данными с удаленным последовательным портом.
	CTS_TIMEOUT_ON	Тайм-аут ожидания включения CTS в рукопожатии RS-232 RTS/CTS в блоке SERIAL_TX.
	CTS_TIMEOUT_OFF	Тайм-аут ожидания отключения CTS, в рукопожатии RS-232 RTS/CTS, в блоке SERIAL_TX.
	TX_TIMEOUT_ERROR	Тайм-аут ожидания окончания передачи в SERIAL_TX.
	RX_TIMEOUT_ERROR	Тайм-аут при ожидании всех символов в блоке SERIAL_RX или блоке SERIAL_RX_EXTENDED.
	FB_SET_CTRL_NOT_ALLOWED	Блок SET_CTRL нельзя использовать, если квитирование отличается от RS232_MANUAL.
	FB_GET_CTRL_NOT_ALLOWED	Блок GET_CTRL нельзя использовать, если квитирование отличается от RS232_MANUAL.
	FB_SERIAL_RX_NOT_ALLOWED	SERIAL_RX недоступен для очереди RX, расширенный режим.
	FB_SERIAL_RX_EXTENDED_NOT_ALLOWED	SERIAL_RX_EXTENDED недоступен для очереди RX в обычном режиме.
	DCD_INTERRUPT_NOT_ALLOWED	Прерывание сигналом DCD не может быть включено, если последовательный порт не имеет соответствующего контакта.

## 5. КОНФИГУРАЦИЯ

Тип данных	Варианты	Описание
	CTS_INTERRUPT_NOT_ALLOWED	Прерывание по сигналу CTS не может быть включено, если квитирование отличается от RS232_MANUAL или если последовательный порт не имеет соответствующего контакта.
	DSR_INTERRUPT_NOT_ALLOWED	Прерывание сигналом DSR не может быть включено, если последовательный порт не имеет соответствующего контакта. (ЦП Nexto не имеют этого сигнала в локальных портах)
	NOT_CONFIGURED	Функциональный блок нельзя использовать перед последовательным портом Конфигурация.
	INTERNAL_ERROR	Указывает, что в последовательном порту возникла внутренняя проблема.
SERIAL_STOPBITS	STOPBITS_1	Перечислите все возможности стоп-битов.
	STOPBITS_2	
	STOPBITS_1_5	

Таблица 150: Последовательные функциональные блоки Типы данных

### 5.14.1.1. SERIAL\_CFG

Этот функциональный блок используется для настройки и инициализации нужного последовательного порта. После вызова блока каждая очередь RX и TX, связанная с последовательными портами, и FIFO RX и TX перезапускаются.



Рисунок 138: Конфигурация серийного блока

Входные параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
PARAMETERS	SERIAL_PARAMETERS	Эта структура определяет параметры конфигурации последовательного порта, как описано в типе данных SERIAL_PARAMETERS.

Таблица 151: Входные параметры SERIAL\_CFG

Выходные параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.

## 5. КОНФИГУРАЦИЯ

<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
<b>STATUS</b>	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: - NO_ERROR - ILLEGAL_SERIAL_PORT - ILLEGAL_SERIAL_MODE - ILLEGAL_BAUDRATE - ILLEGAL_DATA_BITS - ILLEGAL_PARITY - ILLEGAL_STOP_BITS - ILLEGAL_HANDSHAKE - ILLEGAL_UART_RX_THRESHOLD - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - DCD_INTERRUPT_NOT_ALLOWED - CTS_INTERRUPT_NOT_ALLOWED - DSR_INTERRUPT_NOT_ALLOWED

Таблица 152: Выходные параметры SERIAL\_CFG

Пример использования на языке ST, после вставки библиотеки Nexto Serial в проект:

```
PROGRAM UserPrg
VAR

Config: SERIAL_CFG;

Port: SERIAL_PORT := COM1;

Parameters: SERIAL_PARAMETERS := (BAUDRATE := BAUD9600,
DATABITS := DATABITS_8,

STOPBITS := STOPBITS_1,
PARITY := PARITY_NONE,
HANDSHAKE := RS232_RTS,
UART_RX_THRESHOLD := 8,
MODE :=NORMAL_MODE,
ENABLE_RX_ON_TX := FALSE,
ENABLE_DCD_EVENT := FALSE,
ENABLE_CTS_EVENT := FALSE);
Status: SERIAL_STATUS;
END_VAR
```

```
Config.PARAMETERS := Parameters;

//FUNCTION:

Config();

//OUTPUTS:

Status := Config.STATUS;    // Если нужно исправить ошибку.
```

### 5.14.1.2. SERIAL\_GET\_CFG

Функциональный блок используется для захвата желаемой конфигурации последовательного порта.



Рисунок 139: Блок для захвата последовательной конфигурации

Входные параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
<b>PORT</b>	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 153: Входные параметры SERIAL\_GET\_CFG

Выходные параметры	Тип	Описание
<b>DONE</b>	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
<b>EXEC</b>	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее статус отображается после завершения блока.

Выходные параметры	Тип	Описание
<b>STATUS</b>	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
<b>PARAMETERS</b>	SERIAL_PARAMETER S	Эта структура получает параметры конфигурации последовательного порта, как описано в типе данных SERIAL_PARAMETERS.

Таблица 154: SERIAL\_GET\_CFG Выходные параметры

## 5. КОНФИГУРАЦИЯ

---

Пример использования на языке ST, после вставки библиотеки в проект:

```
PROGRAM UserPrg
VAR

GetConfig: SERIAL_GET_CFG;
Port: SERIAL_PORT := COM1;

Parameters: SERIAL_PARAMETERS;
Status: SERIAL_STATUS;

END_VAR

//INPUTS:

GetConfig.REQUEST := TRUE;

GetConfig.PORT := Port;

//FUNCTION:

GetConfig();

//OUTPUTS:
```

### 5.14.1.3. SERIAL\_GET\_RX\_QUEUE\_STATUS

Этот блок используется для чтения некоторой информации о статусе очереди RX, специально разработанной для нормального режима, но его также можно использовать в расширенном режиме.

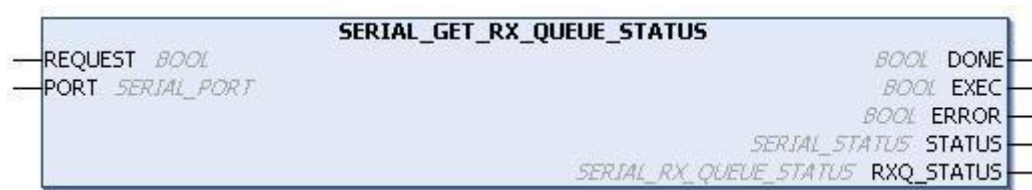


Рисунок 140: Блок, используемый для визуализации статуса очереди RX

Входные параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
<b>PORT</b>	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 155: Входные параметры SERIAL\_GET\_RX\_QUEUE\_STATUS

Выходные параметры	Тип	Описание
<b>DONE</b>	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
<b>EXEC</b>	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.
<b>STATUS</b>	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS, таковы: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - NOT_CONFIGURED
<b>RXQ_STATUS</b>	SERIAL_RX_QUEUE_STATUS	Возвращает статус/ошибку очереди RX, как описано в типе данных SERIAL_RX_QUEUE_STATUS.

Таблица 156: SERIAL\_GET\_RX\_QUEUE\_STATUS Выходные параметры

Пример использования на языке ST, после вставки библиотеки в проект и настройки последовательного порта:

```
PROGRAM UserPrg
VAR

Get_Status: SERIAL_GET_RX_QUEUE_STATUS;
Port: SERIAL_PORT := COM1;

Status: SERIAL_STATUS;

Status_RX: SERIAL_RX_QUEUE_STATUS;
END_VAR

//INPUTS:

Get_Status.REQUEST := TRUE;
Get_Status.PORT := Port;

//FUNCTION:

Get_Status();

//OUTPUTS:

Get_Status.DONE;
```

5.14.1.4. SERIAL\_PURGE\_RX\_QUEUE

Этот функциональный блок используется для очистки очереди RX последовательного порта, локальной и удаленной. UART RX FIFO также перезапускается.

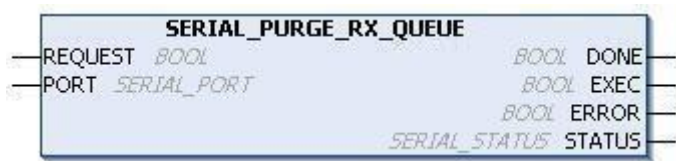


Рисунок 141: Блок, используемый для очистки очереди RX

Входные параметры	Тип	Описание
REQUEST	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
PORT	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.

Таблица 157: Входные параметры SERIAL\_PURGE\_RX\_QUEUE

Выходные параметры	Тип	Описание
DONE	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
EXEC	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
ERROR	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

## 5. КОНФИГУРАЦИЯ

<b>STATUS</b>	<b>SERIAL_STATUS</b>	В случае, если переменная <b>ERROR</b> имеет значение <b>true</b> , структура <b>STATUS</b> покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных <b>SERIAL_STATUS</b> , таковы: - <b>NO_ERROR</b> - <b>ILLEGAL_SERIAL_PORT</b> - <b>PORT_BUSY</b> - <b>HW_ERROR_UART</b> - <b>HW_ERROR_REMOTE</b> - <b>NOT_CONFIGURED</b>
---------------	----------------------	---

Таблица 158: Выходные параметры **SERIAL\_PURGE\_RX\_QUEUE**

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```
PROGRAM UserPrg
VAR

Purge_Queue: SERIAL_PURGE_RX_QUEUE;
Port: SERIAL_PORT := COM1;

Status: SERIAL_STATUS;
END_VAR

//INPUTS:

Purge_Queue.REQUEST := TRUE;
Purge_Queue.PORT := Port;

//FUNCTION:

Purge_Queue();

//OUTPUTS:
```

### 5.14.1.5. SERIAL\_RX

Этот функциональный блок используется для получения буфера последовательного порта в обычном режиме очереди приема. В этом режиме каждый символ в очереди **RX** занимает один байт, который содержит полученные данные, сохраняя 5, 6, 7 или 8 бит, в соответствии с конфигурацией последовательного интерфейса.



Рисунок 142: Блок, используемый для чтения значений буфера приема

Входные параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, если она принимает значение <b>true</b> , разрешает использование функционального блока.
<b>PORT</b>	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных <b>SERIAL_PORT</b> .
<b>RX_BUFFER_POINTER</b>	POINTER TO BYTE	Указатель массива байтов для получения значений буфера.

<b>RX_BUFFER_LENGTH</b>	UINT	Укажите ожидаемое количество символов в массиве байтов. В случае, если доступно больше ожидаемых байтов, из массива байтов будет прочитано только ожидаемое количество, остальное будет оставлено в очереди RX (максимальный размер равен 1024 символам).
<b>RX_TIMEOUT</b>	UINT	Укажите время ожидания для получения ожидаемого количества символов. В случае, если он меньше необходимого для приема символов, будет указан вывод RX_TIMEOUT_ERROR из параметра STATUS. Когда указанное значение в мс равно нулю, функция вернет данные из буфера.

Таблица 159: SERIAL\_RX Входные параметры

Выходные параметры	Тип	Описание
<b>DONE</b>	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
<b>EXEC</b>	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

Выходные параметры	Тип	Описание
<b>STATUS</b>	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_NOT_ALLOWED - NOT_CONFIGURED
<b>RX_RECEIVED</b>	UINT	Возвращает полученное количество символов. Это число может быть в пределах нуля и настроенного значения в RX_BUFFER_LENGTH. Если оно меньше, функциональный блок укажет на ошибку.
<b>RX_REMAINING</b>	UINT	Возвращает количество символов, которые все еще находятся в очереди RX после выполнения функционального блока.

Таблица 160: SERIAL\_RX Выходные параметры

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```
PROGRAM UserPrg
VAR

Receive: SERIAL_RX;

Port: SERIAL_PORT := COM1;           //Max size.

Buffer_Pointer: ARRAY [0..1023] OF BYTE;
Status: SERIAL_STATUS;

END_VAR

Receive.RX_BUFFER_POINTER := ADR(Buffer_Pointer);
Receive.RX_BUFFER_LENGTH  := 1024;   //Max size.
Receive.RX_TIMEOUT := 10000;

//FUNCTION:

Receive();

//OUTPUTS:

Receive.DONE;
Receive.EXEC;
Receive.ERROR;
```

## 5.14.1.6. SERIAL\_RX\_EXTENDED

Этот функциональный блок используется для получения буфера последовательного порта с использованием расширенного режима очереди приема, как показано в разделе Конфигурация последовательных интерфейсов.



Рисунок 143: Блок, используемый для чтения буфера приема

Входные параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
<b>PORT</b>	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
<b>RX_BUFFER_POINTER</b>	POINTER TO SERIAL_RX_CHAR_EXTENDED	Указатель массива SERIAL_RX_CHAR_EXTENDED для получения значений буфера.
<b>RX_BUFFER_LENGTH</b>	UINT	Укажите ожидаемый номер символа в массиве SERIAL_RX_CHAR_EXTENDED. В случае, если доступно больше ожидаемых байтов, из массива байтов будет прочитано только ожидаемое количество, остальное будет оставлено в очереди RX (максимальный размер равен 1024 символам).
<b>RX_TIMEOUT</b>	UINT	Укажите время ожидания для получения ожидаемого количества символов. В случае, если он меньше необходимого для приема символов, будет указан вывод RX_TIMEOUT_ERROR из параметра STATUS. Когда указанное значение в мс равно нулю, функция вернет данные из буфера.

Таблица 161: SERIAL\_RX\_EXTENDED Входные параметры

Выходные параметры	Тип	Описание
<b>DONE</b>	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
<b>EXEC</b>	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

Выходные параметры	Тип	Описание
<b>STATUS</b>	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_RX_BUFF_LENGTH - RX_TIMEOUT_ERROR - FB_SERIAL_RX_EXTENDED_NOT_ALLOWED - NOT_CONFIGURED
<b>RX_RECEIVED</b>	UINT	Возвращает полученное количество символов. Это число может быть в пределах нуля и настроенного значения в RX_BUFFER_LENGTH. Если оно меньше, функциональный блок укажет на ошибку.
<b>RX_REMAINING</b>	UINT	Возвращает количество символов, которые все еще находятся в очереди RX после выполнения функционального блока.
<b>RX_SILENCE</b>	UINT	Возвращает время молчания в очереди RX, измеренное с момента завершения приема последнего символа. Единица времени составляет 10 мкс. Этот выходной параметр Тип важен для определения времени молчания в таких протоколах, как MODBUS RTU. Это может быть не время молчания после последнего полученного символа этим функциональным блоком, так как оно верно только в том случае, если RX_REMAINING = 0.

Таблица 162: SERIAL\_RX\_EXTENDED Выходные параметры

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR

Receive_Ex: SERIAL_RX_EXTENDED;
Port: SERIAL_PORT := COM1;

Buffer_Pointer: ARRAY [0..1023] OF SERIAL_RX_CHAR_EXTENDED;
Status: SERIAL_STATUS;

END_VAR

//INPUTS:

Receive_Ex.REQUEST := TRUE;
Receive_Ex.PORT := Port;

Receive_Ex.RX_BUFFER_POINTER := ADR(Buffer_Pointer);

```

```
//OUTPUTS:

Receive_Ex.DONE;
Receive_Ex.EXEC;
Receive_Ex.ERROR;

Status := Receive_Ex.STATUS;  // Если нужно исправить ошибку.

Receive_Ex.RX_RECEIVED;

Receive_Ex.RX_REMAINING;
```

#### 5.14.1.7. SERIAL\_SET\_CTRL

Этот блок используется для записи управляющих сигналов (RTS и DTR), когда они доступны в последовательном порту. Он также может установить состояние занятости для передачи с помощью параметра BREAK, и его можно использовать только в том случае, если сигнал модема настроен для RS232\_MANUAL.



Рисунок 144: Блок для записи контрольных сигналов

Входные параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
<b>PORT</b>	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
<b>RTS_VALUE</b>	BOOL	Значение, записываемое в сигнал RTS.
<b>RTS_EN</b>	BOOL	Разрешает запись параметра RTS_VALUE.
<b>DTR_VALUE</b>	BOOL	Значение, записываемое в сигнал DTR.
<b>DTR_EN</b>	BOOL	Разрешает запись параметра DTR_VALUE.
<b>BREAK</b>	BOOL	Если это так, включает логический 0 (занято) в линии передачи.

Таблица 163: SERIAL\_SET\_CTRL Входные параметры

Выходные параметры	Тип	Описание
<b>DONE</b>	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
<b>EXEC</b>	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

## 5. КОНФИГУРАЦИЯ

<b>STATUS</b>	<b>SERIAL_STATUS</b>	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS: <ul style="list-style-type: none"><li>- NO_ERROR</li><li>- ILLEGAL_SERIAL_PORT</li><li>- PORT_BUSY</li><li>- HW_ERROR_UART</li><li>- HW_ERROR_REMOTE</li><li>- FB_SET_CTRL_NOT_ALLOWED</li><li>- NOT_CONFIGURED</li></ul>
---------------	----------------------	---

Таблица 164: SERIAL\_SET\_CTRL Выходные параметры

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```
PROGRAM UserPrg
VAR

Set_Control: SERIAL_SET_CTRL;
Port: SERIAL_PORT := COM1;
Status: SERIAL_STATUS;
END_VAR

//INPUTS:

Set_Control.REQUEST := TRUE;
Set_Control.PORT := Port;
Set_Control.RTS_VALUE := FALSE;
Set_Control.RTS_EN := FALSE;
Set_Control.DTR_VALUE := FALSE;
Set_Control.DTR_EN := FALSE;
Set_Control.BREAK := FALSE;

//FUNCTION:

Set_Control();

//OUTPUTS:
```

## 5. КОНФИГУРАЦИЯ

### 5.14.1.8. SERIAL\_TX

Этот функциональный блок используется для передачи буфера данных через последовательный порт и завершается только после того, как все байты были переданы или после тайм-аута (генерация ошибок).

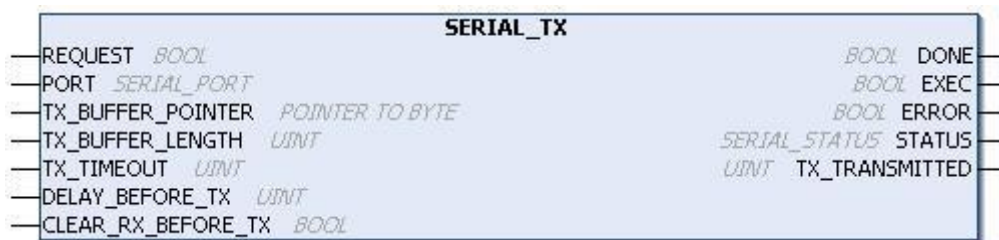


Рисунок 145: Блок передачи значений по серийному номеру

Входные параметры	Тип	Описание
<b>REQUEST</b>	BOOL	Эта переменная, если она принимает значение true, разрешает использование функционального блока.
<b>PORT</b>	SERIAL_PORT	Выберите последовательный порт, как описано в типе данных SERIAL_PORT.
<b>TX_BUFFER_POINTER</b>	POINTER TO BYTE	Указатель массива байтов для передачи значений буфера.
<b>TX_BUFFER_LENGTH</b>	UINT	Укажите ожидаемое количество символов в передаваемом массиве байтов (максимальный размер очереди TX — 1024 символа).
<b>TX_TIMEOUT</b>	UINT	Укажите тайм-аут для завершения передачи, включая фазу квитирования. Указанное значение в мс должно быть положительным и отличным от нуля.
<b>DELAY_BEFORE_TX</b>	UINT	Укажите задержку в мс между вызовом функционального блока и началом передачи. Эта переменная может использоваться при обмене данными с некоторыми модемами.
<b>CLEAR_RX_BEFORE_TX</b>	BOOL	При значении «истина» очередь RX и UART FIFO RX стираются перед началом передачи. Такое поведение типично для полудуплексных протоколов ведущий/ведомый.

Таблица 165: SERIAL\_TX Входные параметры

Выходные параметры	Тип	Описание
<b>DONE</b>	BOOL	Эта переменная принимает значение true, когда блок полностью выполнен. В противном случае это неверно.
<b>EXEC</b>	BOOL	Эта переменная имеет значение true, пока блок выполняется. В противном случае это неверно.
<b>ERROR</b>	BOOL	Эта переменная имеет значение true, когда блок завершает выполнение с ошибкой. В противном случае это неверно. Он связан с переменной DONE, так как ее состояние отображается после завершения блока.

Выходные параметры	Тип	Описание
<b>STATUS</b>	SERIAL_STATUS	В случае, если переменная ERROR имеет значение true, структура STATUS покажет ошибку, обнаруженную во время выполнения блока. Возможные состояния, уже описанные в типе данных SERIAL_STATUS: - NO_ERROR - ILLEGAL_SERIAL_PORT - PORT_BUSY - HW_ERROR_UART - HW_ERROR_REMOTE - ILLEGAL_TX_BUFF_LENGTH - ILLEGAL_TIMEOUT - CTS_TIMEOUT_ON - CTS_TIMEOUT_OFF - TX_TIMEOUT_ERROR - NOT_CONFIGURED
<b>TX_TRANSMITTED</b>	UINT	Возвращает номер переданного байта, который должен быть равен TX_BUFFER_LENGTH, но может быть меньше, если во время передачи произошла ошибка.

Таблица 166: SERIAL\_TX Выходные параметры

Пример использования на языке ST после вставки библиотеки в проект и настройки последовательного порта:

```

PROGRAM UserPrg
VAR

Transmit: SERIAL_TX;

Port: SERIAL_PORT := COM1;

Buffer_Pointer: ARRAY [0..9] OF BYTE := [0,1,2,3,4,5,6,7,8,9];
Status: SERIAL_STATUS;

END_VAR

//INPUTS:

Transmit.REQUEST := TRUE;
Transmit.PORT := Port;

Transmit.TX_BUFFER_POINTER := ADR(Buffer_Pointer);
Transmit.TX_BUFFER_LENGTH := 10;

Transmit.TX_TIMEOUT := 10000;

Transmit.DELAY_BEFORE_TX := 1000;
Transmit.CLEAR_RX_BEFORE_TX := TRUE;

//FUNCTION:

```

### 5.14.2. Обновление входов и выходов

По умолчанию локальная шина и встроенный ввод-вывод ЦП обновляются при каждом цикле выполнения MainTask. Функции обновления позволяют асинхронно обновлять эти точки ввода-вывода в любой точке кода пользовательского приложения.

Когда функциональные блоки для обновления входов и выходов не используются, обновление выполняется в каждом цикле Главной задачи.

#### ВНИМАНИЕ

При запуске ЦП этой серии входы и выходы только обновляются для чтения и подготавливаются для записи при выполнении MainTask.

Все остальные системные задачи, запускаемые до MainTask, будут иметь недействительные входы и выходы.

#### 5.14.2.1. REFRESH\_INPUT

Этот функциональный блок используется для обновления заданных входов модуля без необходимости ожидания завершения цикла. Важно отметить, что фильтры, настроенные в MasterTool IEC XE, и время обновления входов модуля должны быть учтены в эффективном времени обновления входов в приложении, разработанном

#### ВНИМАНИЕ

Функция REFRESH\_INPUT должна использоваться только в MainTask.

Чтобы обновить входы в других задачах, необходимо выбрать параметр «Включить обновление ввода-вывода для каждой задачи». Для получения дополнительной информации об этом параметре см. Таблицу 37.

пользователем.

#### ВНИМАНИЕ

Функция REFRESH\_INPUT не поддерживает входные данные, которые были сопоставлены с символьными переменными. Для правильной работы необходимо, чтобы ввод отображался на переменную в прямом представлении в памяти входных переменных (%I).

#### ВНИМАНИЕ

Функция REFRESH\_INPUT обновляет только прямые переменные %I, которые объявлены на вкладке «Шина: сопоставление входов/выходов» модуля, адресуемого в соответствующей стойке/слоте функции. В случае коммуникационных модулей/интерфейсов (MODBUS, Profibus и т. д.) обновление не включает прямые переменные сопоставления устройств.

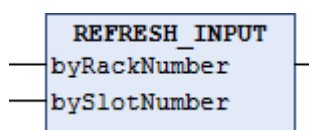


Рисунок 146: Блок для обновления ввода

Входные параметры	Тип	Описание
byRackNumber	BYTE	Номер стойки.
bySlotNumber	BYTE	Номер позиции, к которой подключается модуль.

Таблица 167: REFRESH\_INPUT Входные параметры

Возможный *ТИП РЕЗУЛЬТАТ*:

- **OK\_SUCCESS:** Успех выполнения.

## 5. КОНФИГУРАЦИЯ

- **ERROR\_FAILED:** Эта ошибка возвращается, если функция вызывается для модуля, который имеет только выходы, или если опция
- Всегда обновлять переменные (находятся на экране Конфигурация модуля, вкладка I/O Mapping) не отмечен.
- **ERROR\_NOTSUPPORTED:** Вызванная подпрограмма не поддерживается продуктом.
- **ERROR\_PARAMETER:** неверный/неподдерживаемый параметр.
- **ERROR\_MODULE\_ABSENT:** Модуль отсутствует в шине.
- **ERROR\_MODULE\_NOTCONRISUNOKD:** Модуль не настроен в приложении.
- **ERROR\_MODULE\_NOTRUNNING:** Модуль не запущен (не в рабочем состоянии).
- **ERROR\_MODULE\_COMMFAIL:** сбой связи с модулем.
- **ERROR\_MODULE\_NOTFOUND:** Модуль не найден в приложении или не поддерживается.

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR

Info: TYPE_RESULT;

byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR

//INPUTS:

byRackNumber := 0;

bySlotNumber := 10;
```

### 5.14.2.2. REFRESH\_OUTPUT

Этот функциональный блок используется для обновления указанных выходов модуля. Нет необходимости ждать окончания цикла. Важно отметить, что время обновления выходов модуля должно быть учтено в эффективном времени обновления выходов в приложении, разработанном пользователем.

#### ВНИМАНИЕ

Функция REFRESH\_OUTPUT должна использоваться только в MainTask.  
Для обновления выходов в других задачах необходимо выбрать параметр Включить обновление ввода-вывода для каждой задачи. Для получения дополнительной информации об этом параметре см. Таблицу 37.

#### ВНИМАНИЕ

Функция REFRESH\_OUTPUT не поддерживает входные данные, которые были сопоставлены с символьными переменными. Для правильной работы необходимо, чтобы ввод отображался на переменную в прямом представлении в памяти входных переменных (%Q).

#### ВНИМАНИЕ

Функция REFRESH\_OUTPUT обновляет только прямые переменные %Q, которые объявлены на вкладке «Шина: отображение входов/выходов» модуля, адресованного в соответствующей стойке/слоте функции. В случае коммуникационных модулей/интерфейсов (MODBUS, Profibus и т. д.) обновление не включает прямые переменные сопоставления устройств.

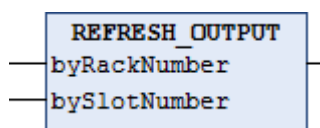


Рисунок 147: Блок для обновления вывода

Входные параметры	Тип	Описание
byRackNumber	BYTE	Номер стойки.
bySlotNumber	BYTE	Номер позиции, к которой подключен модуль.

Таблица 168: REFRESH\_OUTPUT Входные параметры

Возможный *ТИП РЕЗУЛЬТАТ*:

- **OK\_SUCCESS:** Успех выполнения.
- **ERROR\_FAILED:** Эта ошибка возвращается, если функция вызывается для модуля, который имеет только выходы, или если опция
- Всегда обновлять переменные (находятся на экране Конфигурация модуля, вкладка I/O Mapping ) не отмечен.
- **ERROR\_NOTSUPPORTED:** Вызванная подпрограмма не поддерживается продуктом.
- **ERROR\_PARAMETER:** неверный/неподдерживаемый параметр.
- **ERROR\_MODULE\_ABSENT:** Модуль отсутствует в шине. **ERROR\_MODULE\_NOTCONRISUNOKD:** Модуль не настроен в приложении. **ERROR\_MODULE\_NOTRUNNING:** Модуль не запущен (не в рабочем состоянии). **ERROR\_MODULE\_COMMFAIL:** сбой связи с модулем.
- **ERROR\_MODULE\_NOTFOUND:** Модуль не найден в приложении или не поддерживается.

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR

Info: TYPE_RESULT;

byRackNumber: BYTE;
bySlotNumber: BYTE;
END_VAR

//INPUTS:

byRackNumber := 0;

bySlotNumber := 10;

//FUNCTION:

```

### 5.14.3. Функциональный блок PID

#### ВНИМАНИЕ

Функциональный блок PID, описанный до предыдущей версии L данного руководства, устарел и был удален из этого руководства.

Функциональные блоки PID, PID\_INT и PID\_REAL, описанные до версии C MP399609, также устарели и были удалены из более новых версий этого руководства. Пользователи, которым требуется описание этих устаревших функциональных блоков по причинам технического обслуживания, должны использовать версию C MP399609.

Функциональные блоки PID, PID\_INT и PID\_REAL не должны использоваться в новых проектах. Эти функциональные блоки были заменены более новыми функциональными блоками с дополнительными ресурсами, такими как автоматическая настройка и поддержка каскадного управления, переопределения и прямой связи. Эти новые функциональные блоки описаны в MU214610 и доступны после версии 1.1.0.0 библиотеки NextoPID.

#### 5.14.4. Сохранение таймера

Сохранение таймера — это функциональный блок, разработанный для приложений, таких как часы производственной линии, которые должны сохранять свое значение и перезапускать отсчет с той же точки в случае сбоя питания. Значения, сохраненные функциональным блоком, равны нулю только в случае холодного сброса, сброса источника или загрузки нового приложения (см. Руководство пользователя MasterTool IEC XE - MU299609), когда счетчики продолжают работать, даже когда приложение остановлено (Режим остановки).

##### ВНИМАНИЕ

Важно подчеркнуть, что для правильного функционирования блоков Timer Retain переменные должны быть объявлены как Retain (VAR RETAIN). Также важно отметить, что в режиме симуляции функциональные блоки Timer Retain не работают должным образом из-за того, что для правильного поведения требуется ЦП Nexto.

Три блока, уже имеющиеся в библиотеке NextoStandard программного обеспечения MasterTool IEC XE, описаны ниже (о порядке вставки библиотеки см. Руководство по программированию MasterTool IEC XE — MP399609, раздел «Библиотека»).

##### 5.14.4.1. TOF\_RET

Функциональный блок TOF\_RET реализует временную задержку для отключения выхода. Когда состояние входа IN изменяется с (ИСТИНА) на (ЛОЖЬ) или спадающий фронт, будет отсчитываться заданное время PT, и в конце этого времени выход Q будет переведен в состояние (ЛОЖЬ). Когда вход IN находится в состоянии логической 1 (ИСТИНА), выход Q остается в том же состоянии (ИСТИНА), даже если это произошло в середине процесса счета. Время PT может быть изменено во время подсчета, так как блок принимает новое значение, если подсчет еще не завершен. На рисунке 148 показан блок TOF\_RET, а на рисунке 149 показано его графическое поведение.



Рисунок 148: Блок TOF\_RET

Входные параметры	Тип	Описание
IN	BOOL	Эта переменная, когда получает задний фронт, включает подсчет блоков.
PT	TIME	Эта переменная определяет предел счета блоков (временную задержку).

Таблица 169: TOF\_RET Входные параметры

Выходные параметры	Тип	Описание
Q	BOOL	Эта переменная выполняет задний фронт, когда переменная PT (временная задержка) достигает своего максимального значения.
ET	TIME	Эта переменная показывает текущую временную задержку.

Таблица 170: TOF\_RET Выходные параметры

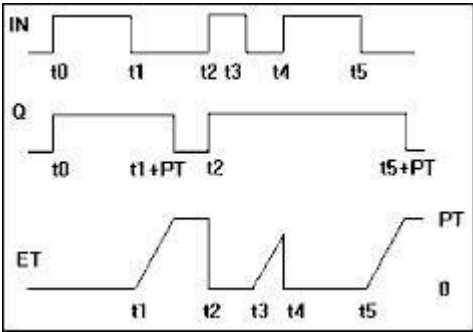


Рисунок 149: Блокировать графическое поведение TOF\_RET

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR RETAIN

bStart : BOOL := TRUE;
TOF_RET : TOF_RET;
END_VAR

// Когда bStart=FALSE начинает отсчет
TOF_RET( IN := bStart,

PT := T#20S);

// Действия, выполняемые в конце подсчета IF
(TOF_RET.Q = FALSE) THEN

bStart := TRUE;
END_IF
```

5.14.4.2. TON\_RET

TON\_RET реализует временную задержку для включения выхода. Когда состояние входа IN изменяется с (FALSE) на (TRUE) или нарастающий фронт, будет отсчитываться заданное время PT, и в конце этого времени выход Q будет переведен в состояние (TRUE). Когда вход IN находится в состоянии логического 0 (ЛОЖЬ), выход Q остается в том же состоянии (ЛОЖЬ), даже если это происходит в середине процесса счета. Время PT может быть изменено во время подсчета, так как блок принимает новое значение, если подсчет еще не завершен. На рисунке 150 показан блок TON\_RET, а на рисунке 151 — его графическое поведение.



Рисунок 150: Функциональный блок TON\_RET

Входные параметры	Тип	Описание
IN	BOOL	Эта переменная, когда получает нарастающий фронт, включает подсчет функциональных блоков.
PT	TIME	Эта переменная определяет предел счета блоков (временную задержку).

Таблица 171: TON\_RET Входные параметры

5. КОНФИГУРАЦИЯ

Выходные параметры	Тип	Описание
Q	BOOL	Эта переменная выполняет нарастающий фронт, когда переменная PT (временная задержка) достигает своего максимального значения.
ET	TIME	Эта переменная показывает текущую временную задержку.

Таблица 172: TON\_RET Выходные параметры

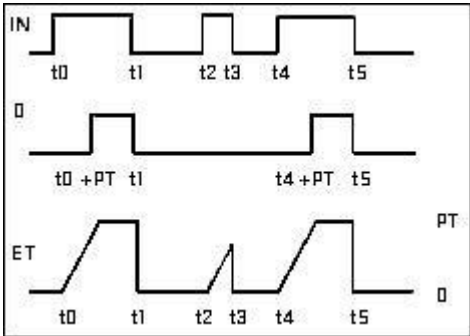


Рисунок 151: Блокировка графического поведения TON\_RET

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR RETAIN

bStart : BOOL;
TON_RET : TON_RET;
END_VAR

// Quando bStart=TRUE начинает считать
TON_RET( IN := bStart, PT := T#20S);

// Действия, выполняемые в конце подсчета

IF (TON_RET.Q = TRUE) THEN

bStart := FALSE;
END_IF
```

5. 14.4.3. TP\_RET

Функциональный блок TP\_RET работает как триггер. Таймер, который запускается, когда состояние входа IN изменяется с (FALSE) на (TRUE), то есть передний фронт, он увеличивается до тех пор, пока не будет достигнут предел времени PT. Во время подсчета выход Q равен (TRUE), в противном случае — (FALSE). Время PT может быть изменено во время подсчета, так как блок принимает новое значение, если подсчет еще не завершен. Рисунок 152 изображает TP\_RET, а Рисунок 153 показывает его графическое поведение.



Рисунок 152: Функциональный блок TP\_RET

Входные параметры	Тип	Описание
IN	BOOL	Эта переменная, когда получает нарастающий фронт, включает подсчет функциональных

5. КОНФИГУРАЦИЯ

		блоков.
PT	TIME	Эта переменная определяет предел счета функциональных блоков (временную задержку).

Таблица 173: TP\_RET Входные параметры

Выходные параметры	Тип	Описание
Q	BOOL	Эта переменная истинна во время подсчета, иначе ложна.
ET	TIME	Эта переменная показывает текущую временную задержку.

Таблица 174: TP\_RET Выходные параметры

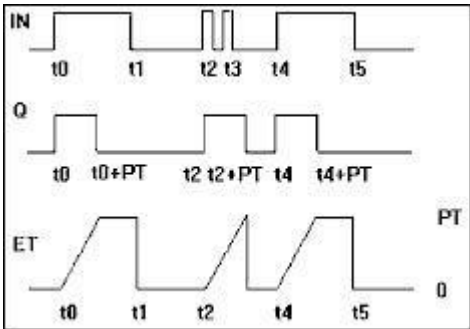


Рисунок 153: TP\_RET Блокировка графического поведения

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR RETAIN

bStart : BOOL;
TP_RET : TP_RET;
END_VAR

// Конфигурация TP_RET
TP_RET( IN := bStart,
```

```

bStart := FALSE;

// Действия, выполняемые во время подсчета
IF (TP_RET.Q = TRUE) THEN

// Выполняется, пока счетчик активирован ELSE

// Выполняется, когда счетчик деактивирован
END_IF

```

#### 5.14.5. Журнал пользователя

Функция, позволяющая пользователю создавать собственные записи и записывать файлы журнала на карту памяти, установленную в ЦП. Файлы создаются в определенном каталоге карты памяти в формате CSV, что позволяет просматривать их в текстовых редакторах и электронных таблицах. Разделителем был символ точки с запятой. Дополнительные сведения об использовании карты памяти см. в разделе Карта памяти.

Доступны две функции: одна для информации журнала, а другая для удаления всех записей. Ниже приведено описание типов данных, используемых функциями:

Тип данных	Вариант	Описание
USER_LOG_EVENT_TIPS	USER_LOG_EVENT_ERROR	Пользователь может использовать наилучшую индикацию в соответствии с серьезностью сообщения журнала.
	USER_LOG_EVENT_DEBUG	
	USER_LOG_EVENT_INFO	
	USER_LOG_EVENT_WARN	
USER_LOG_MESSAGE		Сообщение журнала длиной не более 150 символов.
USER_LOG_ERROR_CODES	USER_LOG_OK	Операция была проведена успешно.
	USER_LOG_FAILED	Операция не была проведена успешно. Причину сбоя можно проверить в системных журналах – см. раздел Системный журнал.
	USER_LOG_BUFFER_FULL	Сообщения добавляются за пределами возможностей обработки.
	USER_LOG_NO_MEMORY	В то время не было ресурсов для проведения операции.
	USER_LOG_FILE_SYSTEM_ERROR	Произошла ошибка при доступе к карте памяти или нет свободного места. Информацию об ошибках можно проверить в журналах системы – см. раздел Системный журнал.
	USER_LOG_NO_MEMORY_CARD	В процессоре есть карта памяти.
	USER_LOG_MEMORY_CARD_FULL	На карте памяти нет свободного места.
	USER_LOG_PROCESSING	Ресурс занят выполнением последней операции, например удалением всех лог-файлов.

Таблица 175: Тип данных для журнала пользователя

Ниже описаны две функции, доступные в библиотеке LibLogs в MasterTool IEC XE. Чтобы выполнить процедуру вставки библиотеки, см. Руководство по программированию MasterTool IEC – MP399609, раздел Библиотеки.

**ВНИМАНИЕ**

Пользовательские журналы доступны только до версии 1.3.0.20 процессоров серии Nexto. Так же для использования этой функции необходима версия 1.40 или выше MasterTool IEC XE.

## 5.14.5.1. UserLogAdd

Эта функция используется для добавления нового сообщения пользовательского журнала, добавляя новую строку в файл журнала на карте памяти. Сообщение должно иметь максимальную длину 150 символов и тип события сообщения. Переменные приложения могут быть зарегистрированы с помощью преобразования в строку и конкатенации с основным сообщением. Информация о дате и времени в формате UTC (отметка времени) автоматически добавляется в сообщение с разрешением в миллисекунды, в котором было зарегистрировано событие. Информация о дате и времени также используется при формировании имен файлов журнала.

Функцию UserLogAdd можно использовать для ввода нескольких сообщений в рамках одной задачи, а также в разных прикладных задачах. Однако независимо от каждого выполнения функции в приложении, выполнения одной и той же задачи или разных задач, все они используют одну и ту же функцию для записи нужных сообщений. По этой причине рекомендуется, чтобы добавление сообщений с помощью функции UserLogAdd в приложении удерживалось каждые 50 мс, чтобы предотвратить повторную перегрузку буфера. Если функция выполняется в периоды короче, чем указано, но соблюдается среднее время 50 мс между каждым добавлением сообщения в конце интервала для задачи, также предотвращается возврат перегрузки буфера. Чтобы журналы добавлялись корректно, важно соблюдать ограничения по времени при вставке карты и при запуске ЦП, указанные в разделе Карта памяти. После операции функция возвращает Варианты для заданного Типа USER\_LOG\_ERROR\_CODES в виде Таблицы 175. Когда функция возвращает USER\_LOG\_OK, сообщение не было зарегистрировано и функция UserLogAdd должна быть повторно выполнена с желаемым сообщением. В случае возврата последовательных сбоев записи карта памяти может быть повреждена. Замена на исправную карту памяти гарантирует, что последние зарегистрированные сообщения будут записываться на неповрежденную карту, поскольку ЦП не перезагружается.

На рисунке ниже представлены функции UserLogAdd и Таблица под входными параметрами:

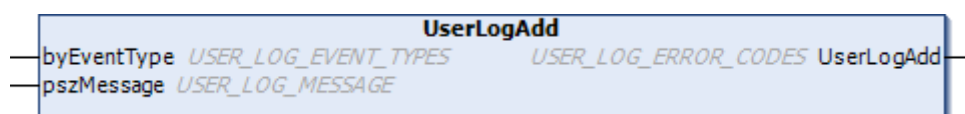


Рисунок 154: Функция UserLogAdd Function

Входные параметры	Тип	Описание
<b>byEventТип</b>	BYTE	Эта переменная указывает Тип события добавляемого журнала как Варианты для Типа данных USER_LOG_EVENT_TIPS.
<b>pszMessage</b>	USER_LOG_MESSAGE	Эта переменная должна содержать набор символов, составляющих сообщение, которое будет добавлено в файл журнала. Сообщение должно содержать не более 150 символов.

Таблица 176: Входные параметры UserLogAdd

Файлы журналов создаются и размещаются на карте памяти в определенном каталоге в зависимости от серийного номера ЦП и установленной версии микропрограммы. Например, для ЦП с серийным номером 445627 и версией микропрограммы 1.4.0.4 расположение, в которое файлы журнала должны быть записаны на карту памяти, — MemoryCard/UserLog/445627/1.4.0.4/.

Имена лог-файлов формируются по дате и времени (метке времени) первого сообщения. За исключением случаев, когда есть проблема с использованием этого имени, например, другой существующий файл с таким же именем, в этой ситуации используются мгновенные дата и время. Имя файла следует следующему шаблону: год/месяц/день/час/минута/секунда/миллисекунда.CSV. В случае проблем с доступом к файлу из-за дефектного сектора, не позволяющего продолжить запись, к имени этого файла будет добавлено расширение «поврежден» и будет создан новый файл. Количество журналов на файл не фиксировано и зависит от размера сообщений. Количество создаваемых файлов ограничено 1024 файлами с максимальным размером 1 МБ каждый, поэтому на карте памяти требуется 1 ГБ свободного места. При достижении лимита в 1024 файла, созданных на карте памяти, во время работы ЦП удаляются самые старые файлы, чтобы сохранить файлы с последними журналами, даже в случаях частичного ручного удаления файлов в каталоге, в котором находятся файлы.

## 5. КОНФИГУРАЦИЯ

Просмотр файлов журнала можно выполнять с помощью рабочих листов или обычных текстовых редакторов. Объединенная информация для улучшения визуализации может использовать точку с запятой между строками сообщения для их разделения. Нужно быть осторожным при форматировании ячеек со значениями с плавающей запятой.

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR

    eLogError : USER_LOG_ERROR_CODES;
    sMessage : USER_LOG_MESSAGE;

END_VAR

IF (m_rTemperature > MAX_TEMPERATURE_ACCEPT) THEN
    sMessage := 'Temperature higher than expected: ';

    sMessage := concat(sMessage, REAL_TO_STRING(m_rTemperature));
    sMessage := concat(sMessage, '°');
```

Пример содержимого файла журнала: (UserLog-201308271506245738.csv)

```
Model; NX3008
Serial number; 445627
Firmware version; 1.4.0.4

27/08/2013 15:06:24.5738; WARN; Overtemperature: 25°
```

### 5.14.5.2. UserLogDeleteAll

Функция UserLogDeleteAll выполняет удаление файлов журнала, находящихся в каталоге, созданном специально для ЦП, в который вставлена карта памяти, т. е. удаляются только журналы, содержащиеся в каталоге, названном версией микропрограммы ЦП, который существует в каталоге с ЦП. серийная версия. Удаляемые лог-файлы — это только те файлы, которые существуют на момент монтирования карты памяти и генерируются функцией UserLogAdd. Журналы других процессоров и файлы, добавленные пользователем вручную во время выполнения, не удаляются. Рисунок ниже представляет функцию UserLogDeleteAll.

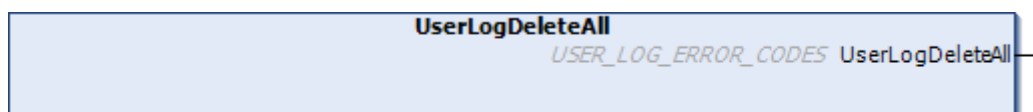


Рисунок 155: Функция UserLogDeleteAll

Пример использования на языке ST:

```
PROGRAM UserPrg
VAR

    eLogError : USER_LOG_ERROR_CODES;

END_VAR

IF (m_DeleteLogs = TRUE) THEN
    eLogError := UserLogDeleteAll();
    m_DeleteLogs := FALSE;
```

**ВНИМАНИЕ**

Возврат функции UserLogDeleteAll не указывает на завершение операции, а только на подтверждение выполнения, которое может занять много времени, если в каталоге есть сотни лог-файлов. Функция для записи нового пользовательского журнала сейчас недоступна, возвращая параметр USER\_LOG\_PROCESSING для любой операции. Результат операции также можно проверить в системном журнале.

**5.14.6. ClearRtuDiagnostic**

Эта функция не поддерживается этой серией процессоров.

**5.14.7. ClearEventQueue**

Функцию ClearEventQueue, доступную в библиотеке LibRtuStandard, можно использовать, когда необходимо очистить очередь событий ЦП и все экземпляры драйверов.

Согласно приведенной ниже Таблице результат выполнения функции будет отображаться в ее выходной переменной.

Наименование	ENUM (UDINT)	Описание результатов
OK_SUCCESS	0	Успех
ERROR_FAILED	1	Общая ошибка
ERROR_NOTSUPPORTED	2	Вызванная подпрограмма не поддерживается продуктом
ERROR_PARAMETER	3	Недопустимый/неподдерживаемый параметр
ERROR_MODULE_ABSENT	16	Модуль отсутствует в шине
ERROR_MODULE_NOTCONFIGURED	17	Модуль не настроен в приложении
ERROR_MODULE_NOTRUNNING	18	Модуль не запущен (не в рабочем состоянии)
ERROR_MODULE_COMMFAIL	19	Сбой связи с модулем
ERROR_MODULE_NOTFOUND	20	Модуль не найден в приложении или не поддерживается

Таблица 177: Результаты функции ClearEventQueue

Используя пример на языке ST, где вызов функции будет очищать очередь событий и, следовательно, сбрасывать диагностику использования очереди событий драйверов связи T\_DIAG\_DNP\_SERVER\_1.tClient\_\*.tQueueDiags.wUsage:

```

PROGRAM UserPrg
VAR

    ClearEventQueueStatus : TYPE_RESULT;
END_VAR

```

**5.15. Веб страницы пользователей**

Эта функция, также называемая Web Visualization или просто Webvisu, позволяет внедрить упрощенную SCADA, встроенную в ПЛК. Экраны визуализации разрабатываются в той же среде, что и приложение ПЛК, с использованием MasterTool IEC XE. После загрузки приложения ПЛК запускает веб-сервер, на котором размещена эта специальная веб-страница.

Полную информацию об этой функции можно найти в справке MasterTool IEC XE.

**5.16. Устройство****5.16.1. Управление пользователями и права доступа**

Он предоставляет функции для определения учетных записей пользователей и подтверждения прав доступа к проекту и к ЦП. С помощью программы MasterTool IEC XE можно создавать и управлять пользователями и

## 5. КОНФИГУРАЦИЯ

группами, настройками, различными уровнями прав доступа к проекту.

В то же время процессоры Nexto имеют систему управления правами пользователей, которая блокирует или разрешает определенные действия для каждой группы пользователей в процессоре. Для получения дополнительной информации см. Руководство пользователя MasterTool IEC XE MT8500 — MU299609 в разделе «Управление пользователями и права доступа».

### 5.16.2. Настройки ПЛК

На этой вкладке общего редактора устройств вы делаете основные настройки для Конфигурации ПЛК, например, обработку входов и выходов и задачу цикла шины.

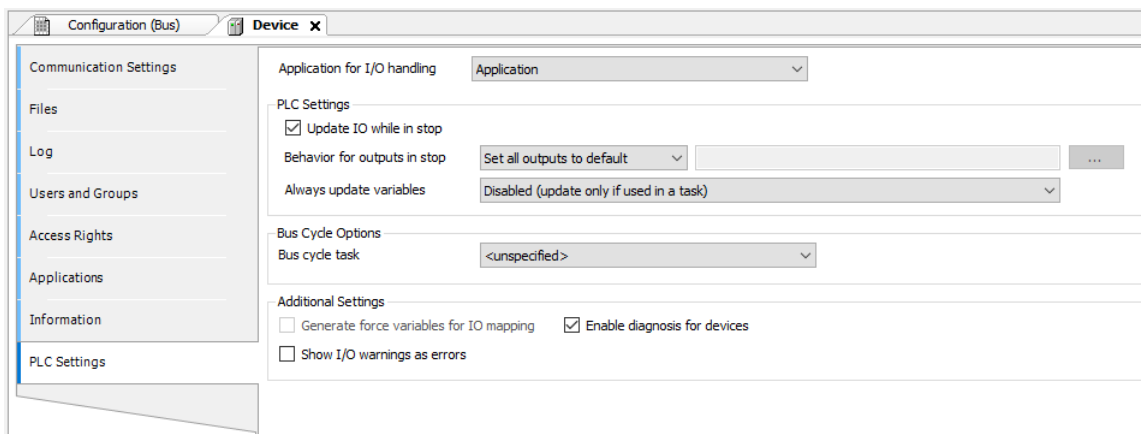


Рисунок 156: Настройки ПЛК

Параметр	Описание
Приложение для обработки ввода/вывода	Приложение, отвечающее за обработку ввода/вывода.

Параметр	Описание
Обновить ввод/вывод при остановке	<b>TRUE:</b> значения входных и выходных каналов также обновляются, когда ПЛК находится в режиме STOP. Если таймер контрольной системы обнаруживает неисправность, выходы устанавливаются на предустановленные значения По умолчанию. <b>FALSE:</b> значения входных и выходных каналов в режиме STOP не обновляются.
Поведение выходов при остановке	Обработка выходных каналов, когда контроллер переходит в режим STOP: Сохранение значений: текущие значения сохраняются. Все выходы со значением «По умолчанию»: назначаются значения «По умолчанию», полученные в результате сопоставления ввода-вывода. Выполнение программы: обработка выходных значений управляется программой, содержащейся в проекте, которая выполняется в режиме STOP. Введите название программы в поле справа.
Всегда обновлять переменные	Глобально определяет, обновляются ли переменные ввода/вывода в задаче цикла шины. Эта настройка действует для переменных ввода/вывода ведомых устройств и модулей только в том случае, если в их настройках обновления определено значение «деактивировано». Деактивировано (обновлять, только если они используются в задаче): переменные ввода-вывода обновляются, только если они используются в задаче. Включено 1 (использовать задачу цикла шины, если она не используется ни в одной задаче): Переменные ввода/вывода в задаче цикла шины обновляются, если они не используются ни в какой другой задаче. Включено 2 (всегда в задаче цикла шины): Все

## 5. КОНФИГУРАЦИЯ

	переменные в каждом цикле задачи цикла шины обновляются независимо от того, используются ли они и отображаются ли они на входной или выходной канал.
Задача автобусного цикла	Задача, управляющая циклом шины. По умолчанию вводится задача, определенная описанием устройства. По умолчанию применяются настройки цикла шины вышестоящего шинного устройства (используйте настройки цикла вышестоящей шины). Это означает, что дерево устройств просматривается вверх для следующего допустимого определения задачи цикла шины.
Принудительные переменные для сопоставления ввода/вывода	<b>TRUE:</b> при компиляции приложения для каждого канала ввода/вывода создаются две глобальные переменные, которые сопоставляются с переменной в диалоговом окне сопоставления ввода/вывода.
Активировать диагностику для устройств	<b>TRUE:</b> библиотека диагностики устройств САА интегрирована в проект. Неявный функциональный блок генерируется для каждого устройства. Если для устройства уже существует функциональный блок, то либо создается расширенный функциональный блок (пример: EtherCAT), либо добавляется другой экземпляр функционального блока. Затем он содержит общую реализацию диагностики устройства.
Отображать предупреждения ввода-вывода как ошибки	Предупреждения относительно Конфигурации ввода-вывода отображаются как ошибки.
Включить символический доступ для ввода/вывода	<b>TRUE:</b> входные и выходные переменные (VAR_INPUT и VAR_OUTPUT) автоматически создаются для каналов ввода/вывода устройства. Для этого для каждого ведомого создается расширенный функциональный блок. Основой является существующий функциональный блок ведомого устройства. Этот тип автоматически сгенерированного функционального блока может быть доступен непосредственно в коде приложения.

Таблица 178: Настройки ПЛК

## 5.17. Брандмауэр

Брандмауэр предназначен для повышения безопасности устройства во время его использования. Основная функция брандмауэра — фильтрация пакетов данных, входящих и исходящих из устройства. Реализованный фильтр использует информацию из каждого пакета данных, чтобы решить, разрешен ли этот пакет. Основными используемыми параметрами являются интерфейсы ввода/вывода, порт, протокол транспортного уровня, а также адреса источника и получателя.

### 5.17.1. Конфигурация

Конфигурация брандмауэра выполняется через специальный раздел, расположенный на вкладке «Управление» веб-страницы контроллера, как показано ниже:

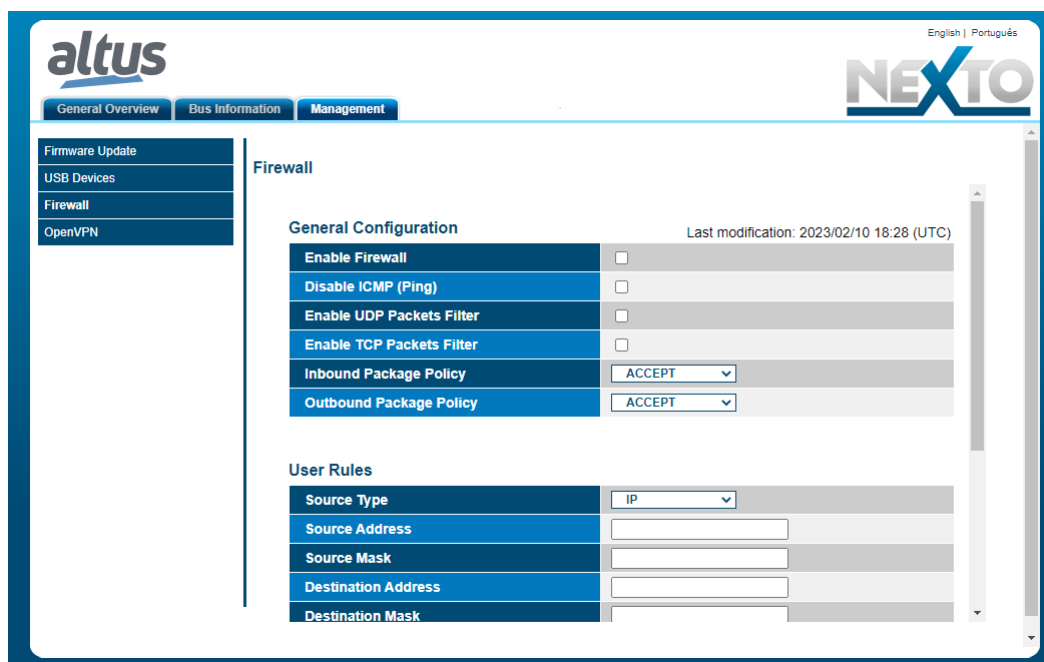


Рисунок 157: Экран конфигурации брандмауэра

Брандмауэр является отдельной функцией от MasterTool IEC XE, то есть он не требует никакого взаимодействия с инструментом программирования. Настройки, примененные на веб-странице, вступают в силу после подтверждения с помощью кнопки «Применить» и автоматически сохраняются в контроллере. Если функция включена, она снова будет работать даже после перезагрузки устройства.

В следующих разделах описываются возможные настройки брандмауэра, разделенные в соответствии с таблицами на этой веб-странице.

### 5.17.2. Общая конфигурация

На изображении ниже показаны все настройки в таблице общей конфигурации:

General Configuration		Last modification: 2023/02/10 18:28 (UTC)
Enable Firewall	<input type="checkbox"/>	
Disable ICMP (Ping)	<input type="checkbox"/>	
Enable UDP Packets Filter	<input checked="" type="checkbox"/>	
Filter UDP per IP	<input type="checkbox"/>	
UDP Packages per Second	<input type="text" value="0"/>	
Burst of UDP Packets	<input type="text" value="0"/>	
Enable TCP Packets Filter	<input checked="" type="checkbox"/>	
Filter TCP per IP	<input type="checkbox"/>	
TCP Packages per Second	<input type="text" value="0"/>	
Burst of TCP Packets	<input type="text" value="0"/>	
Inbound Package Policy	<input type="text" value="ACCEPT"/>	
Outbound Package Policy	<input type="text" value="ACCEPT"/>	

Рисунок 158: Таблица общих настроек брандмауэра

Эта Таблица динамически расширяется путем выбора Вариантов, чтобы включить фильтры пакетов UDP и TCP, показывая все элементы, которые могут быть обнаружены. Первый элемент этой Таблицы, Включить брандмауэр, используется для включения и отключения этой функции. Когда брандмауэр включен, настройки веб-страницы при отправке на устройство будут применяться к файлам Конфигурации, а затем брандмауэр будет фильтровать то, что было проверено Рисунокд. Если Брандмауэр отключен, сделанная Конфигурация сохраняется, но правила не применяются в контроллере.

Поле Отключить ICMP (Ping) включает или отключает защиту от протокола ICMP. При включенной защите контроллер не будет отвечать на Ping-запросы, так как будет отбрасывать пакеты, использующие протокол ICMP. Когда отключено, работа устройства для ответов Ping сохраняет нормальное поведение.

Когда включено, поля, которые включают фильтрацию пакетов UDP и TCP, фильтруют эти протоколы в соответствии с ограничениями, установленными в соответствующих полях. Правило фильтрации пакетов работает следующим образом: для того, чтобы пакет был принят, должны быть доступны «кредиты», и один кредит используется для принятия пакета данных.

Настройка поля Burst of XXX Packets задает начальную стоимость пакетов (кредитов), которые будут приняты. Таким образом, можно установить лимит переполнения для этих пакетов, где при большом потоке пакетов будет принято только количество conRiseunokd. Поле XXX Packages per Second указывает, сколько кредитов будет зарабатывать это правило в секунду. Например, если значение равно 5, то каждую секунду правило будет получать пять новых кредитов, поэтому оно сможет принять еще пять пакетов. Ограничением для этого приращения количества кредитов является сама Конфигурация Burst of XXX Packets, и установленный здесь предел не превышает, даже при приращении пакетов каждую секунду. Эти настройки применяются как запас, где при получении пакета данных сначала проверяется, есть ли доступный кредит в запасе, а затем принимается решение, принимать пакет или нет. Если пакет принят в этом количественном фильтре, он перенаправляется в фильтр других правил брандмауэра.

Параметр «Фильтровать XXX по IP» заставляет правило различать исходные адреса каждого пакета и применять фильтры «Пакет в секунду» и «Пакет переполнения» индивидуально для каждого IP-адреса Адреса. Итак, возвращаясь к предыдущему примеру, можно считать, что каждый исходный адрес имеет свой запас кредитов, и один адрес не может использовать кредиты, которые находятся в запасе, зарезервированном для других.

**ВНИМАНИЕ**

Отрицательные значения не допускаются для полей XXX Packages per Second и Burst of XXX Packets. Если заданы отрицательные значения, при применении настроек на экране появится сообщение об ошибке с указанием поля, в котором возник конфликт. Если фильтр включен, но значения в этих полях оставлены равными 0, фильтр не применяется.

Настройки в этой таблице применяются с помощью кнопки «Применить», которая появляется на рисунке 160.

## 5. КОНФИГУРАЦИЯ

В полях для выбора как входящей, так и исходящей политик есть Варианты для принятия и удаления. Если брандмауэр активен, при поступлении пакетов данных проверяются все настроенные правила, а затем применяется политика, настроенная для этих пакетов, будь то «Принять» или «Отбросить». Таким образом, если установлена политика принятия, Ассерпт, все пакеты, которые не соответствуют ни одному настроенному правилу, будут приняты брандмауэром, а если установлена политика отклонения, Drop, все они будут отброшены.

### 5.17.3. Правила пользователя

Таблица правил пользователя была создана, чтобы обеспечить больший контроль над настройками правил брандмауэра. С его помощью вы можете использовать Рисунок для различных правил динамически и с более точными фильтрами.

The screenshot shows the Altus NEXTO management interface. On the left is a sidebar with 'altus' logo and navigation tabs: General Overview, Bus Information, and Management. Under Management, there are links for Firmware Update, USB Devices, Firewall, and OpenVPN. The main area is titled 'Firewall' and contains the 'User Rules' configuration form. The form has the following fields: Source Type (dropdown set to IP), Source Address (text input), Source Mask (text input), Destination Address (text input), Destination Mask (text input), Interface (dropdown set to NET1), Action (dropdown set to ACCEPT), Service Port (dropdown set to Other... with a value of 1), Protocol (dropdown set to UDP/TCP), and Direction (dropdown set to INPUT/OUTPUT). An 'Add to list' button is located at the bottom right of the form. Below the form is a table with columns ID, Action, and User Rules. The table currently displays 'No rules are applied yet!'. An 'Apply' button is at the bottom right of the table area.

Рисунок 159: Таблица конфигурации правил пользователей брандмауэра

Эта Таблица меняет свой формат в соответствии с выбранным типом источника, который может быть IP или MAC. Когда типом является IP, таблица содержит элементы, показанные на рисунке выше, но когда тип выбран как MAC, поля маски источника и назначения исчезают, а также поле адреса назначения. Пункт Source Address теперь принимает MAC-адрес в качестве входных данных. Кроме того, правило MAC на основе адресов можно настроить только как правило ввода. Другими словами, поле Direction будет заполнено значением INPUT.

В полях «Адрес источника» и «Адрес назначения» вы можете ввести адреса, которые будут настроены для этого конкретного правила, а с помощью полей «Маска источника» и «Маска назначения» вы можете настроить сетевой диапазон для этого правила. Если вы настроили только адрес, правилу будет назначен только адрес, но с разными масками сети Конфигурации вы можете получить группы IP разных размеров, которые будут применяться к правилу.

Конфигурация интерфейса позволяет индивидуально выбрать каждый физический или виртуальный интерфейс, доступный для контроллера. В зависимости от того, какой интерфейс вы выберете для данного правила, брандмауэр будет фильтровать только пакеты данных, входящие или исходящие из интерфейса. Если вы выберете вариант Любой, это правило не будет иметь фильтра интерфейса. Таким образом, правило фильтрации будет действовать для всех доступных интерфейсов.

В поле «Действие» есть три варианта конфигурации: ПРИНЯТЬ, ОТМЕНИТЬ и ОТКЛОНИТЬ. Действие определяет, что следует делать с пакетом, характеристики которого соответствуют применяемому правилу. Если выбрано действие ПРИНЯТЬ, будет принят пакет данных, имеющий характеристики в соответствии с правилом. Если это DROP, пакет будет отброшен, и отправителю пакета не будет отправлено никакого ответа. Наконец, если установлено значение REJECT, пакет будет отклонен, а отправителю будет отправлен ответ о том, что запрошенный узел недоступен.

Поле Service Port используется для указания того, какие порты будут настроены в этом правиле. Все служебные порты, которые имеют определенный протокол или стандарт связи для контроллера, например, протокол MODBUS со стандартным портом 502, доступны с именем службы и портом, используемыми рядом с ним. Таким образом, если вы настроили правило для протокола MODBUS, будет применен порт 502, если вы настроили правило для сервиса WebVisu, будет применен порт 8080 и так далее для остальных протоколов, перечисленных в чекбоксе.

В этом поле также есть две другие настройки: «Любой» и «Другой». При выборе параметра Любой правило применяется ко всем служебным портам, кроме служб, за исключением порта 80, после чего создаются два

## 5. КОНФИГУРАЦИЯ

правила с использованием следующих диапазонов портов: 1:79 и 81:65535. Если вы выберете опцию Other, появится текстовое поле, в котором вы можете настроить порт, который вы хотите, за исключением порта 80. Чтобы настроить порт, вы можете ввести его номер в текстовое поле, но если вы хотите добавить более один порт, вы должны использовать разделитель «&», и если вы хотите вставить диапазон портов, просто введите начальный и конечный порт, используя разделитель «:».

Пример настройки портов 120, 144 и диапазона от 1300 до 1450 в одном поле: 120 и 144 и 1300:1450. Это поле не принимает значений за пределами диапазона 1:65535, порта 80 или повторов порта. HTTP-порт 80 можно задать, только выбрав его из списка известных протоколов, и его нельзя применить к интерфейсу NET 1. Таким образом, если выбран протокол HTTP, поля интерфейса NET 1 и Any не будут доступны для выбора.

В поле «Протокол» вы можете выбрать между протоколами UDP, TCP и UDP/TCP. Если вы выберете опцию UDP/TCP, на брандмауэре будут созданы два правила, по одному для каждого транспортного протокола.

В поле Direction вы можете выбрать между INPUT, OUTPUT и INPUT|OUTPUT. Эти Варианты заставляют правило применяться к пакетам, прибывающим на устройство, опция INPUT, или уходящим с него, опция OUTPUT. Если параметр соединения настроен, будут созданы два правила, по одному для каждого параметра направления.

На рисунке ниже показано, как применяется правило:

ID	Action	User Rules
0	Accept as input	Source is 192.168.18.120/24 destination is 192.168.18.17/21 on port 8080 ov...
1	Accept as input	Source is 192.168.18.120/24 destination is 192.168.18.17/21 on port 8080 ov...

Рисунок 160: Таблица применения правил пользователей брандмауэра

После заполнения полей, как вы хотите подтвердить Правило брандмауэра, вы должны нажать кнопку Добавить в список. При этом все настройки будут проанализированы на наличие недопустимых значений или дублирующихся правил. Невозможно добавить два правила с одинаковыми параметрами адреса, маски, интерфейса, порта и направления. При обнаружении конфликта будет выведено сообщение с указанием поля, содержащего недопустимую настройку, или идентификационного номера правила в Таблице, настройки которого вызвали конфликт с вновь настроенным.

После того, как все параметры будут проверены, правило будет добавлено в список под Конфигурация Таблицы. Этот список автоматически расширяется по мере добавления или удаления правил. Если вы хотите исключить правило из списка, вы можете навести указатель мыши на правило, которое хотите исключить. Когда вы это сделаете, справа появится красная кнопка X, как показано на предыдущем рисунке. Нажав на нее, правило будет удалено из Таблицы.

При добавлении новых правил или удалении существующих в таблице правил необходимо нажать кнопку Применить ниже, чтобы Конфигурация была применена к устройству.

### ВНИМАНИЕ

Во время применения правил брандмауэра может возникнуть кратковременная нестабильность связи Ethernet.

## 5.18. OpenVPN

VPN (виртуальная частная сеть), используемая для просмотра незащищенных сетей, передачи данных или просто доступа в Интернет с высоким уровнем безопасности и конфиденциальности. Виртуальную сеть VPN можно понимать как туннель, по которому информация перемещается безопасно, защищенная сертификатами безопасности и ключами. OpenVPN — это сервис с открытым исходным кодом, что означает, что его можно использовать и распространять бесплатно, а его исходный код открыт для модификаций при необходимости.

Основная цель VPN — безопасное общение в незащищенной сети. Чтобы сделать это возможным, используется шифрование данных на основе сертификатов и ключей, сгенерированных с использованием TLS, Transport Layer Security, протокола, выполняющего 256-битное шифрование, одного из самых безопасных.

Для выполнения Конфигурации клиента или сервера OpenVPN была создана страница OpenVPN на вкладке «Управление» веб-страницы ЦП. Как показано на рисунке ниже.

The screenshot displays the OpenVPN configuration interface. On the left, there is a sidebar with navigation links: Firmware Update, USB Devices, Firewall, and OpenVPN. The main area is titled 'OpenVPN' and contains a 'General Configuration' section. At the top of this section is an 'Enabled' checkbox and an 'Import' button. Below these are several configuration fields, each with a label and a value or dropdown menu:

Mode	Server
IP Address	10.8.123.0
Mask Address	255.255.255.0
Communication between Clients	Enable
Maximum Connected Clients	7
Protocol	UDP
Topology	subnet
Logs Level	5
Keep Alive Ping	10
Keep Alive Timeout	30
CA Certificate	sample_server_inline_ca

Рисунок 161: Экран конфигурации OpenVPN

Поскольку она находится на вкладке «Управление», доступ к этой странице защищен паролем. В следующих разделах описываются настройки и функции этой страницы.

### 5.18.1. Импорт Конфигурация

Чтобы быстро и легко настроить VPN на своем устройстве, вы можете воспользоваться кнопкой Import, которая изображена на картинке 161 в правом верхнем углу страницы. При нажатии на эту кнопку открывается окно файлового менеджера, в котором вы можете выбрать файл Конфигурация. Следует выбирать файлы с расширением .conf или .ovpn. Когда вы выбираете файл, его содержимое будет прочитано, и присутствующие параметры Конфигурации заполнят соответствующие поля Конфигурации на веб-странице.

Чтобы параметры файла интерпретировались правильно, они должны соответствовать стандартному синтаксису файла Конфигурация OpenVPN.

Если в файле Конфигурация есть файлы безопасности, сертификаты или ключи, записанные вместе с другими параметрами, они будут прочитаны и разделены на отдельные файлы внутри контроллера для использования.

## 5.18.2. Конфигурация OpenVPN

OpenVPN	
Mode	Server
IP Address	10.8.123.0
Mask Address	255.255.255.0
Communication between Clients	Enable
Maximum Connected Clients	7
Protocol	UDP
Topology	subnet
Logs Level	5
Keep Alive Ping	10
Keep Alive Timeout	30
CA Certificate	sample_server_inline_ca
Device Certificate	sample_server_inline_device
Device Key	sample_server_inline_device
TA Key	sample_server_inline_ta
Key of TA	SHA256

Рисунок 162: Таблица конфигурации сервера OpenVPN

☐ Enabled Import

OpenVPN	
Mode	Client
Remote IP	192.168.16.127
Protocol	UDP
Logs Level	3
Keep Alive Ping	5
Keep Alive Timeout	20
CA Certificate	device_1_bruno_berwanger@altus_com_br@berwange
Device Certificate	device_1_bruno_berwanger@altus_com_br@berwange
Device Key	device_1_bruno_berwanger@altus_com_br@berwange
TA Key	device_1_bruno_berwanger@altus_com_br@berwange
Key of TA	SHA256

Apply

Рисунок 163: Таблица конфигурации клиента OpenVPN

В этом разделе показано, как выполняется Конфигурация OpenVPN. Настройки будут разделены на три части: настройки, общие для обоих режимов работы, настройки, уникальные для сервера, и настройки, уникальные для клиента.

### 5.18.2.1. Общие конфигурации

Глядя на Рисунки с клиентской Конфигурацией, Рисунок 163, и серверной Конфигурацией, Рисунок 162, можно определить, что некоторые параметры одинаковы для обеих Конфигураций. Это:

#### 5.18.2.1.1. Режим

С помощью Конфигурации Режимы вы можете выбрать один из двух Вариантов, клиент или сервер. При выборе одного из двух режимов настройки Таблица автоматически меняются, чтобы разрешить Конфигурацию необходимых полей для каждого режима работы.

#### 5.18.2.1.2. Протокол

В этом поле настраивается, какой транспортный протокол будет использоваться для связи VPN. Он может быть установлен между UDP и TCP

#### ВНИМАНИЕ

Конфигурация сервера и всех его клиентов должна быть одинаковой. С расходящейся конфигурацией OpenVPN не может осуществляться связь.

#### 5.18.2.1.3. Уровень журналов

В этом поле задается уровень, который получит файл журнала. Диапазон значений от 0 до 5, где 0 — самый базовый уровень, а 5 — самый продвинутый.

Уровень 0 отображает только журналы о некоторых критических сбоях в OpenVPN, а уровни 4 и выше используются для отладки, поскольку в файл журнала записывается много информации. Для нормальной работы рекомендуется использовать значение 3.

Это поле принимает только числа в качестве ввода. Вы не можете использовать буквы или специальные символы.

#### 5.18.2.1.4. Пакет проверки активности

В этом поле задается время в секундах, когда запрос Ping будет перенаправлен. Этот запрос служит для проверки соединения между сервером и клиентами.

Этот параметр можно установить как на сервере, так и на клиентах OpenVPN, но если этот параметр установлен на сервере, клиенты будут использовать значение сервера, а не значение, установленное для них. Если сервер не имеет такой настройки, каждый клиент принимает свою настройку в обычном режиме. Если вы хотите отключить пинг между сервером и клиентами, установите значение 0.

Это поле принимает только числа в качестве ввода. Вы не можете использовать буквы или специальные символы.

#### 5.18.2.1.5. Тайм-аут поддержания активности

В этом поле задается время в секундах, когда истечет время ожидания запроса Ping. По истечении этого времени, без ответа от другого VPN-устройства, оно будет считаться отключенным.

Этот параметр можно установить как на сервере, так и на клиентах OpenVPN, но если этот параметр установлен на сервере, клиенты будут принимать половину значения сервера, а не значение, установленное на них. Клиенты получают половину суммы, чтобы убедиться, что они отключены в случае отключения сервера. Если сервер не имеет такой настройки, каждый клиент принимает свою настройку в обычном режиме. Если вы хотите отключить эту функцию, установите значение 0.

Это поле принимает только числа в качестве ввода. Вы не можете использовать буквы или специальные символы.

#### 5.18.2.1.6. Файлы безопасности

В полях Сертификат ЦС, Сертификат устройства, Ключ устройства и Ключ ТА вы должны выбрать, какой файл безопасности, сертификат или ключ будут использоваться для установления соединения OpenVPN. Варианты в каждом поле со списком фильтруются в соответствии с типом файла ключа или сертификата, хотя различия между ключами и сертификатами отсутствуют.

Чтобы можно было выбрать файл, он должен быть сначала импортирован.

Все файлы безопасности необходимы для правильной связи между клиентами и VPN-сервером, за исключением ключа ТАР. Этот ключ является необязательным для связи, но если он используется на сервере, он становится обязательным для всех клиентов на сервере.

Дополнительную информацию о создании сертификатов и ключей безопасности на основе TLS см. в разделе «Управление ключами и сертификатами TLS».

## 5. КОНФИГУРАЦИЯ

---

### 5.18.2.1.7. Ключ ТА

В поле Ключ ТА указывается, какой Тип шифрования будет применяться к Ключу ТА. Это поле остается скрытым до тех пор, пока вы не выберете файл для ключа TLS, поскольку оно используется только вместе с этим ключом. Значение по умолчанию для этого параметра — SHA1, но вы можете выбрать одно из следующих значений: SHA256, SHA512 и MD5 в дополнение к SHA1 по умолчанию.

### 5.18.2.2. Эксклюзивная конфигурация сервера

Эксклюзивные Конфигурации сервера, показанные на рисунке 162, описаны ниже.

#### ВНИМАНИЕ

Эта настройка должна быть одинаковой для клиентов и серверов в одной и той же сети OpenVPN. Если значение этого поля отличается между клиентом и сервером, соединение не будет установлено.

#### 5.18.2.2.1. Сетевой адрес

Диапазон IP-адресов, который будет использоваться для назначения адресов сервера и клиента для сети VPN, определяется сервером путем установки полей Address IP и Mask Address. Все IP-адреса, которые будут назначены клиентам и серверу, будут взяты из указанного диапазона.

IP-адрес адреса сервера всегда является первым доступным значением в диапазоне configured, а для назначения IP-адресов клиентам используются значения, все еще доступные в диапазоне, поэтому первое доступное значение назначается при подключении клиентов. Например, если сеть настроена с адресами 10.8.12.4 и маской 255.255.255.248, сервер примет IP 10.8.12.5, который является первым доступным адресом в диапазоне разрешенных. Однако, если установлена маска 255.255.255.255.0, сервер примет IP 10.8.12.1, который является первым доступным адресом в диапазоне.

Поля IP-адреса и адреса маски принимают только те настройки, которые имеют синтаксис IP-адреса и адреса маски соответственно.

Если настроено что-либо, выходящее за рамки стандарта, появится предупреждающее сообщение, информирующее вас о том, что произошла ошибка.

#### 5.18.2.2.2. Общение между клиентами

В этом поле вы можете включить или отключить связь между клиентами в сети VPN. Если для параметра выбрано значение Disabled, напрямую может выполняться только взаимодействие клиент-сервер. Если выбранная опция включена, она разрешит связь между самими клиентами в дополнение к связи клиент-сервер.

#### 5.18.2.2.3. Максимальное количество подключенных клиентов

В этом поле вы можете установить максимальное количество клиентов, которые могут одновременно подключаться к серверу. Это поле принимает только числовые символы, а минимальное значение равно 1.

#### 5.18.2.2.4. Частные сети

Когда вы выбираете режим работы OpenVPN в качестве сервера, будет отображаться Таблица, обычно скрытая, что позволяет Конфигурацию частных сетей, которые могут быть ниже сервера и каждого клиента.

altus NEXTO

English | Español | Português

CPU Overview System Overview CPU Management

Firmware Update  
USB Devices  
Firewall  
OpenVPN

### OpenVPN

Keep Alive Timeout: 30

CA Certificate: sample\_server\_inline\_ca

Device Certificate: sample\_server\_inline\_device

Device Key: sample\_server\_inline\_device

TA Key: sample\_server\_inline\_ta

Key of TA: SHA256

Private Networks			
Type	Common Name	Address	Mask
Server	server	192.168.80.0	255.255.255.0
Client	client2	192.168.70.0	255.255.255.0
Server			

Apply

Security Files

Select certificates and keys: Escolher arquivos Nenhum arquivo escolhido

Рисунок 164: Таблица конфигурации частной сети OpenVPN

Чтобы настроить частную сеть, расположенную ниже сервера, просто выберите тип сети в качестве сервера и настройте сетевые адреса и маску. Настройка частной сети для клиента требует, помимо установки Типа в качестве Клиента, ввода Общего имени клиента, владеющего сетью, которую проверяют.

Общее имя клиента задается при создании сертификата устройства. Этот параметр вводится при создании сертификата и уникален для каждого клиента и сервера. Конфигурация этих частных сетей создает таблицу маршрутизации, которая будет проверяться при получении или отправке пакетов через VPN.

На рисунке выше показана Конфигурация подсети 80 на сервере OpenVPN, затем будет настроено правило маршрутизации, которое будет перенаправлять пакеты данных, которые будут получены VPN, на интерфейс устройства, настроенный в этой сети. Он также создает правило, внутреннее для сервера, согласно которому, если пакет данных имеет подсеть 70, этот пакет будет маршрутизироваться и пересылаться через туннель VPN. То же самое происходит с клиентом client2, но с переключением подсетей, потому что ниже этого клиента находится подсеть 70, и он будет пересылать пакеты с подсетью 80 в VPN-туннель.

См. следующий рисунок для примера архитектуры:

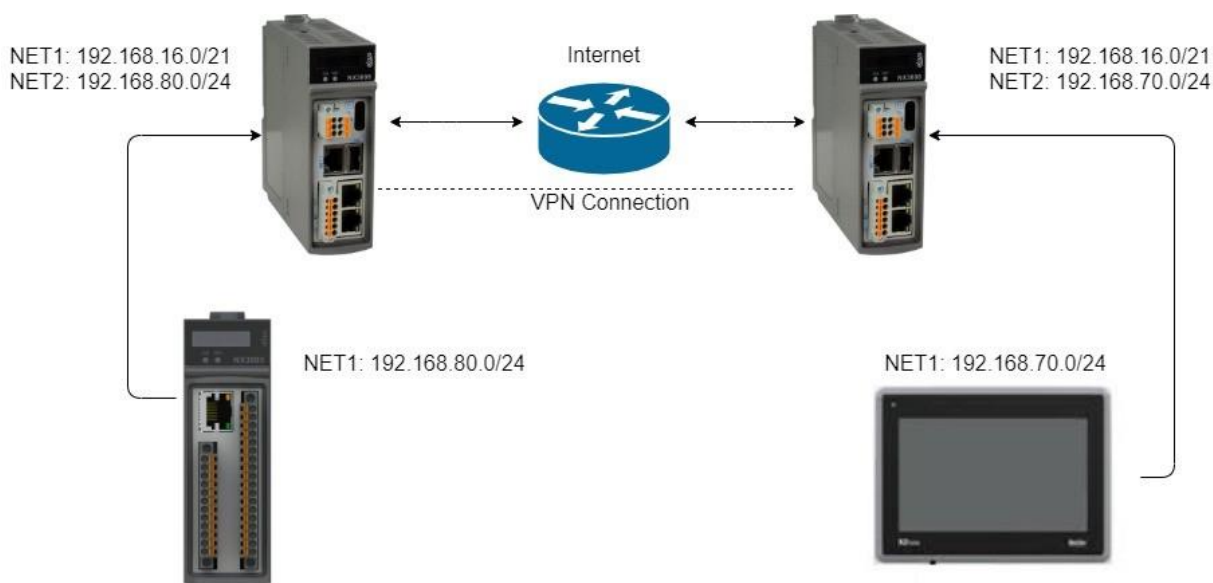


Рисунок 165: Пример архитектуры с частными сетями

## 5. КОНФИГУРАЦИЯ

На примере изображения NX3008 слева имеет частную сеть 80, настроенную на его NET 2, и к ней подключен NX3003 в той же сети. NX3008 справа имеет частную сеть 70, настроенную на его NET 2 и подключенную к ней ЧМИ в той же сети. Пример архитектуры реализует связь между NX3003 и устройствами НМІ через VPN путем настройки их соответствующих частных сетей.

После заполнения полей, показанных на Рисунок 164, нужной Конфигурацией, необходимо нажать синюю кнопку +, которая появляется справа от полей Конфигурация, чтобы правило было добавлено в Таблицу. Если вы хотите удалить правило, перетащите указатель мыши на правило, которое хотите удалить, и справа появится красный крестик, как показано на изображении 164. Щелчком по этому крестику правило удаляется из Таблицы.

Чтобы настройки, представленные в таблице, были применены к устройству, необходимо нажать кнопку «Применить» и подтвердить операцию в появившемся окне подтверждения.

### 5.18.2.3. Эксклюзивные клиентские конфигурации

На странице есть только одна Конфигурация, уникальная для клиентов OpenVPN, которую вы можете видеть на рисунке 163. Эта Конфигурация является IP Remote.

#### 5.18.2.3.1. Удаленный IP

В поле «Удаленный IP» задается адрес, по которому VPN-сервер ожидает связи с клиентами. Если на компьютере установлен сервер OpenVPN, удаленную Конфигурацию IP необходимо выполнить в соответствии с IP-адресом этого компьютера. Это поле также принимает имена хостов в качестве удаленного адреса, поэтому вы можете установить IP или имя хоста в этом параметре.

#### ВНИМАНИЕ

Из-за необходимости учитывать такие разные параметры, IP-адреса и имена хостов единственная проверка, которая существует в этом поле, — это наличие данных. Будьте внимательны при выполнении настройки.

#### 5.18.2.4. Настройки приложения

Чтобы включить эту функцию, необходимо установить флажок «Включено», показанный на рисунке выше. Если вы хотите просто применить сделанные вами настройки и не включать OpenVPN, снимите этот флажок.

После того, как вы сделали все нужные настройки, настройки необходимо применить к устройству, для этого используйте кнопку Применить. Эта кнопка показана на Рисунке 163 в правом нижнем углу. Когда настройки применены и VPN включена, веб-страница выполнит автоматическую прокрутку к таблице статуса OpenVPN, отображаемой в разделе Таблица статуса.

### 5.18.3. Файлы безопасности

Файлы безопасности используются для безопасного установления связи OpenVPN, выполняя роль шифрования и дешифрования пакетов данных, которые будут проходить через VPN-туннель. В разделе «Управление ключами и сертификатами TLS» описано, как создавать ключи и сертификаты TLS. Вот скриншот, на котором показан раздел, отвечающий за управление файлами безопасности:

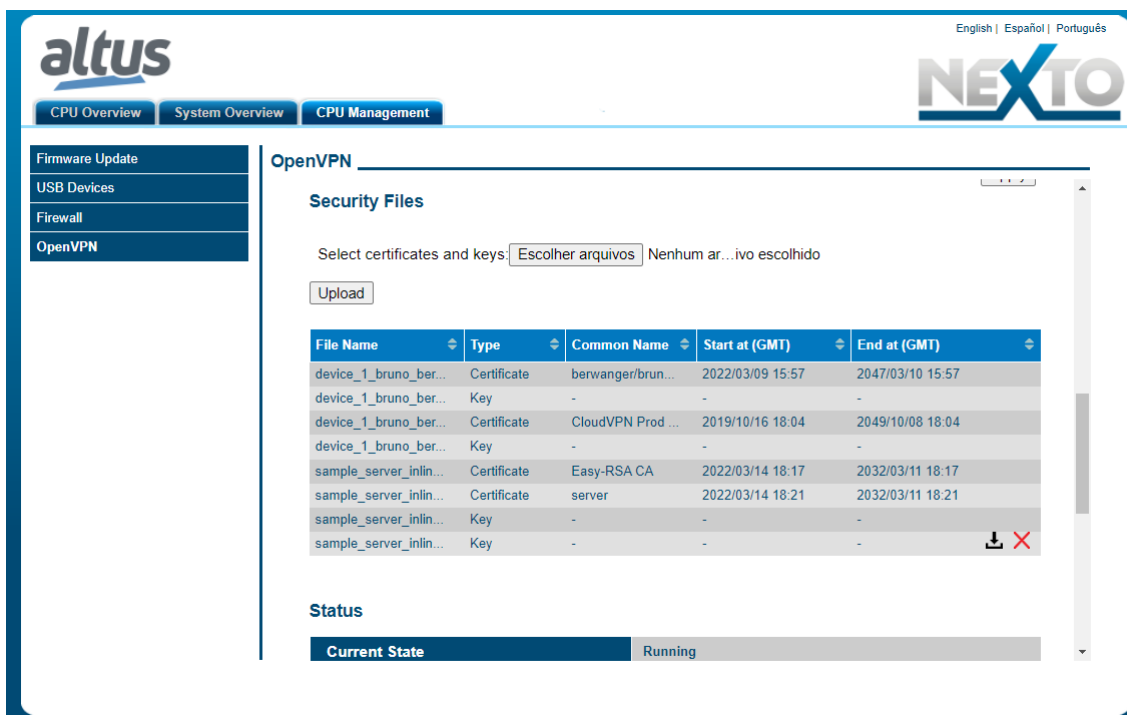


Рисунок 166: Таблица файлов безопасности OpenVPN

В этом разделе веб-страницы вы можете управлять файлами безопасности. Вы можете импортировать файлы, контролировать действительность сертификатов, загружать файлы, загруженные на устройство, и удалять загруженные файлы.

Нажав кнопку **Выбрать файлы**, вы можете импортировать сертификаты и ключи, эти файлы должны иметь соответствующие расширения **.crt** и **.key**. Эта кнопка открывает окно файлового менеджера и позволяет выбрать один файл, т. е. несколько файлов.

**ВНИМАНИЕ**

В контроллер можно импортировать не более 12 файлов.

Управление файлами осуществляется в Таблице, которая показана на рисунке 166. Эта Таблица добавляет новые элементы или удаляет их по мере выполнения операций импорта или удаления. Вы можете определить, является ли файл ключом или сертификатом, по второму элементу в списке, Типу, который указывает, что это за файл. Для сертификатов также отображаются их общие имена и даты истечения срока действия, как начала, так и окончания срока действия.

Вы можете восстановить файл, который был импортирован в деталь, а также удалить его. Когда вы наводите указатель мыши на файл в Таблице, появляются две кнопки, одна для скачивания и одна для удаления. Кнопка загрузки представляет собой черную стрелку, указывающую вниз, а кнопка удаления — красную X.

**5.18.4. Таблица статуса**

Разработанная для мониторинга данных, таблица статуса OpenVPN автоматически расширяется при изменении настроек и отображает различные данные о соединении, такие как состояние VPN, IP-адрес VPN, назначенный этому устройству, передаваемые данные и используемые файлы безопасности. для общения.

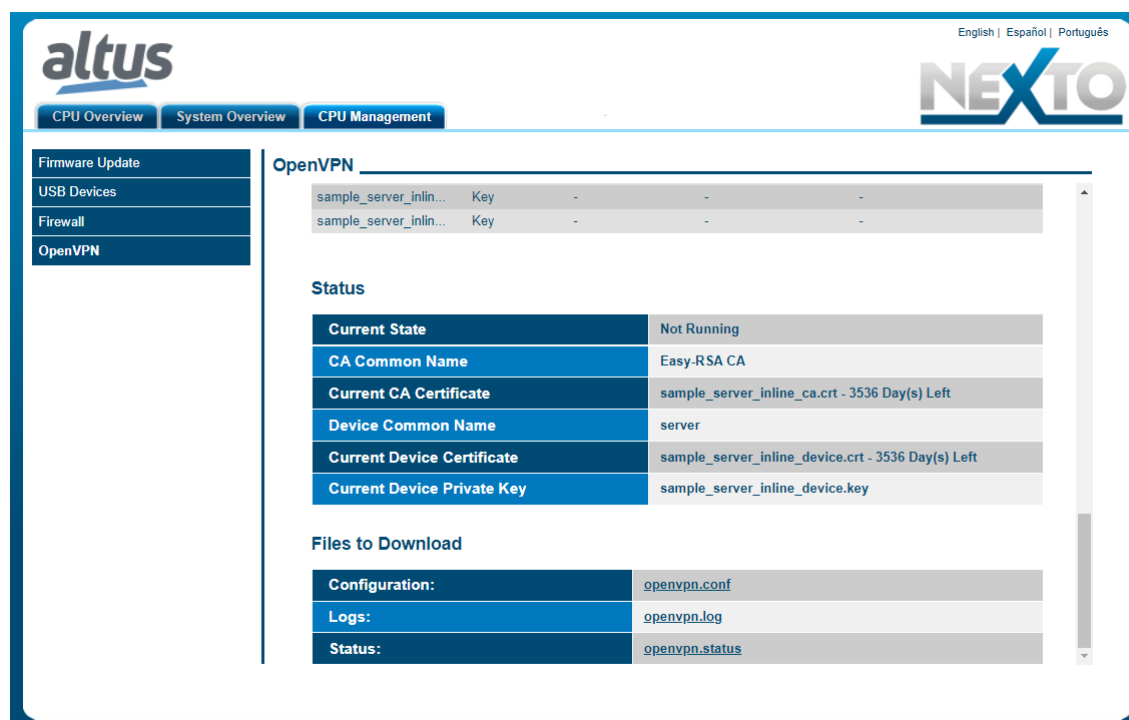


Рисунок 167: Таблица статуса OpenVPN с отключенной функцией

Когда VPN отключен, Таблица имеет несколько параметров. Поле Текущее состояние указывает, включена ли VPN, а другие поля показывают, какие сертификаты и ключи используются для связи VPN. Если один из файлов безопасности не был выбран, вместо его имени появится символ «-», указывающий, что файл не сконфигурирован.

В полях общего имени для центра сертификации и устройства отображаются имена, присвоенные соответствующим сертификатам, центру сертификации и устройству.

Рядом с именем файла каждого сертификата отображается оставшееся время в днях до истечения срока его действия.

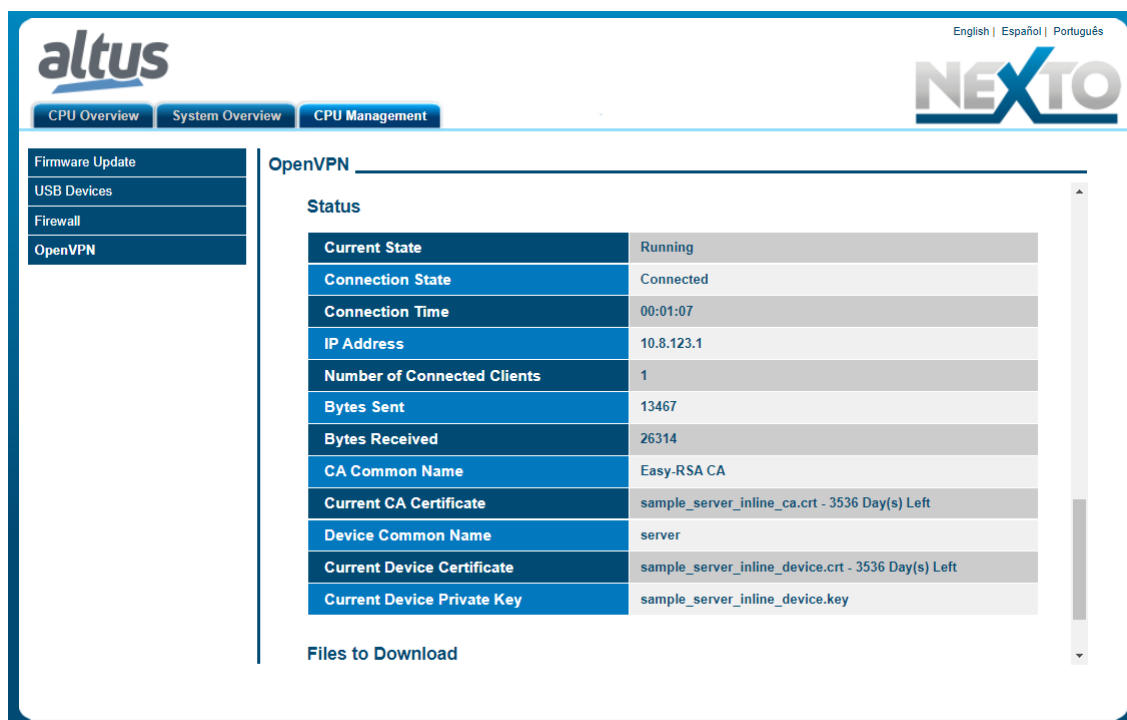


Рисунок 168: Таблица статуса OpenVPN с включенной функцией

Когда функция включена и настройки применяются к устройству, ячейки Таблицы динамически изменяются, чтобы отображалась оставшая информация. Информацию о статусе подключения OpenVPN можно найти в первых двух темах списка.

Элемент Current State имеет состояния Not Running, Starting service... и Running, которые соответственно указывают на то, что VPN отключена, запускается или включена.

Элемент «Состояние подключения» имеет состояния «Не подключено», «Подключение...» и «Подключено».

Другая информация, которую можно получить из этой Таблицы, — это общее время подключения, IP-адрес устройства и объем отправленных и полученных данных в байтах. Состояние того, сколько клиентов подключено в данный момент, отображается только тогда, когда OpenVPN работает как сервер.

### 5.18.5. Файлы для загрузки

Вы можете проверить информацию, сгенерированную OpenVPN, через файлы состояния и журналы. Список файлов для загрузки отображается только при наличии файла для загрузки. Если такового нет, появится сообщение "Файл в контроллере не найден!" отображается. При нажатии на любую из ссылок запрошенный файл будет загружен через браузер.

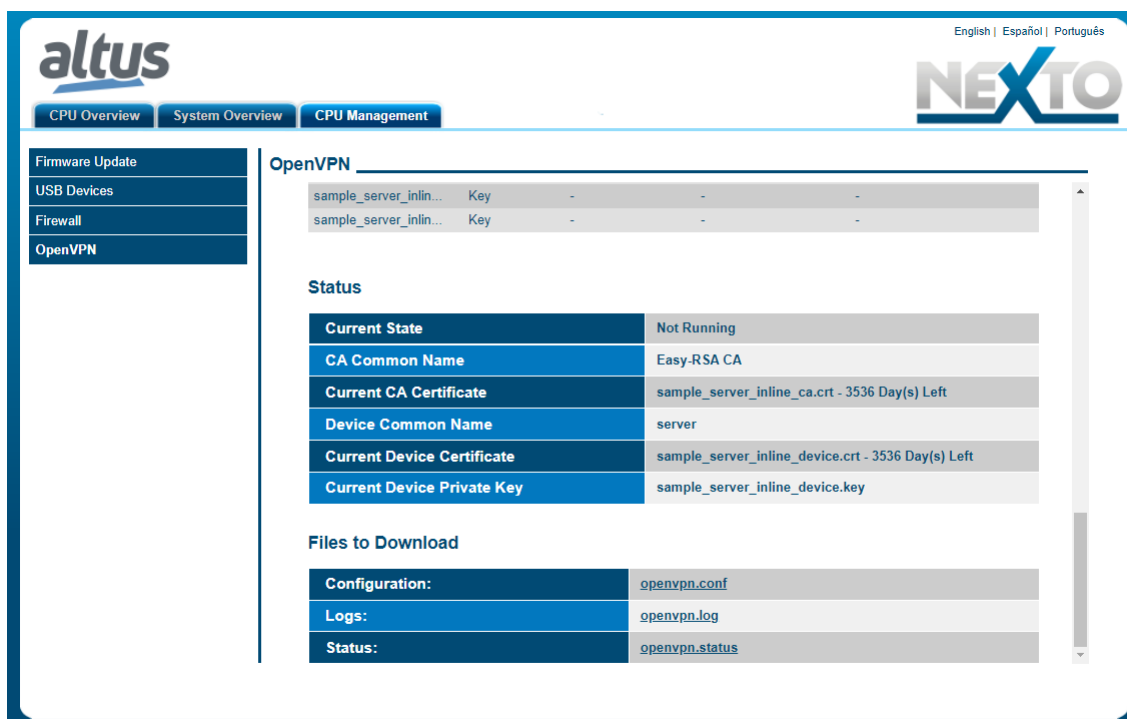


Рисунок 169: Раздел загрузки OpenVPN

### 5.18.6. Конфигурация архитектуры

В этом разделе будут рассмотрены некоторые возможные конфигурации для OpenVPN, такие как архитектуры Хост-хост, Хост-объект и Объект-объект.

### 5.18.6.1. Хост-хост



Рисунок 170: Пример архитектуры Хост-хост

Эта топология позволяет установить соединение между двумя хостами VPN. Оба хоста могут быть выбраны для использования в качестве сервера, затем другой должен быть настроен в качестве клиента, или оба хоста могут быть настроены как клиенты и иметь третий хост, который будет сервером для сети VPN.

Настройка этого типа архитектуры не требует какой-либо конкретной Конфигурации. Другими словами, нет никаких ограничений на настройки, доступные на веб-странице OpenVPN.

### 5.18.6.2. Хост объект

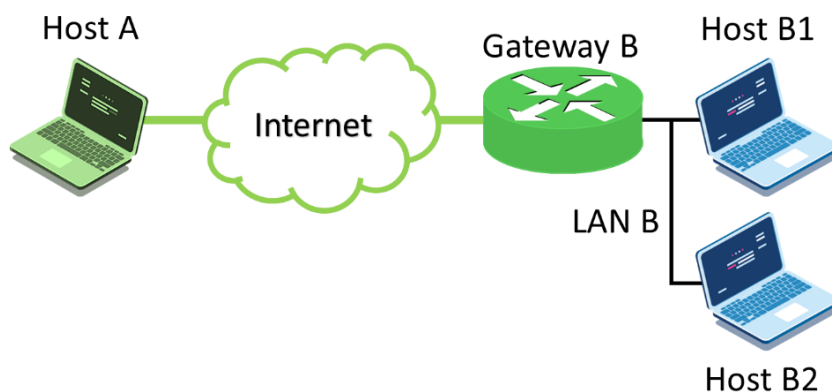


Рисунок 171: Пример архитектуры Хост - объект

Эта топология позволяет установить соединение между двумя хостами VPN, но один из этих хостов также выступает в качестве шлюза в сеть VPN. Через этот шлюз выполняется маршрутизация для установления связи между хостами A, B1, B2 и шлюзом B. В этом сценарии сервером может быть либо хост A, либо шлюз B. Когда один сервер в сети, другой будет клиентом.

Хосты B1 и B2, которые находятся в частной сети Lan B ниже шлюза B, не должны поддерживать OpenVPN, чтобы иметь возможность обмениваться данными, поскольку все коммуникации обрабатываются сетевым шлюзом VPN.

Чтобы обеспечить связь между всеми устройствами в сети, вам необходимо создать правила маршрутизации для VPN-туннеля. Пожалуйста, обратитесь к разделу Частные сети, чтобы узнать, как создавать правила частной сети.

Эта архитектура VPN-подключения требует определенных конфигураций. Сервер должен иметь свою топологию Конфигурация как подсеть, то есть Конфигурация по умолчанию контроллера, для настройки частных сетей под шлюзом B, как показано на изображении выше.

Вам также необходимо ввести адрес частной сети Lan B, которая будет обмениваться данными через VPN. Данная Конфигурация делается с помощью команды `push "route Lan_B_IP Lan_B's_Mask"` и требуется вне зависимости от того, находится частная сеть ниже клиента или OpenVPN сервера, но если частная сеть ниже VPN клиента, то необходимо добавить, в Помимо этой команды, следующая Конфигурация: `route route Lan_B_IP Lan_B's_Mask`. Эти настройки записываются в файл конфигурации VPN-сервера.

### 5.18.6.3. Объект-объект

Ниже приведен рисунок, представляющий соединение объект-объект:

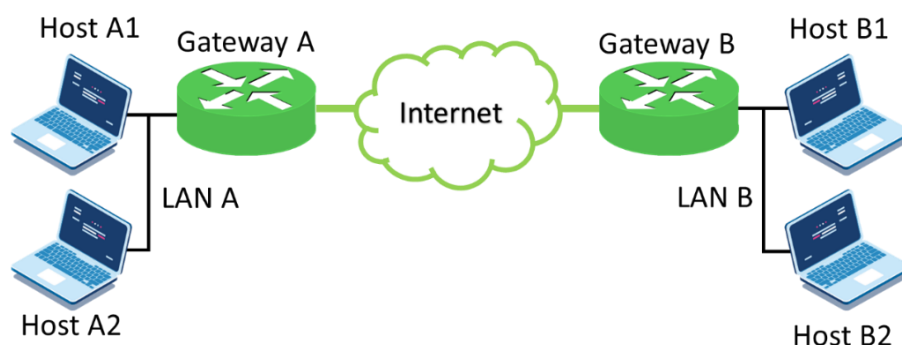


Рисунок 172: Пример архитектуры объект-объект

Эта топология позволяет установить соединение между двумя хостами VPN, каждый из которых действует как шлюз в сеть VPN. Через эти шлюзы предоставляется доступ для установления связи между хостами A1, A2, B1, B2, шлюзом А и шлюзом В. В этом сценарии любой шлюз может взять на себя роль сервера, поэтому другой будет клиентом.

Ни один из хостов, находящихся в частной сети ниже одного из двух шлюзов, не должен поддерживать OpenVPN, чтобы иметь возможность обмениваться данными, поскольку все коммуникации обрабатываются сетевыми шлюзами VPN.

Чтобы обеспечить связь между всеми устройствами в сети, вам необходимо создать правила маршрутизации для VPN-туннеля. Пожалуйста, обратитесь к разделу Частные сети, чтобы узнать, как создавать правила частной сети.

Конфигурации для этой архитектуры требуют тех же конкретных настроек, которые описаны в разделе Host-to-Site, с той разницей, что теперь есть две частные сети, и обе должны следовать конфигурации, которая была продемонстрирована. Предполагая, что шлюз А является сервером в этом соединении, вы должны добавить в файл конфигурации следующие команды: `push «route Lan_A_IP Lan_A's_Mask»`, `route Lan_B_IP Lan_B's_Mask` и `push «route Lan_B_IP Lan_B's_Mask»`. Если сервер Шлюз В, то в файле Конфигурация будет добавлено: `push «route Lan_B_IP Lan_B's_Mask»`, `route Lan_A_IP Lan_A's_Mask` и `push «route Lan_A_IP Lan_A's_Mask»`.

## 6. Обслуживание

Одной из особенностей серии Nexто является генерация диагностики аномалий, будь то сбои, ошибки или режимы работы, что позволяет оператору легко выявлять и решать проблемы, возникающие в системе.

ЦП Nexто позволяют визуализировать диагностические данные, генерируемые системой, различными способами, а именно:

Диагностика One Touch Diag через светодиоды    Диагностика через WEB    Диагностика через переменные  
Диагностика через функциональные блоки

Первая — это инновационная функция серии Nexто, которая обеспечивает быстрый доступ к нештатным условиям приложения. Второй — чисто визуальный, формируется двумя светодиодами, размещенными на панели (DG и WD), а также светодиодами, размещенными в разъеме RJ45 (исключительно для подключения Ethernet). Следующей особенностью является графическая визуализация на WEB-странице стойки и соответствующих конфигунокд модулей с возможностью индивидуального доступа к рабочему состоянию и активной диагностике. Диагностика также хранится непосредственно в переменных ЦП, либо в прямом представлении (%Q), либо в атрибутах (переменная АТ), и может использоваться пользовательским приложением, например, в системе диспетчерского управления. Последние представляют специфические условия функционирования системы.

Эта функция диагностики предназначена для выявления возможных проблем с установкой системы или конфигурацией, а также проблем или недостатков сети связи.

### 6.1. Диагностика модуля

#### 6.1.1. Диагностика одним касанием

One Touch Diag (OTD), или диагностика одним касанием, является эксклюзивной функцией серии Nexто для программируемых контроллеров. С помощью этой новой концепции пользователь может проверять диагностику любого модуля, подключенного к системе, прямо на графическом дисплее ЦП одним касанием диагностического переключателя модуля. Это мощный диагностический инструмент, который можно использовать в автономном режиме (без использования программного обеспечения для контроля или программирования), что упрощает поиск и быстрое решение возможных проблем.

Клавиша диагностики расположена на верхней части ЦП, в легкодоступном месте и, кроме предоставления активной диагностики, позволяет получить доступ к навигационному меню, описанному в разделе Конфигурация – Информационное и Конфигурационное меню ЦП.

На рисунке ниже показано расположение переключателя процессора:

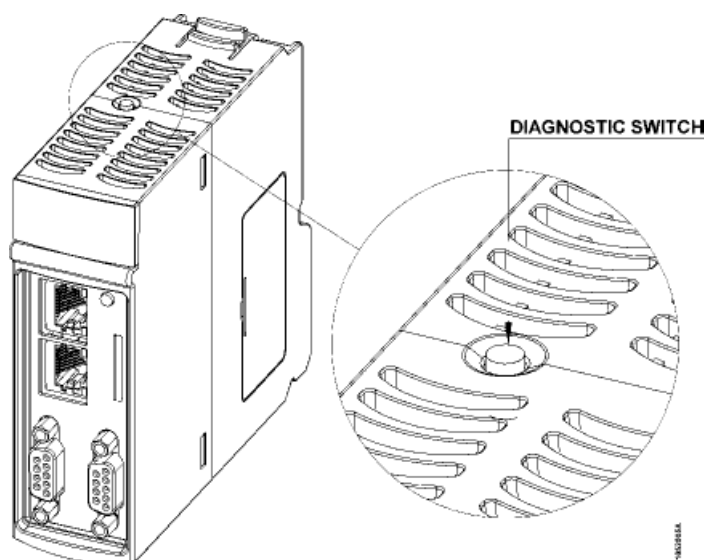


Рисунок 173: Диагностический переключатель

Одним кратким касанием ЦП начинает отображать диагностику шины (когда активен, в противном случае показывает сообщение «NO DIAG»). Первоначально тег визуализируется (конрисункод в свойствах модуля в программе MasterTool IEC XE, в соответствии со стандартом IEC 61131-3), другими словами, имя, присвоенное ЦП, и после этого отображаются все диагностические данные через дисплей ЦП. Сообщения. Этот процесс выполняется дважды на дисплее. Все происходит автоматически, так как пользователю нужно выполнить только первое короткое касание, а ЦП отвечает за отображение диагностики. Диагностика других модулей, присутствующих на шине, также отображается на графическом дисплее ЦП кратким нажатием кнопки диагностического модуля в той же

## 6. ОБСЛУЖИВАНИЕ

презентационной модели диагностики.

На приведенном ниже рисунке показан процесс, начинающийся с короткого касания, а условия и время ЦП представлены в прямоугольниках меньшего размера. Важно подчеркнуть, что диагностика может иметь более одного экрана, другими словами, указанное время на блок-схеме ниже действительно для одного из них.

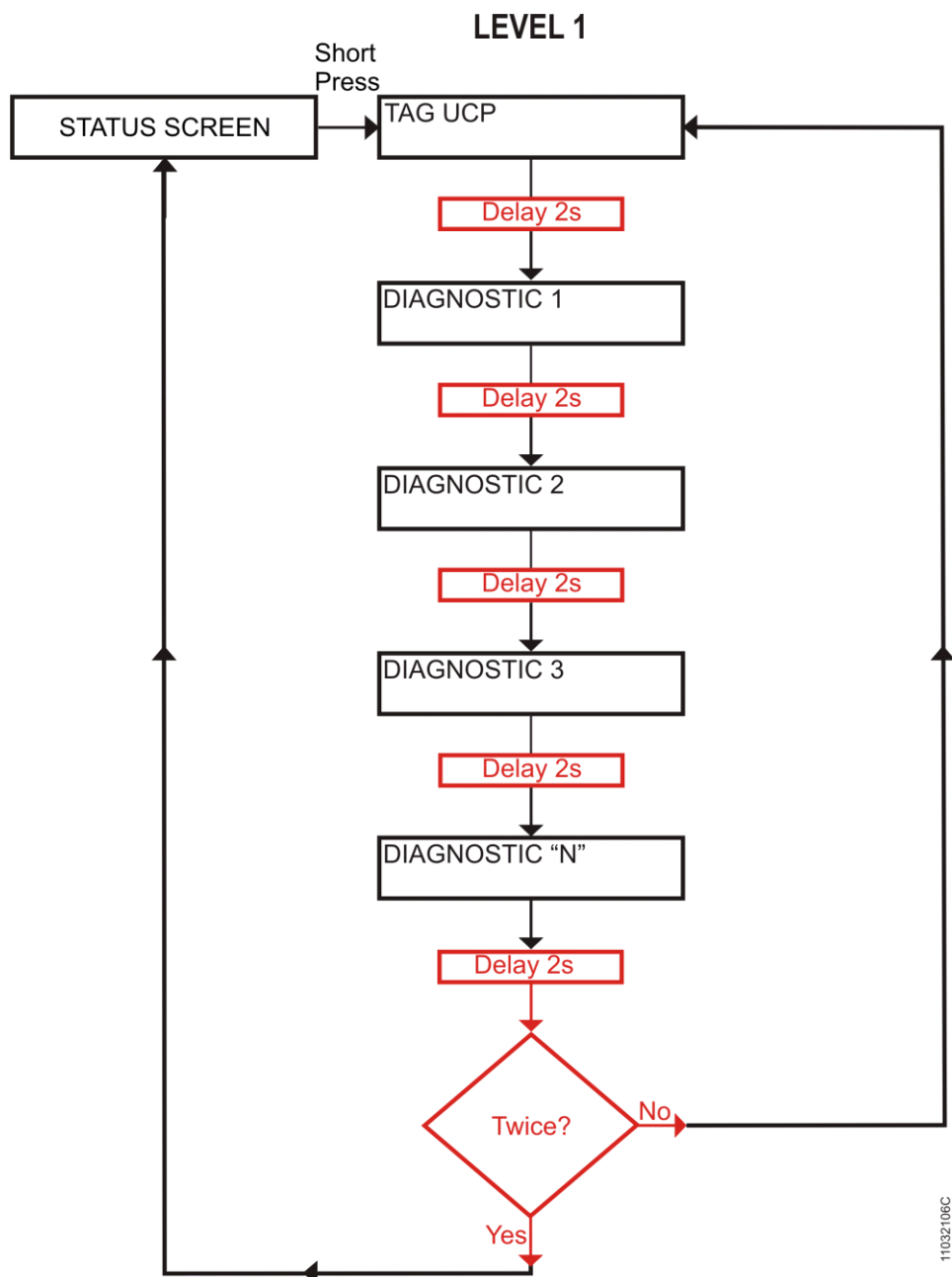


Рисунок 174: Визуализация диагностики ЦП

Перед завершением всего процесса визуализации достаточно коротко коснуться диагностического переключателя в любой момент или нажать диагностический переключатель с любого модуля ввода/вывода, подключенного к шине. Кроме того, важно отметить, что One Touch Diag может быть доступен, когда модуль находится в рабочем режиме.

В случае выполнения длинного касания ЦП переходит в меню навигации, которое описано в разделе Конфигурация – Информационное и Конфигурационное меню ЦП.

В приведенной ниже таблице показана разница между коротким временем касания, длительным временем касания и залипшей кнопкой.

Тип сенсора	Минимальное время	Максимальное время	Состояние индикации
Без касаний	-	59.99 мс	-
Короткое касание	60 мс	0.99 с	Выпуск
Длинное касание	1 с	20 с	Более 1 с до 20 с
Заблокированный переключатель	20.01 с	(∞)	Диагностические показания см. в Таблице 184.

Таблица 179: Время диагностики одним касанием (One Touch)

Сообщения, отображаемые на графическом дисплее Nexto CPU, соответствующие диагностике, описаны в разделе «Диагностика через переменные» в Таблице 184.

Если в одном из модулей ввода-вывода произойдет какая-либо ситуация с заеданием кнопки, диагностическая кнопка этого модуля остановится или отобразит диагностику на графическом дисплее ЦП при ее нажатии. В этом случае CPU укажет, что имеется модуль с активной диагностикой. Чтобы удалить эту диагностику из ЦП, необходимо выполнить горячую замену в модуле, где диагностика активна.

Дополнительные сведения о процедуре просмотра диагностики ЦП или других шинных модулей см. в описании в Руководстве пользователя серии Nexto — MU214600.

#### 6.1.2. Диагностика через светодиод

Изделие имеет светодиод индикации диагностики (светодиод DG) и светодиод индикации событий таймера контрольной системы (светодиод WD). Таблицы 180 и 181 показывают значение каждого состояния и его соответствующих описаний.

##### 6.1.2.1. DG (диагностический)

Зеленый	Красный	Описание	Причины	Приоритетность
<b>Выкл</b>	<b>Выкл</b>	Не используется	Нет источника питания. Аппаратная проблема	-
<b>Вкл</b>	<b>Выкл</b>	Все приложения в режиме выполнения (Выполнить)	-	3 (низкий)
<b>Выкл</b>	<b>Вкл</b>	Все приложения в режиме остановки (Stop)	-	3 (низкий)
<b>мигающий 2х</b>	<b>Выкл</b>	Шинные модули с диагностикой	По крайней мере, шинный модуль, включая ЦП, имеет активную диагностику.	1
<b>мигающий 3х</b>	<b>Выкл</b>	Форсирование данных	Некоторая область памяти форсируется пользователем через Master-Tool IEC XE	2
<b>Выкл</b>	<b>мигающий 4х</b>	Конфигурация или аппаратная ошибка в шине	Шина повреждена или неправильно настроена	0 (высокий)

Таблица 180: Описание состояний диагностических светодиодов

## 6.1.2.2. WD (Таймер контрольной системы)

Красный экран	Описание	Причины	Приоритетность
Выкл	Не горит таймер контрольной системы	Нормальная работа	3 (низкий)
мигает 1х	Программный таймер контрольной системы	Пользовательское приложение таймер контрольной системы	2
вкл	Аппаратный таймер контрольной системы	Поврежденный модуль и/или поврежденная операционная система	1 (высокий)

Таблица 181: Описание состояния светодиодов Таймер контрольной системы

**Примечания:**

**Программное обеспечение Таймер контрольной системы:** Для того, чтобы убрать индикацию таймера контрольной системы, выполните сброс приложения или выключите и снова включите ЦП. Это происходит, когда время выполнения пользовательского приложения превышает время, установленное таймером контрольной системы.

Диагностику можно проверить в переменной Exception.wExceptionCode, см. Таблицу 188.

**Аппаратный Таймер контрольной системы:** Для сброса любой индикации таймера контрольной системы, как в светодиоде WD, так и в операнде Reset.bТаймер контрольной системыReset, модуль должен быть отключен от источника питания.

Чтобы проверить условия приложения при перезапуске модуля, см. Конфигурации в Таблице 37.

**6.1.2.3. Индикаторы разъема RJ45**

Оба светодиода, размещенные в разъемах RJ45, помогают пользователю в обнаружении проблем с установленной физической сетью, указывая скорость сетевого соединения и наличие трафика связи интерфейса. Значение светодиодов представлено в Таблице ниже.

Желтый	Зеленый	Описание
◦	◦	Сетевая ссылка отсутствует
◦	•	Сеть 1000 Мбайт/с ССЫЛКА
•	◦	Сеть 100 Мбайт/с ССЫЛКА
•	•	10 Мбайт/с сеть ССЫЛКА
X	-	Возникновение передачи или приема по сети Ethernet для этого IP-адреса или на него. Мигает по запросу процессора Nexto, а не при каждой передаче или приеме, другими словами, может мигать на более низкой частоте, чем реальная частота передачи или приема.

Таблица 182: Индикаторы значения Ethernet LEDs

**6.1.3. Диагностики через ВЕБ**

Помимо ранее представленных функций, серия Nexto предоставляет пользователю инновационный инструмент доступа к диагностике системы и рабочим состояниям через веб-страницу.

Использование, помимо того, что является динамичным, очень интуитивно понятно и облегчает работу пользователя. Использование системы контроля можно заменить, если оно ограничено проверкой состояния системы.

Чтобы получить доступ к нужной веб-странице ЦП, достаточно использовать стандартный браузер (Internet Explorer 7 или выше, Mozilla Firefox

3.0 или более поздней версии и Google Chrome 8 или более поздней версии) и введите в адресной строке IP-адрес ЦП (пример: <http://192.168.1.1>). Сначала представлена информация о процессоре, как показано на рисунке 175:

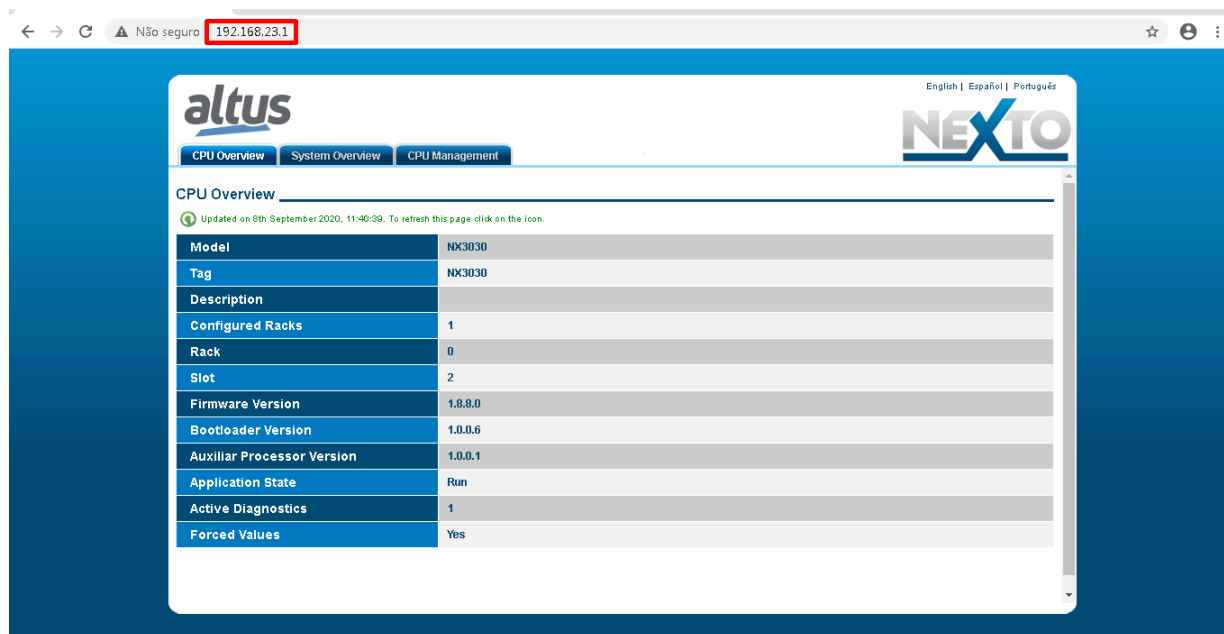


Рисунок 175: Начальный экран

Существует также вкладка «Обзор системы», которую можно просмотреть в стойке или в списке имеющихся модулей (опция в правой части экрана). Пока на ЦП нет приложения, на этой странице будет отображаться Конфигурация с самой большой доступной стойкой и стандартным блоком питания, подключенным к ЦП. При использовании визуализации Rack модули с диагностикой мигают и окрашиваются в красный цвет, как показано на Рисунок 176. В противном случае отображается список с подключенными к системе модулями, тегами и номером активной диагностики:

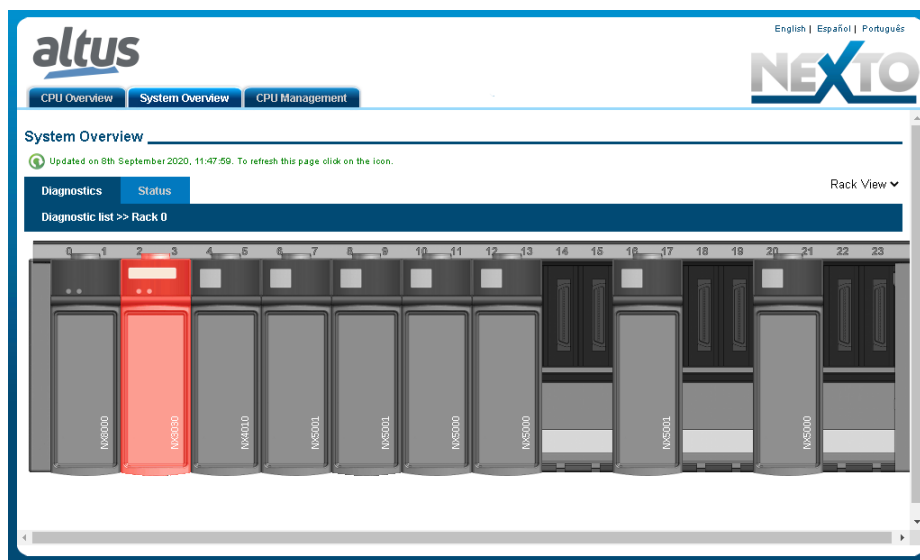


Рисунок 176: Системная информация

Когда модуль с диагностикой нажат, отображается активный(е) модуль(и) диагностики, как показано на Рисунке 177:

**ВНИМАНИЕ**

Когда ЦП перезагружается и приложение попадает в исключение при запуске системы, диагностика недействительна. Необходимо исправить проблему, которая генерирует исключение приложения, чтобы диагностика обновлялась.

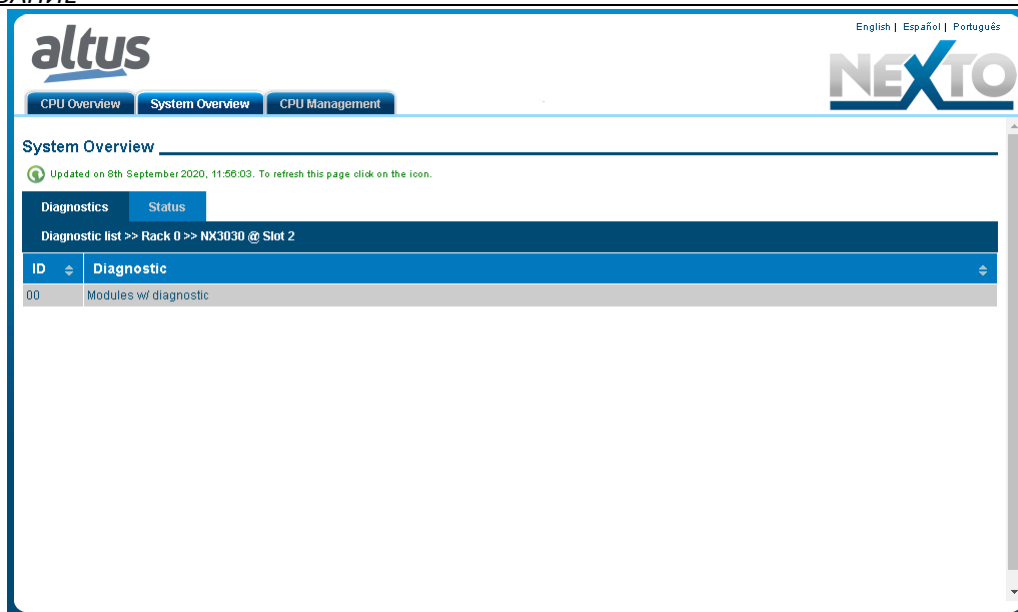


Рисунок 177: Диагностика системы

Если выбрана вкладка «Состояние», на экране отображается состояние всей подробной диагностики, как показано на Рисунке 178:

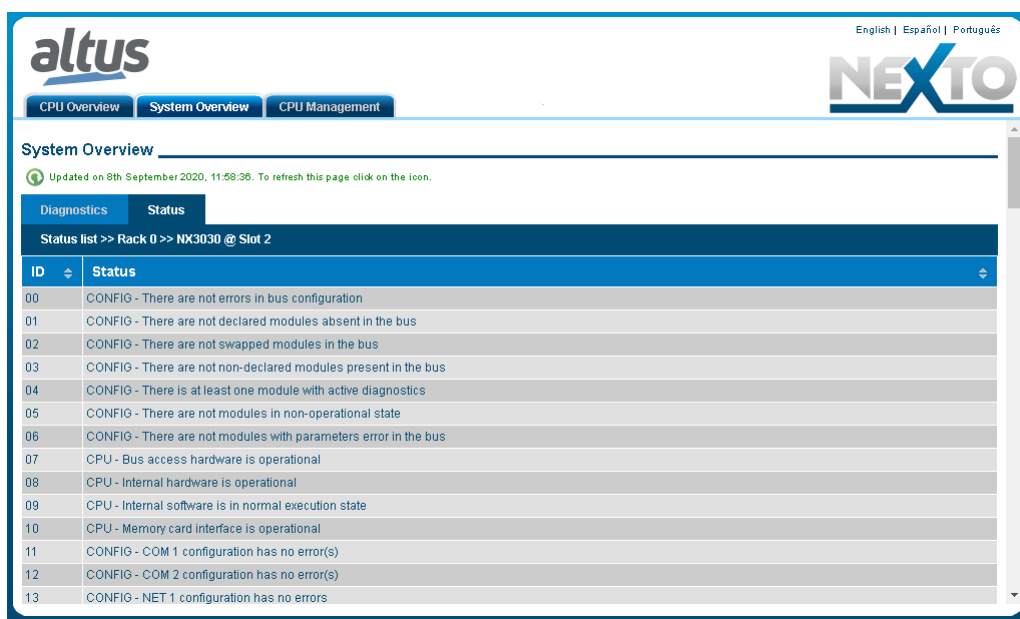


Рисунок 178: Статус системы

Пользователь может выбрать для визуализации два варианта языка: португальский и английский. Просто измените в правой верхней части экрана нужный язык.

#### 6.1.4. Обзорщик диагностики

Diagnostic Explorer — это включение диагностики через WEB в MasterTool IEC XE, чтобы сделать процесс более быстрым и прямым.

Доступ к этой функции происходит двумя способами:

- Доступ к опции «Diagnostic Explorer» в дереве устройств, размещенной на экране MasterTool IEC XE слева, и указание правильного IP-адреса в поле, указанном на Рисунок 179. Помните, что для отображения страницы диагностики пользователь должен быть подключен к ЦП (Начальное программирование – раздел входа в систему).

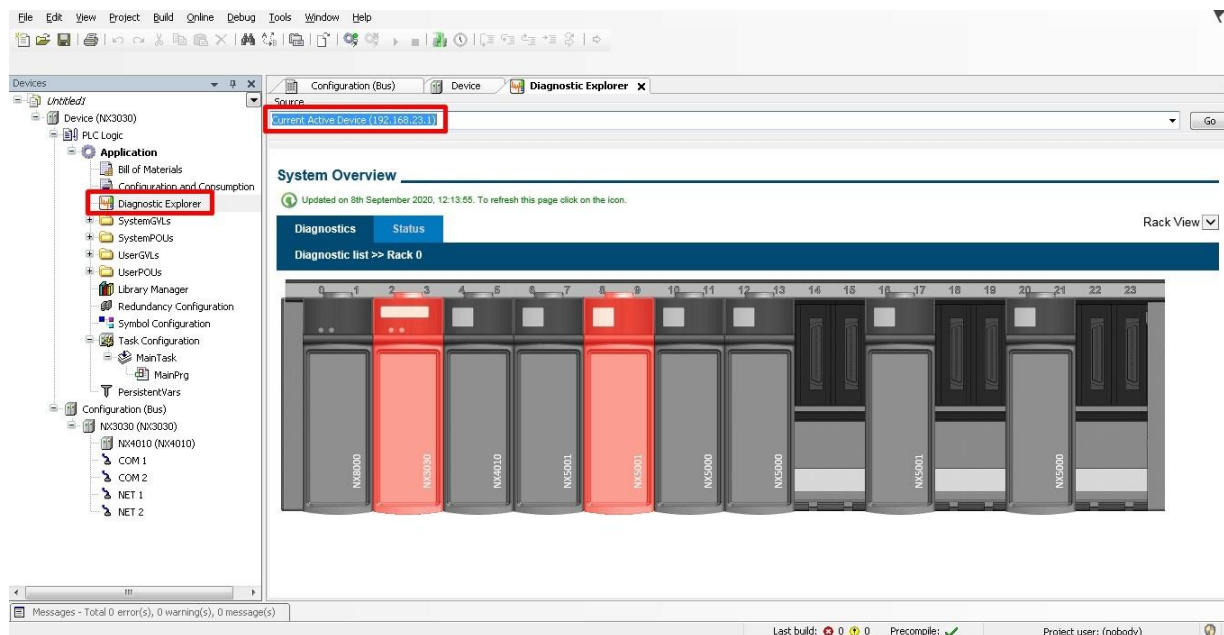


Рисунок 179: Экран проводника диагностики

- Щелкнув правой кнопкой мыши по модулю и выбрав «Диагностика», откроется Проводник диагностики, направляющий на страницу состояния модуля.

#### 6.1.5. Диагностика через переменные

ЦП серии Nexто имеют множество переменных для диагностической индикации. Имеются структуры данных с диагностикой всех модулей, объявленных на шине, отображенных на переменные прямого представления %Q и определенные символически через директиву AT, в GVL System\_Diagnostics, созданные автоматически MasterTool IEC XE.

Таблица ниже резюмирует разделение диагностических байтов/слов:

Byte	Описание
0 to 3	Сводная диагностика процессора.
4 to 1686	Детальная диагностика процессора.

Таблица 183: Отдел диагностики ЦП

##### 6.1.5.1. Обобщенная диагностика

В приведенной ниже таблице показано значение каждого суммарного диагностического бита ЦП:

Прямая переменная	Сообщение по диагностике	AT DG_Module.tSummarized.*	Variable	Описание
Переменная	Бит			
-	-	NO DIAG	-	Активной диагностики нет.
	0	CONFIG. MISMATCH	bConfigMismatch	TRUE – В шине возникла проблема Конфигурация, так как модуль вставлен не в то положение. FALSE – Шина настроена правильно.
	1	ABSENT MODULES	bAbsentModules	TRUE — один или несколько объявленных модулей отсутствуют. FALSE — в шине обнаружены все заявленные модули.

## 6. ОБСЛУЖИВАНИЕ

%QB(n)	2	SWAPPED MODULES	bSwappedModules	TRUE – в шине есть измененные модули.
				FALSE – в шине нет измененных модулей.
	3	NON-DECLARED MODULES	bNonDeclaredModules	TRUE — один или несколько модулей в шине не были объявлены в Конфигурации.
				FALSE — объявлены все модули.
	4	MODULES W/ DIAGNOSTICS	bModulesWithDiagnostic	TRUE – один или несколько модулей на шине имеют активную диагностику.
				FALSE – в модулях на шине нет активной диагностики.
	5	MODULES W/ FATAL ERROR	bModuleFatalError	TRUE — один или несколько модулей на шине находятся в фатальной ошибке.
				FALSE – Все модули работают правильно.
%QB(n+1)	6	MODULES W/ PARAM. ERROR	bModuleParameterError	TRUE – Один или несколько модулей на шине имеют ошибку параметризации.
				FALSE – Все модули параметризованы.
	7	BUS ERROR	bWHSBBusError	ИСТИНА – Индикация ведущего устройства о сбое в шине WHSB.
				FALSE – Шина WHSB работает нормально.
	0	HARDWARE FAILURE	bHardwareFailure	TRUE — аппаратный сбой процессора.
				FALSE — оборудование работает нормально.
	1	SOFTWARE EXCEPTION	bSoftwareException	TRUE — одно или несколько исключений, созданных программным обеспечением.
				FALSE — исключения не генерируются в программном обеспечении.
	3	ERROR IN MEMORY CARD	bMemoryCardError	TRUE – карта памяти вставлена в ЦП, но не работает должным образом.
				FALSE — карта памяти работает правильно.

Прямая переменная		Сообщение по диагностике	Variable DG_Module.tSummarized. *	Описание
Переменная	Бит			
	4	COM1 CONF. ERROR	bCOM1ConfigError	TRUE — произошла ошибка во время или после настройки последовательного интерфейса COM 1.
				FALSE — конфигурация последовательного интерфейса COM 1 верна.
	6	NET1 CONF. ERROR	bNET1ConfigError	TRUE — произошла ошибка во время или после настройки интерфейса NET 1 Ethernet.
				FALSE — Конфигурация интерфейса NET 1 Ethernet верна.
	7	NET2 CONF. ERROR	bNET2ConfigError	TRUE — произошла ошибка во время или после настройки интерфейса NET 2 Ethernet.
				FALSE — Конфигурация интерфейса NET 2 Ethernet верна.
%QB(n+2)	0	NET3 CONF. ERROR	bNET3ConfigError	TRUE — произошла ошибка во время или после настройки интерфейса NET 3 Ethernet.
				FALSE — Конфигурация интерфейса NET 3 Ethernet верна.
	1	INVALID DATE/TIME	bInvalidDateTime	TRUE – дата или час недействительны.
				FALSE – дата и час указаны правильно.
	2	RUNTIME RESET	bRTSReset	TRUE — RTS (система выполнения) была перезапущена хотя бы один раз. Эта диагностика сбрасывается только при перезапуске системы.
				FALSE — RTS (система выполнения) работает нормально.
	3	RETENTIVITY LOST	bRetentivityLost	TRUE — при сохранении сохраняемых данных произошла ошибка.
				FALSE — сохраняемые данные успешно сохраняются.
	4	OTD SWITCH ERROR	bOTDSwitchError	TRUE – Истинно, если ключ OTD был заблокирован более чем на 20 с хотя бы один раз при подаче питания на ЦП. Эта диагностика сбрасывается только при перезапуске системы.
				FALSE – Ключ в настоящее время не заблокирован или был заблокирован, когда ЦП был включен.
	5	WRONG CPU SLOT	bWrongCPUSlot	TRUE - ЦП подключен к слоту, отличному от заявленного в пользовательском приложении.
				FALSE — ЦП правильно подключен к слоту, указанному в пользовательском приложении.
	6	USB PORT ERROR	bUSBDiagnostic	TRUE - USB имеет некоторую диагностику.
				FALSE - USB без проблем.
	0	ABSENT RACK	bAbsentRacks	TRUE — одна или несколько заявленных стоек отсутствуют.

## 6. ОБСЛУЖИВАНИЕ

				FALSE – Отсутствующих стоек нет.
	1	DUPLICATE D RACK	bDuplicatedRacks	TRUE – имеются стойки с повторяющимся идентификационным номером.

Прямая переменная		Сообщение по диагностике	Variable DG_Module.tSummarized. *	Описание
Переменная	Bit			
%QB(n+3)				FALSE – Нет стоек с повторяющимся идентификационным номером.
	2	INVALID RACK	bInvalidRacks	TRUE – имеются стойки с недопустимым идентификационным номером.
				FALSE – Нет стоек с допустимым идентификационным номером.
	3	NON DECLARED RACK	bNonDeclaredRacks	TRUE – имеются стойки с незадекларированным идентификационным номером.
				FALSE – Нет стоек с незадекларированным идентификационным номером.
	4	DUPLICATE D SLOT	bDuplicatedSlots	TRUE – есть несколько повторяющихся адресов слотов.
				FALSE – нет повторяющихся адресов слотов.

Таблица 184: Сводная диагностика ЦП

**Примечания:**

**Переменная прямого представления:** «n» представляет собой значение, установленное в ЦП с помощью программного обеспечения MasterTool IEC XE, например, при диагностике исходного адреса.

Директива AT: В описании символических переменных, которые используют директиву AT для выполнения отображения в переменных с прямой адресацией, синтаксис, который должен быть помещен перед желаемой сводной диагностикой, равен DG\_Module.tSummarized, когда слово модуля заменяется используемым ЦП. Например, для диагностики несовместимой Конфигурации необходимо использовать переменную: DG\_NX3010.tSummarized.bConfigMismatch. Директива AT — это слово, зарезервированное в программном обеспечении для программирования и используемое только для диагностической индикации.

Конфигурация Несоответствие: Диагностика несовместимая Конфигурация формируется, если один или несколько модулей стойки не соответствуют заявленным, т.е. при отсутствии или разных состояниях модулей. Вставленные в шину модули, не заявленные в проекте, не учитываются.

Переставленные модули: Если только два модуля заменены между собой на шине, то можно идентифицировать измененную диагностику. В противном случае проблема трактуется как «Несоответствие конфигурации».

Модули с неустранимой ошибкой: В случае, если диагностика модулей с неустранимой ошибкой верна, необходимо проверить, какой модуль в шине является проблемным, и отправить его в службу технической поддержки Altus, так как он имеет аппаратный сбой.

Модуль с ошибкой параметризации: Если диагностика ошибки параметризации верна, необходимо убедиться, что модуль на шине сконфигурирован правильно, а версия микропрограммы и программного обеспечения MasterTool IEC XE верна. Если проблема возникла при вставке модуля в шину, убедитесь, что модуль поддерживает горячую замену.

Ошибка шины: Считается фатальной ошибкой, прерывающей доступ к модулям в шине. Если диагностика ошибки шины верна, возможно, возникла аппаратная проблема на линиях связи шины, затем обратитесь в службу технической поддержки Altus.

Аппаратный сбой: если диагностика аппаратного сбоя верна, ЦП необходимо отправить в службу технической поддержки Altus, так как у него есть проблемы с RTC, вспомогательным процессором или другими аппаратными ресурсами.

Программное исключение: если диагностика программного исключения верна, пользователь должен проверить свое приложение, чтобы гарантировать, что оно не обращается к памяти неправильно. Если проблема остается, необходимо обратиться в сектор технической поддержки Altus. Коды исключений программного обеспечения описаны далее в таблице подробной диагностики ЦП.

Диагностическое сообщение: диагностические сообщения можно просмотреть на графическом дисплее ЦП с помощью клавиши OTD или с помощью веб-интерфейса на странице диагностики ЦП.

**6.1.5.2. Детальная диагностика**

Таблицы ниже содержат подробную диагностику процессоров серии Nexto. Важно иметь в виду следующие наблюдения, прежде чем обращаться к ним:

Визуализация диагностических структур: Диагностические структуры, добавленные в проект, можно увидеть в элементе «Диспетчер библиотек» древовидного представления MasterTool IEC XE. Там можно увидеть все типы

## 6. ОБСЛУЖИВАНИЕ

данных, определенные в структуре. Счетчики: все счетчики диагностики ЦП возвращаются к нулю при превышении их предельного значения.

Переменная прямого представления: «n» представляет собой значение, настроенное в ЦП через MasterTool IEC XE в качестве начального диагностического адреса.

Директива AT: при описании символических переменных, которые используют директиву AT для отображения в переменные прямого отображения, синтаксис, который следует использовать перед желаемой сводной диагностикой, выглядит следующим образом: DG\_Module.tDetailed., где слово Module должно быть заменено используемым ЦП. Директива AT является зарезервированным словом программиста, и некоторые символьные переменные, использующие эту директиву, указывают на диагностику.

Прямое представление	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QD(n+4)	DWORD	Target. dwCPUModel	NX3003 = 0x3003 NX3004 = 0x3004 NX3005 = 0x3005 NX3008 = 0x3008 NX3010 = 0x3010 NX3020 = 0x3020 NX3030 = 0x3030
%QB(n+8)	BYTE ARRAY(4)	Target. abyCPUVersion	Версия прошивки.
%QB(n+12)	BYTE ARRAY(4)	Target. abyBootloaderVersion	Версия загрузчика.
%QB(n+16)	BYTE ARRAY(4)	Target. abyAuxprocVersion	Версия вспомогательного процессора.
%QB(n+20)	BYTE ARRAY(4)	Target. abyPowprocVersion	Версия мощного процессора.

Таблица 185: Целевая группа детальной диагностики

Прямое представление	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QX(n+24).1	BIT	Hardware. bRTCFailur e	Основной процессор не может обмениваться данными с RTC (часами ЦП).
%QX(n+24).2	BIT	Hardware. bThermometerFailure	Сбой связи между термометром и главным процессором.

Таблица 186: Подробная диагностика оборудования Описание группы

Прямое представление	Размер	AT Variable DG_Module.tDetailed.*	Описание
%QW(n+25)	WORD	Exception. wExceptionCode	Exception code generated by the RTS. See Table 188.
%QB(n+27)	BYTE	Exception. byProcessorLoad	Level, in percentage (%), of charge in the processor.

Таблица 187: Группа подробной диагностики исключений Описание

**Примечание:**

**Код исключения:** код исключения, сгенерированный RTS (Runtime System), можно посмотреть ниже:

Код	Описание	Код	Описание
0x0000	Кода исключения нет.	0x0051	Нарушение доступа.
0x0010	Таймер контрольной системы истекло время выполнения задания МЭК (ПО Таймер контрольной системы).	0x0052	Привилегированная инструкция.
0x0012	Конфигурация ошибки ввода/вывода.	0x0053	Сбой страницы.
0x0013	Ошибка контрольной суммы после загрузки программы.	0x0054	Переполнение стека.

Код	Описание	Код	Описание
0x0014	Ошибка полевой шины.	0x0055	Неверное расположение.
0x0015	Ошибка обновления ввода/вывода.	0x0056	Неверный маневр.
0x0016	Время цикла (выполнение) превышено.	0x0057	Защищенная страница.
0x0017	Онлайн-обновление программы слишком долгое.	0x0058	Двойной провал.
0x0018	Внешние ссылки не разрешены.	0x0059	Неверный опкод.
0x0019	Загрузка отклонена.	0x0100	Несоответствие типа данных.
0x001A	Проект не загружен, так как сохраняемые переменные не могут быть перераспределены.	0x0101	Превышен лимит массивов.
0x001B	Проект не загружен и удален.	0x0102	Деление на ноль.
0x001C	Недостаточно памяти.	0x0103	Переполнение.
0x001D	Ретенционная память повреждена и не может быть сопоставлена.	0x0104	Нельзя продолжать.
0x001E	Проект может быть загружен, но позже вызывает падение.	0x0105	Таймер контрольной системы в загрузке процессора всех обнаруженных задач IEC.
0x0021	Цель запускаемого приложения не соответствует текущей цели.	0x0150	FPU: не указана ошибка.
0x0022	Ошибка запланированных задач. Скачанный файл Checksum с ошибкой.	0x0151	FPU: операнд не является нормальным.
		0x0152	FPU: деление на ноль.
0x0023	Сохраняемое удостоверение не соответствует текущему удостоверению программы загрузочного проекта IEC. Ошибка конфигурации задачи.	0x0153	FPU: Неточный результат.
0x0024	Приложение работает с неправильной целью.	0x0154	FPU: неверная операция.
0x0025	Незаконная инструкция.	0x0155	FPU: переполнение.
0x0026	Ошибка полевой шины.	0x0156	FPU: проверка стека.
0x0050	Ошибка обновления ввода/вывода.	0x0157	FPU: недолив.

Таблица 188: RTS Коды исключений

Прямое представление	Размер	AT DG_Module.tDetailed.* Variable	Описание
%QB(n+28)	BYTE	WebVisualization. byConnectedClients	Количество клиентов, подключенных к веб-визуализации.

Таблица 189: Описание группы подробной диагностики веб-визуализации

Прямое представление	Размер	AT DG_Variable_Module.tDetailed.*	Описание
%QB(n+29)	BYTE	RetainInfo. byCPUInitStatus	Статус запуска процессора: 01: Горячий старт 02: Теплый старт 03: Холодный пуск Примечание. Эти переменные сбрасываются при каждом включении питания.
%QW(n+30)	WORD	RetainInfo. wCPUColdStartCounter	Счетчик холодного запуска: он будет увеличиваться только из-за горячего удаления ЦП на шине, а не из-за команды холодного сброса MasterTool IEC XE. (от 0 до 65535)

## 6. ОБСЛУЖИВАНИЕ

Прямое представление	Размер	AT DG Variable Module.tDetailed.*	Описание
%QW(n+32)	WORD	RetainInfo. wCPUWarmStartCounter	Счетчик теплого запуска: он будет увеличиваться только во время последовательности включений питания системы, а не из-за команды Hot Reset MasterTool IEC XE. (от 0 до 65535).
%QW(n+36)	WORD	RetainInfo. wRTSResetCounter	Счетчик сбросов, выполненных RTS (Runtime System). (от 0 до 65535).

Таблица 190: Подробная диагностика группы RetainInfo

Прямое представление	Размер	AT DG Module.tDetailed.* Variable	Описание
%QX(n+40).0	BIT	Reset. bBrownOutReset	ЦП был перезапущен из-за отказа источника питания при последнем запуске.
%QX(n+40).1	BIT	Reset. bТаймер контрольной системыReset	Центральный процессор был перезапущен из-за активного таймера контрольной системы при последнем запуске.

Таблица 191: Описание группы подробной диагностики сброса

### Примечание:

**Сброс пониженного напряжения:** Диагностика сброса пониженного напряжения верна только тогда, когда источник питания превышает минимальный предел, требуемый его техническими характеристиками, оставаясь низковольтным, т. е. не подвергаясь прерыванию. ЦП идентифицирует падение напряжения питания и выдаст диагностическое сообщение о сбое питания. Когда напряжение восстанавливается, ЦП автоматически перезагружается и отображает диагностическое сообщение сброса пониженного напряжения.

Прямое представление	Размер	AT DG Module.tDetailed.* Variable	Описание
%QX(n+41).0	BIT	Thermometer. bOverTemperatureAlarm	Аварийный сигнал подается из-за внутренней температуры 85 °C или выше.
%QX(n+41).1	BIT	Thermometer. bUnderTemperatureAlarm	Аварийный сигнал генерируется из-за внутренней температуры 0° C или ниже.
%QD(n+42)	DINT	Thermometer. diTemperature	Температура, считываемая внутренним датчиком ЦП.

Таблица 192: Описание группы детальной диагностики термометра

### Примечание:

**Температура:** Чтобы увидеть температуру непосредственно в адресе памяти, необходимо сделать преобразование, так как размер данных DINT и мониторинг ведется в 4 байта. Поэтому рекомендуется использовать связанную символическую переменную, так как она уже обеспечивает конечное значение температуры.

Прямое представление	Размер	AT DG Module.tDetailed.* Variable	Описание
%QD(n+47)	DWORD	Serial.COM1. dwRXBytes	Счетчик символов, полученных от COM 1 (0 а 4294967295).
%QD(n+51)	DWORD	Serial.COM1. dwTXBytes	Счетчик символов, переданных с COM 1 (0 а 4294967295).
%QW(n+55)	WORD	Serial.COM1. wRXPendingBytes	Количество символов, оставшихся в буфере чтения в COM 1 (от 0 до 1024).
%QW(n+57)	WORD	Serial.COM1. wTXPendingBytes	Количество символов, оставшихся в буфере передачи в COM 1 (от 0 до 1024).
%QW(n+59)	WORD	Serial.COM1. wBreakErrorCounter	Передачик удерживает линию данных на нуле слишком долго, в соответствии с настроенным битом данных.

Прямое представление	Размер	AT DG_Modulo.tDetailed.* Variable	Описание
%QW(n+61)	WORD	Serial.COM1. wParityErrorCounter	Принятый кадр имеет несовпадающий бит четности.
%QW(n+63)	WORD	Serial.COM1. wFrameErrorCounter	Полученный кадр имеет неправильную начальную точку, обычно вызванную шумом или несоответствием скорости передачи данных.
%QW(n+65)	WORD	Serial.COM1. wRXOverrunCounter	Когда приемный кольцевой буфер заполнен и начинает терять старые кадры (слишком много кадров, не обработанных устройством).

Таблица 193: Описание группы подробной диагностики Serial COM 1

**Примечание:**

**Счетчик ошибок четности:** Когда последовательный COM 1 настроен без четности, этот счетчик ошибок не будет увеличиваться при получении сообщения с другой четностью. В этом случае будет указана ошибка кадра.

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QB(n+97).0	BIT	CAN. bBusAlarm	Автобус имеет критическую ошибку и выключен.
%QB(n+98)	BYTE	CAN. byBusState	Сообщает о состоянии устройства.
%QD(n+99)	UDINT	CAN. udiTxCounter	Количество измененных пакетов (Tx) на шине CAN ЦП.
%QD(n+103)	UDINT	CAN. udiRxCounter	Количество измененных пакетов (Rx) на шине CAN ЦП.
%QD(n+107)	UDINT	CAN. udiTxErrorCounter	Количество пакетов (Tx) с ошибками на шине CAN ЦП.
%QD(n+111)	UDINT	CAN. udiRxErrorCounter	Количество пакетов (Rx) с ошибками на шине CAN ЦП.
%QD(n+115)	UDINT	CAN. udiLostCounter	Количество пакетов, потерянных на шине CAN ЦП.

Таблица 194: Детальная диагностика CAN Group

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QB(n+135)	BYTE	USB. byUSBDevice	Тип устройства, подключенного к порту USB.
%QX(n+136).0	BIT	USB. bOvercurrent	Устройство, подключенное к порту USB, потребляет больше тока, чем поддерживается.
%QB(n+137)	BYTE	USB.tMassStorage. byMountState	Сообщает о состоянии устройства.
%QD(n+138)	DWORD	USB.tMassStorage. dwFreeSpaceKb	Сообщает о свободном месте на запоминающем устройстве.
%QD(n+142)	DWORD	USB.tMassStorage. dwTotalSizeKb	Сообщает общий размер запоминающего устройства.
%QD(n+148)	DWORD	USB.tSerialConverter. dwRXBytes	Счетчик принятых символов на COM 10 (от 0 до 4294967295).
%QD(n+152)	DWORD	USB.tSerialConverter. dwTXBytes	Счетчик символов, переданных COM 10 (от 0 до 4294967295).
%QW(n+156)	WORD	USB.tSerialConverter. wRXPendingBytes	Количество символов, оставшихся в буфере чтения на COM 10 (от 0 до 4095).
%QW(n+158)	WORD	USB.tSerialConverter. wTXPendingBytes	Количество символов, оставшихся в буфере передачи на COM 10 (от 0 до 1023).

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QW(n+160)	WORD	USB.tSerialConverter. wBreakErrorCounter	Передатчик удерживает линию данных на нуле слишком долго, в соответствии с настроенным битом данных.
%QW(n+162)	WORD	USB.tSerialConverter. wParityErrorCounter	Принятый кадр имеет несовпадающий бит четности.
%QW(n+164)	WORD	USB.tSerialConverter. wFrameErrorCounter	Полученный кадр имеет неправильную начальную точку, обычно вызванную шумом или несоответствием скорости передачи данных.
%QW(n+166)	WORD	USB.tSerialConverter. wRXOverrunCounter	Когда приемный кольцевой буфер заполнен и начинает терять старые кадры (слишком много кадров, не обработанных устройством).
%QX(n+173).0	BOOL	USB.tModem. bConfigured	Указывает, что модем был обнаружен на веб-странице.
%QB(n+174)	BYTE	USB.tModem. byConnectionState	Статус подключения модема.
%QB(n+175)	STRING (15)	USB.tModem. szIP	Строка с IP-адресом, используемым модемом.
%QX(n+282).0	BOOL	USB.tWifiAdapter. bConfigured	Указывает, что адаптер WiFi настроен на веб-странице.
%QB(n+283)	BYTE	USB.tWifiAdapter. byConnectionState	Статус подключения WiFi адаптера.
%QB(n+284)	STRING (15)	USB.tWifiAdapter. szIP	Строка с IP-адресом, используемым WiFi.
%QB(n+300)	STRING (15)	USB.tWifiAdapter. szMask	Строка с маской, используемой WiFi.
%QB(n+316)	STRING (15)	USB.tWifiAdapter. szGateway	Строка со шлюзом, используемым WiFi.
%QB(n+332)	STRING (17)	USB.tWifiAdapter. szMAC	Строка с MAC-адресом WiFi-адаптера (уникальным для устройства).

Таблица 195: Детальная диагностика USB Group

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QX(n+390+([x]-1)*139).0	BIT	Ethernet.NET[x]. bLinkDown	Указывает состояние канала в NET[x].
%QB(n+391+([x]-1)*139)	BYTE	Ethernet.NET[x]. byOperatingMode	Указывает режим работы интерфейса NET[x].
%QB(n+392+([x]-1)*139)	BYTE	Ethernet.NET[x]. byOperatingState	Указывает рабочее состояние интерфейса NET[x].
%QB(n+393+([x]-1)*139)	STRING (15)	Ethernet.NET[x]. szIP	NET[x] Адрес IP.
%QB(n+409+([x]-1)*139)	STRING (15)	Ethernet.NET[x]. szMask	NET[x] Маска подсети.
%QB(n+425+([x]-1)*139)	STRING (15)	Ethernet.NET[x]. szGateway	NET[x] Адрес шлюза.
%QB(n+441+([x]-1)*139)	STRING (17)	Ethernet.NET[x]. szMAC	NET[x] MAC-адрес.
%QB(n+461+([x]-1)*139)	BYTE ARRAY(4)	Ethernet.NET[x]. abyIP	NET[x] Адрес IP.
%QB(n+465+([x]-1)*139)	BYTE ARRAY(4)	Ethernet.NET[x]. abyMask	NET[x] Маска подсети.
%QB(n+469+([x]-1)*139)	BYTE ARRAY(4)	Ethernet.NET[x]. abyGateway	NET[x] Адрес шлюза.

## 6. ОБСЛУЖИВАНИЕ

<b>%QB(n+473+([x]-1)*139)</b>	BYTE ARRAY( 6)	Ethernet.NET[x]. abyMAC	NET[x] MAC-адрес.
<b>%QW(n+479+([x]-1)*139)</b>	WORD	Ethernet.NET[x]. NICteaming	Информация о статусе NICteaming порта NET[x].
<b>%QD(n+481+([x]-1)*139)</b>	DWORD	Ethernet.NET[x] .dwTransmittedBytes	Счетчик байтов, отправленных через порт NET[x] (от 0 до 4294967295).

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QD(n+485+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwTransmittedPackets	Счетчик пакетов, отправленных через порт NET[x] (от 0 до 4294967295).
%QD(n+489+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwTransmittedDropErrors	Счетчик потери соединения при передаче через порт NET[x] (от 0 до 4294967295).
%QD(n+493+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwTransmittedCollisionErrors	Счетчик ошибок при передаче через порт NET[x] (от 0 до 4294967295).
%QD(n+497+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwReceivedBytes	Счетчик байтов, полученных через порт NET[x] (от 0 до 4294967295).
%QD(n+501+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwReceivedPackets	Счетчик пакетов, полученных через порт NET[x] (от 0 до 4294967295).
%QD(n+505+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwReceivedDropErrors	Счетчик потери соединения при приеме через порт NET[x] (от 0 до 4294967295).
%QD(n+509+([x]-1)*139)	DWORD	Ethernet.NET[x]. dwReceivedFrameErrors	Счетчик ошибок кадров при приеме через порт NET[x] (от 0 до 4294967295).

Таблица 196: NET[x] Подробная диагностика группы Ethernet

**Примечание:**

**Прямое представление:** «n» — это адрес, указанный в поле «Начальный адрес» %Q на экране «Конфигурация PCU». «[x]» — это номер интерфейса Ethernet ЦП, где, например, NET 1 представляет значение 1.

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QB(n+807)	BYTE	UserFiles. byMounted	Указывает, может ли память, используемая для записи пользовательских файлов, принимать данные.
%QD(n+808)	DWORD	UserFiles. dwFreeSpaceKB	Свободное место в памяти файлов пользователя в килобайтах.
%QD(n+812)	DWORD	UserFiles. dwTotalSizeKB	Емкость памяти файлов пользователя в Кбайтах.

Таблица 197: Пользовательские файлы группы подробной диагностики

**Примечание:**

**Пользовательский раздел:** Пользовательский раздел представляет собой область памяти, зарезервированную для хранения данных в CPU. Например: файлы с расширением PDF, файлы с расширением DOC и другие данные.

Прямое представление	Размер	AT DG_Modulo.tDetailed.* variable	Описание
%QB(n+817)	BYTE	UserLogs. byMounted	Состояние памяти, в которую вставлены пользовательские журналы.
%QW(n+818)	WORD	UserLogs. wFreeSpaceKB	Свободное место в памяти журнала пользователя в килобайтах.
%QW(n+820)	WORD	UserLogs. wTotalSizeKB	Пользователь регистрирует объем памяти в килобайтах.

Таблица 198: Журналы пользователей группы подробной диагностики

Прямое представление	Размер	AT DG_Module.tDetailed.* variable	Описание
%QB(n+823)	BYTE	MemoryCard. byMounted	Состояние карты памяти: 00: карта памяти не установлена 01: Карта памяти вставлена и установлена
%QX(n+824).0	BIT	MemoryCard. bMemcardtoCPUEnabled	Уровень защиты карты памяти: Чтение данных с карты памяти авторизованным процессором.
%QX(n+824).1	BIT	MemoryCard. bCPUtoMemcardEnabled	Запись данных на карту памяти авторизованным ЦП.
%QD(n+825)	DWORD	MemoryCard. dwFreeSpacekB	Свободное место на карте памяти в килобайтах.
%QD(n+829)	DWORD	MemoryCard. dwTotalSizekB	Емкость карты памяти в килобайтах.

Таблица 199: Описание группы подробной диагностики карты памяти

Прямое представление	Размер	AT DG_Module.tDetailed.* Variable	Описание
%QB(n+833)	BYTE	WHSB. byHotSwapAndStartupStatus	Сообщает о нештатной ситуации в шине, вызвавшей остановку приложения для каждого режима горячей замены. См. Таблицу 201 для получения дополнительной информации.
%QD(n+835)	DWORD ARRAY (32)	WHSB. adwRackIOErrorStatus	Выявление ошибок в модулях ввода/вывода по отдельности. Дополнительные сведения об этой диагностике см. в примечаниях ниже.
%QD(n+963)	DWORD ARRAY (32)	WHSB. adwModulePresenceStatus	Статус наличия заявленных модулей ввода/вывода в шинах по отдельности. Дополнительные сведения об этой диагностике см. в примечаниях ниже.
%QB(n+1091)	BYTE	WHSB. byWHSBBusErrors	Счетчик отказов в шине WHSB. Этот счетчик перезапускается при подаче питания (от 0 до 255).

Таблица 200: Описание группы подробной диагностики WHSB

**Заметки:**

**Диагностика ошибок модулей шины:** Каждое двойное слово из этого диагностического массива представляет собой стойку, позиции которой представлены битами этих двойных слов. Итак, Bit-0 DWORD-0 эквивалентен нулевой позиции стойки с нулевым адресом. Каждый из этих битов является результатом логической операции ИЛИ между диагностикой Несовместимая конфигурация (bConfigMismatch), отсутствующими модулями (bAbsentModules), замененными модулями (bSwappedModules), модулем с фатальной ошибкой (bModuleFatalError) и рабочим состоянием модуля. в определенном положении.

**Состояние присутствия модуля:** Каждое DWORD из этого диагностического массива представляет собой стойку, позиции которой представлены битами этих DWORD. Итак, бит-0 из DWORD-0 эквивалентен нулевой позиции стойки с нулевым адресом. Таким образом, если модуль присутствует, этот бит будет истинным. Важно отметить, что эта диагностика действительна для всех модулей, кроме блоков питания, ЦП и незадекларированных модулей, например, те, которые не находятся в стойке на соответствующей позиции (бит остается в ложном состоянии).

**Ситуации, в которых приложение останавливается:** Коды для возможных ситуаций, в которых приложение останавливается, можно посмотреть ниже:

Код	Исчисляемый	Описание
00	INITIALIZING	Это состояние представлено, пока другие состояния не готовы.
01	RESET_ТАЙМЕР КОНТРОЛЬНОЙ СИСТЕМЫ	Приложение в режиме остановки из-за сброса аппаратного таймера контрольной системы или времени выполнения, когда не отмечена опция «Запускать пользовательское приложение после сброса Таймера контрольной системы».

## 6. ОБСЛУЖИВАНИЕ

02	ABSENT_MODULES_HOT_SWAP _DISABLED	Приложение в режиме остановки из-за установки диагностики отсутствия модулей, когда для режима горячей замены установлено значение «Отключено» или «Отключено, только для объявленных модулей».
03	CFG_MISMATCH_HOT_SWAP _DISABLED	Приложение в режиме остановки из-за установки диагностики несоответствия конфигурации, когда для режима горячей замены установлено значение «Отключено» или «Отключено, только для объявленных модулей».

Код	Исчисляемый	Описание
04	ABSENT_MODULES_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики «Отсутствующие модули», когда для режима «горячей» замены установлено значение «Включено, с согласованностью запуска» или «Включено, с согласованностью запуска только для объявленных модулей».
05	CFG_MISMATCH_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики несовместимой конфигурации, когда для режима горячей замены установлено значение «Включено, с согласованностью запуска» или «Включено, с согласованностью запуска только для объявленных модулей».
06	APPL_STOP_ALLOWED_TO_RUN	Приложение в режиме остановки, и все согласования выполнены успешно. Приложение можно перевести в режим работы.
07	APPL_STOP_MODULES_NOT_READY	Приложение в режиме остановки и все соответствия выполнены успешно, но модули ввода/вывода не могут запустить систему. Невозможно перевести приложение в режим выполнения.
08	APPL_STOP_MODULES_GETTING_READY_TO_RUN	Приложение в режиме остановки, и все согласования выполнены успешно. Модули ввода/вывода готовятся к запуску системы. Невозможно перевести приложение в режим выполнения.
09	NORMAL_OPERATING_STATE	Приложение в режиме выполнения.
10	MODULE_CONSISTENCY_OK	Внутреннее использование.
11	APPL_STOP_DUE_TO_EXCEPTION	Приложение в режиме остановки из-за исключения в ЦП.
12	DUPLICATED_SLOT_HOT_SWAP_DISABLED	Приложение в режиме остановки из-за установки диагностики дублированных слотов, когда режим горячей замены «Отключен» или «Отключен, только для объявленных модулей».
13	DUPLICATED_SLOT_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики дублированных слотов, когда для режима горячей замены установлено значение «Включено, с согласованностью запуска» или «Включено, с согласованностью запуска только для объявленных модулей».
14	DUPLICATED_SLOT_HOT_SWAP_ENABLED	Приложение в режиме остановки из-за установки диагностики дублированных слотов, когда режим горячей замены «Включен, без согласованности при запуске».
15	NON_DECLARED_MODULE_HOT_SWAP_STARTUP_CONSISTENCY	Приложение в режиме остановки из-за установки диагностики недеclared модулей, когда режим горячей замены «включен, с согласованностью запуска».
16	NON_DECLARED_MODULE_HOT_SWAP_DISABLED	Приложение в режиме остановки из-за установки диагностики «Недекларированные модули», когда режим «горячей замены» отключен.

Таблица 201: Коды ситуаций, при которых приложение останавливается

Прямое представление	Размер	AT DG_Module.tDetailed.* Variable	Описание
%QB(n+1092)	BYTE	Application. byCPUState	Информирует о рабочем состоянии CPU: 01: Все пользовательские приложения находятся в режиме выполнения 03: Все пользовательские приложения находятся в режиме остановки
%QX(n+1093).0	BIT	Application. bForcedIOs	Имеется одна или несколько принудительных точек ввода и вывода.

Таблица 202: Описание группы подробной диагностики приложений

Прямое представление	Размер	AT DG variable_Modulo.tDetailed.*	Описание
%QD(n+1094)	DWORD	Rack. dwAbsentRacks	Каждый бит представляет собой идентификационный номер стойки, если какой-либо бит равен TRUE, это означает, что стойка с таким идентификационным номером отсутствует.
%QD(n+1098)	DWORD	Rack. dwDuplicatedRacks	Каждый бит представляет собой идентификационный номер стойки. Если какой-либо бит равен TRUE, это означает, что более одной стойки настроены с одним и тем же идентификационным номером.
%QD(n+1102)	DWORD	Rack. dwNonDeclaredRacks	Каждый бит представляет собой идентификационный номер стойки, если какой-либо бит равен TRUE, это означает, что имеется стойка, сконфигурированная с идентификационным номером, не объявленным в проекте.

Таблица 203: Подробная диагностика группы Rack

Прямое представление	Размер	AT DG variable_Modulo.tDetailed.*	Описание
%QD(n+1108)	DWORD	ApplicationInfo. dwApplicationCRC	32-битный CRC приложения. Когда приложение модифицируется и отправляется в ЦП, вычисляется новый CRC.

Таблица 204: Подробная диагностика группы ApplicationInfo

Прямое представление	Размер	AT DG variable_Modulo.tDetailed.*	Описание
%QX(n+1120).0	BIT	SNTP. bServiceEnabled	Служба SNTP включена.
%QB(n+1121)	BYTE	SNTP. byActiveTimeServer	Указывает, какой сервер активен: 00: Нет активного сервера. 01: Основной сервер активен. 02: Дополнительный сервер активен.
%QW(n+1122)	WORD	SNTP. wPrimaryServerDownCount	Количество раз, когда основной сервер был недоступен (от 0 до 65535).
%QW(n+1124)	WORD	SNTP. wSecondaryServerDownCount	Количество раз, когда дополнительный сервер был недоступен (от 0 до 65535).
%QD(n+1126)	DWORD	SNTP. dwRTCTimeUpdatedCount	Количество обновлений RTC службой SNTP (от 0 до 4294967295).
%QB(n+1130)	BYTE	SNTP. byLastUpdateSuccessful	Указывает статус последнего обновления: 00: Не обновляется. 01: Последнее обновление не удалось. 02: Последнее обновление прошло успешно.
%QB(n+1131)	BYTE	SNTP. byLastUpdateTimeServer	Указывает, какой сервер использовался при последнем обновлении: 00: Нет обновлений. 01: Основной сервер. 02: Дополнительный сервер.
%QB(n+1132)	BYTE	SNTP.sLastUpdateTime. byDayOfMonth	День последнего обновления 24/7.
%QB(n+1133)	BYTE	SNTP.sLastUpdateTime. byMonth	Месяц последнего обновления 24/7.
%QW(n+1134)	WORD	SNTP.sLastUpdateTime. wYear	Год последнего обновления 24/7.
%QB(n+1136)	BYTE	SNTP.sLastUpdateTime. byHours	Час последнего обновления 24/7.
%QB(n+1137)	BYTE	SNTP.sLastUpdateTime. byMinutes	Минута последнего обновления 24/7.

## 6. ОБСЛУЖИВАНИЕ

<b>%QB(n+1138)</b>	BYTE	SNTP.sLastUpdateTime. bySeconds	Второе из последних обновлений 24/7.
<b>%QW(n+1140)</b>	WORD	SNTP.sLastUpdateTime. wMilliseconds	Миллисекунды последнего обновления 24/7.

Таблица 205: Подробная диагностика группы SNTP

Прямое представление	Размер	AT DG_Module.tDetailed.* variable	Описание
%QX(n+1146+[x]*107).0	BIT	RSTP.aBridge.aBridge[x]. bProtocolEnabled	Указывает, включен ли протокол RSTP.
%QB(n+1146+[x]*107.1)	BIT	RSTP.aBridge.aBridge[x]. bBridgeIsRoot	Указывает, является ли мост корневым.
%QB(n+1147+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. eProtocol	Указывает версию протокола.
%QB(n+1148+[x]*107)	STRING (22)	RSTP.aBridge.aBridge[x]. sBridgeId	Идентификатор локального моста (приоритет и MAC).
%QB(n+1171+[x]*107)	STRING (22)	RSTP.aBridge.aBridge[x]. sRootId	Идентификатор корневого моста (приоритет и MAC).
%QB(n+1194+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. eRootPort	Указывает, какой порт является корневым.
%QB(n+1195+[x]*107)	DWORD	RSTP.aBridge.aBridge[x]. dwRootPathCost	Стоимость пути к корню моста.
%QB(n+1199+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. byBridgeHelloTime	Время приветствия настроено
%QB(n+1200+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. byRootHelloTime	Указывает, включен ли протокол RSTP.
%QB(n+1201+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. byBridgeMaxAge	Max Age настроено.
%QB(n+1202+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. byRootMaxAge	Max Age изучено.
%QW(n+1203+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. byBridgeForwDelay	Forward Delay настроено.
%QD(n+1204+[x]*107)	BYTE	RSTP.aBridge.aBridge[x]. byRootForwDelay	Forward Delay изучено.
%QD(n+1205+[x]*107)	DWORD	RSTP.aBridge.aBridge[x]. dwTxHoldCount	Лимит BPDU, передаваемых за цикл.
%QW(n+1209+[x]*107)	DWORD	RSTP.aBridge.aBridge[x]. dwTimeSinceTC	Время в секундах с момента последнего изменения топологии.
%QW(n+1213+[x]*107)	DWORD	RSTP.aBridge.aBridge[x]. dwTCCCount	Сколько раз менялась топология.
%QW(n+1217+[x]*107+[y]*18)	BYTE	RSTP.aBridge.aBridge[x]. aPort[y].eName	Имя порта.
%QW(n+1218+[x]*107+[y]*18)	BYTE	RSTP.aBridge.aBridge[x]. aPort[y].eRole	Функция порта.
%QW(n+1219+[x]*107+[y]*18)	BYTE	RSTP.aBridge.aBridge[x]. aPort[y].eState	Состояние Порты.
%QW(n+1220+[x]*107+[y]*18)	WORD	RSTP.aBridge.aBridge[x]. aPort[y].wIdentity	Идентификатор порта.
%QW(n+1222+[x]*107+[y]*18)	DWORD	RSTP.aBridge.aBridge[x]. aPort[y].dwPathCost	Стоимость порта.
%QW(n+1226+[x]*107+[y]*18).0	BIT	RSTP.aBridge.aBridge[x]. aPort[y].bOperEdge	Указывает, работает ли порт как Edge.
%QW(n+1226+[x]*107+[y]*18).1	BIT	RSTP.aBridge.aBridge[x]. aPort[y].bOperP2P	Указывает, работает ли порт в режиме «точка-точка».
%QW(n+1227+[x]*107+[y]*18)	DWORD	RSTP.aBridge.aBridge[x]. aPort[y].dwNumTxBPDU	Количество кадров BPDU, переданных через порт.
%QW(n+1231+[x]*107+[y]*18)	DWORD	RSTP.aBridge.aBridge[x]. aPort[y].dwNumRxBPDU	Количество кадров BPDU, полученных портом.

Таблица 206: Подробная диагностика RSTP Group

**Примечание:**

Прямое представление: «n» — это адрес, установленный в поле «Начальный адрес» %Q на экране «Конфигурация PCU».  
«[x]» — это номер моста. «[y]» — это номер порта внутри моста.

### 6.1.6. Диагностика через функциональные блоки

Функциональные блоки позволяют визуализировать некоторые параметры, к которым иначе нельзя получить доступ. Функция расширенной диагностики находится в библиотеке NextoStandard и описана ниже.

#### 6.1.6.1. GetTaskInfo

Эта функция возвращает информацию о задаче конкретного приложения.



Рисунок 180: Функция GetTaskInfo

Ниже описаны параметры, которые необходимо передать функции, чтобы она вернула информацию о приложении.

Внутренний параметр	Тип	Описание
psAppName	POINTER TO STRING	Имя приложения.
psTaskName	POINTER TO STRING	Название задачи.
pstTaskInfo	POINTER TO stTask-Info	Указатель для получения информации о приложении.

Таблица 207: Внутренние параметры GetTaskInfo

Данные, возвращаемые функцией, через указатель, указанный во входных параметрах, описаны на Таблице ниже.

Возвращаемые параметры	Размер	Описание
dwCurScanTime	DWORD	Время цикла задачи (выполнение) с разрешением 1 мкс.
dwMinScanTime	DWORD	Минимальное время цикла задачи с разрешением 1 мкс.
dwMaxScanTime	DWORD	Максимальное время цикла задачи. Разрешение 1 мкс.
dwAvgScanTime	DWORD	Среднее время цикла задачи с разрешением 1 мкс.
dwLimitMaxScan	DWORD	Максимальное время цикла задачи до появления сторожевого таймера.
dwIECCycleCount	DWORD	Счетчик циклов IEC.

Таблица 208: Внешние параметры GetTaskInfo

Возможный ERROR CODE:

NoError: успешное выполнение;

TaskNotPresent: нужной задачи не существует.

Пример использования на языке ST:

```

PROGRAM UserPrg
VAR

sAppName : STRING;

psAppName : POINTER TO STRING;

sTaskName : STRING;

psTaskName : POINTER TO STRING;

pstTaskInfo : POINTER TO stTaskInfo;
TaskInfo : stTaskInfo;

Info : ERRORCODE;
END_VAR

// ВХОДЫ:

sAppName := 'Application';    // Переменная получает имя приложения.
psAppName := ADR(sAppName); // Указатель с названием приложения.

sTaskName := 'MainTask';    // Переменная получает имя задачи.
psTaskName := ADR(sTaskName); // Указатель с названием задачи.
pstTaskInfo := ADR(TaskInfo); // Указатель, который получает
информацию о задаче.

```

## 6.2. Графический экран

Графический дисплей, доступный в этом продукте, имеет важный инструмент для управления процессом, так как через него можно распознать возможные состояния ошибки, активные компоненты или наличие диагностики. Кроме того, вся диагностика, включая модули ввода/вывода, представлена пользователю через графический дисплей. Дополнительные сведения об использовании диагностического ключа и его визуализации см. в разделе «Диагностика одним касанием».

На рисунке ниже можно увидеть доступные символы на графическом дисплее этого продукта, а затем их соответствующие значения.

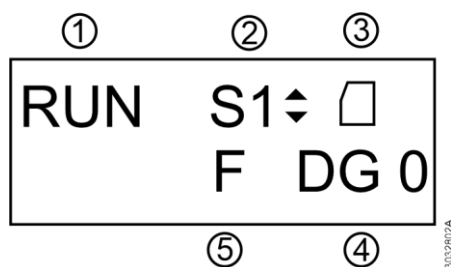


Рисунок 181: Экран состояния процессора

### Расшифровка:

1. Индикация работы состояния ЦП. В случае, если приложение CPU работает, состояние RUN. В случае, если приложение ЦП остановлено, состояние будет STOP, а когда оно остановлено в метке очистки приложения, состояние будет BRKP. Дополнительные сведения см. в разделе «Рабочие состояния процессора».
2. Индикация трафика COM 1. Стрелка вверх (▲) указывает на передачу данных, а стрелка вниз (▼) указывает на прием данных. Дополнительную информацию об интерфейсе COM 1 см. в разделе «Последовательные интерфейсы».
3. Индикация наличия карты памяти. Дополнительные сведения об установке см. в разделе «Установка карты памяти».
4. Индикация количества активных диагностик ЦП. Если отображаемое число отличается от 0 (нуля), значит, в ЦП активна диагностика. Подробнее об их отображении на графическом дисплее ЦП с помощью диагностической клавиши см. в разделе Диагностика одним касанием.

## 6. ОБСЛУЖИВАНИЕ

5. Принудительные переменные в индикации CPU. Если на графическом дисплее отображается символ «F», переменная принудительно задается пользователем, будь то символическое, прямое представление или АТ. Для получения дополнительной информации о форсировании переменных см. раздел «Запись и форсирование переменных».

Помимо символов, описанных выше, процессоры Nexto могут отображать на графическом дисплее некоторые сообщения, соответствующие процессу, который выполняется в данный момент.

В таблице ниже представлены сообщения и их соответствующие описания:

Сообщение	Описание
FORMATTING...	Указывает, что процессор форматирует карту памяти.
FORMATTING ERROR	Указывает, что произошла ошибка при форматировании карты памяти процессором.
WRONG FORMAT	Указывает, что формат карты памяти неверен.
INCORRECT PASSWORD	Указывает, что введенный пароль отличается от настроенного пароля.
TRANSFERRING...	Указывает, что проект передается.
TRANSFERRING ERROR	Указывает на то, что при переносе проекта произошла ошибка, вызванная какой-либо проблемой с картой памяти или ее удалением во время переноса.
TRANSFERRING COMPLETE	Указывает, что передача была выполнена успешно.
TRANSFERRING TIMEOUT	Указывает на тайм-аут (время связи истекло) во время переноса проекта.
CPU TИП MISMATCH	Указывает, что модель ЦП отличается от модели, настроенной в проекте на карте памяти.
VERSION MISMATCH	Указывает, что версия процессора отличается от версии, настроенной в проекте на карте памяти.
APPLICATION CORRUPTED	Указывает, что приложение на карте памяти повреждено.
APPLICATION NOT FOUND	Указывает, что на карте памяти нет приложения для передачи в ЦП.
CRC NOT FOUND	Указывает, что приложение CRC не существует.
MCF FILE NOT FOUND	Указывает на отсутствие файла MCF на карте памяти.
NO TAG	В MasterTool IEC XE нет тега conRiseynokd для процессора.
NO DESC	В MasterTool IEC XE нет контекстного описания для процессора.
MSG. ERROR	Указывает на наличие ошибок в диагностических сообщениях запрошенного модуля (модулей).
SIGNATURE MISSING	Указывает, что продукт вызвал непредвиденную проблему. Свяжитесь с сектором технической поддержки Altus.
APP. ERROR RESTARTING	Указывает, что в приложении произошла ошибка, и среда выполнения перезапускает приложение.
APP. NOT LOADED	Указывает, что среда выполнения не будет загружать приложение.
LOADING APP.	Указывает, что среда выполнения загрузит приложение.
WRONG SLOT	Указывает, что ЦП находится в неправильном положении в стойке.
FATAL ERROR	Указывает на наличие серьезных проблем при запуске ЦП, таких как неправильно смонтированные разделы ЦП. Пожалуйста, свяжитесь со службой поддержки Altus.
HW-SW MISMATCH	Указывает, что аппаратное и программное обеспечение ЦП несовместимы, так как продукт вызвал непредвиденную проблему. Пожалуйста, свяжитесь со службой поддержки Altus.
UPDATING FIRMWARE	Указывает, что микропрограмма обновляется в ЦП.

Сообщение	Описание
RECEIVING FIRMWARE	Указывает, что файл обновления передается в ЦП.
UPDATED	Показывает обновленную версию микропрограммы в ЦП.
UPDATE ERROR	Указывает, что во время обновления микропрограммы ЦП произошла ошибка, вызванная сбоем связи или проблемами конфигурации.
REBOOTING SYSTEM...	Указывает, что ЦП перезагружается для обновления

Таблица 209: Другие сообщения графического дисплея

### 6.3. Системный журнал

Системный журнал — это функция, доступная в программаторе MasterTool IEC XE. Это важный инструмент для управления процессом, так как он позволяет находить события на ЦП, которые могут указывать на состояние ошибки, наличие активных компонентов или активную диагностику. Такие события можно просматривать в хронологическом порядке с точностью до миллисекунд, с объемом хранения до одной тысячи записей журнала, хранящихся во внутренней памяти ЦП, которые невозможно удалить.

Чтобы получить доступ к этим журналам, просто перейдите в дерево устройств и дважды щелкните устройство, затем перейдите на вкладку «Журнал», где можно увидеть сотни операций, таких как: максимальное количество циклов задач, доступ пользователей, онлайн-изменения, загрузка приложений. и загрузка, синхронизация приложений между ЦП, обновление прошивки между другими событиями и действиями.

Чтобы просмотреть журналы, просто нужно подключиться к ЦП (выбран активный путь) и нажать . При нажатии этой кнопки журналы отображаются и мгновенно обновляются. Когда кнопка не нажата, журналы будут удерживаться на экране, это означает, что эта кнопка имеет две стадии: одна удерживает состояние журнала при обновлении, а состояние обновления отключено.

Журналы можно фильтровать по четырем типам: предупреждения, ошибки, исключения и информация. Еще один способ отфильтровать сообщения, отображаемые пользователю, — это выбрать компонент для просмотра.

Отметка времени на вкладке «Журнал» отображается MasterTool после информации, предоставленной устройством (ЦП). MasterTool может отображать отметку времени по местному времени (время компьютера) или по Гринвичу, если установлен флажок времени UTC

Для получения дополнительной информации о системных журналах см. Руководство пользователя MasterTool IEC XE — MU299609 и подраздел «Часы RTC и синхронизация времени» данного руководства.

#### ВНИМАНИЕ

Системные журналы ЦП серии Nexto, начиная с версии прошивки 1.4.0.33, перезагружаются в случаях перезапуска ЦП или перезагрузки системы выполнения, то есть можно будет проверить более старые журналы, когда один таких ситуаций возникает.

### 6.4. Не загружается приложение при запуске

При необходимости пользователь может не загружать существующее приложение на ЦП во время его запуска. Просто включите ЦП с нажатой кнопкой диагностики и удерживайте ее до тех пор, пока не появится сообщение «APP. НЕ ЗАГРУЖЕНО». При попытке входа в систему программное обеспечение MasterTool IEC XE укажет, что в ЦП нет приложения. Для перезагрузки приложения необходимо перезагрузить ЦП или выполнить новую загрузку приложения.

### 6.5. Общие проблемы

Если при включении ЦП он не работает, необходимо проверить следующие пункты: Температура в помещении находится в пределах поддерживаемого устройством диапазона?

На блок питания стойки подается правильное напряжение?

Вставлен ли модуль питания в крайний левый угол стойки (если смотреть на стойку спереди), за которым следует ЦП серии Nexto?

Имеются ли сетевые устройства, такие как концентраторы, коммутаторы или маршрутизаторы, запитанные, взаимосвязанные, подключенные и работающие должным образом? Правильно ли подключен сетевой кабель Ethernet к порту NET 1 или NET 2 процессора Nexto и к сетевому устройству? ЦП Nexto Series включен, находится в режиме выполнения (Run) и не проводит диагностику оборудования?

Если ЦП Nexto указывает режим выполнения (Выполнение), но не отвечает на запрошенные сообщения, будь то через MasterTool IEC XE или протоколы, необходимо проверить следующие элементы:

## 6. ОБСЛУЖИВАНИЕ

Конфигурация параметров CPU Ethernet правильная?

Правильно ли настроен соответствующий протокол связи в ЦП? Правильно ли включены переменные, которые разрешают отношения MODBUS?

Если проблема не обнаружена, обратитесь в службу технической поддержки Altus.

### 6.6. Поиск неисправностей

Таблица ниже показывает симптомы некоторых проблем с их возможными причинами и решениями. Если проблема не устранена, обратитесь в службу технической поддержки Altus.

Симптом	Возможная причина	Решение
Не включается	Отсутствие питания или неправильное питание.	Убедитесь, что ЦП правильно подключен к стойке. Выключите питание и снимите все модули с шины, кроме источника питания и процессора. Включите питание шины и проверьте работу источника питания, внешнего и в стойке. Проверьте, попадает ли напряжение питания на контакты источника питания Nexto и правильно ли оно поляризовано.
Экран CPU показывает сообщение WRONG SLOT	ЦП в неправильном положении.	ЦП должен занимать слоты 0 и 1 стойки 0. Установите его в правильное положение, выключите и снова включите.
	Несовместимая модель стойки.	Используйте в проекте/приложении стойку, совместимую со стойкой конфигурации.
	УСР или стойка повреждены.	Замените ЦП или модуль стойки и проверьте, сохраняется ли проблема.
Не коммуницирует	Плохой контакт или неправильная конфигурация.	Проверьте каждое соединение кабеля связи. Проверьте конфигурацию последовательного интерфейса и интерфейса Ethernet в программе MasterTool IEC XE.

Таблица 210: Поиск неисправностей

### 6.7. Профилактика

- Необходимо ежегодно проверять, надежно ли подключены соединительные кабели, без скопления пыли, в основном защитные устройства.
- В средах, подверженных чрезмерному загрязнению, оборудование необходимо периодически очищать от пыли, мусора и т.п.
- Диоды TVS, используемые для защиты от переходных процессов, вызванных атмосферными разрядами, должны периодически проверяться, так как они могут быть повреждены или разрушены, если поглощенная энергия превышает предел. Во многих случаях неисправность может быть не визуальной. В критических случаях рекомендуется периодическая замена диодов TVS, даже если они не показывают визуальных сигналов неисправности.
- Автобусная герметичность и чистота каждые полгода.
- Для получения дополнительной информации см. Руководство по серии Nexto — MU214600.

## 7. Приложения

### 7.1. Управление ключами TLS и сертификатами

В этом разделе рассматривается создание файлов безопасности, сертификатов и ключей с использованием TLS. Сертификаты, прокомментированные ниже, подписаны СА. Этот тип сертификата рассматривает сущность, называемую Центром сертификации (CA), для создания сертификатов. Этот объект может быть официальной службой или простым компьютером. Необходимо только ограничить доступ к ЦС, чтобы избежать нарушения безопасности, поскольку этот объект может генерировать сертификаты для любого устройства. На изображении ниже показано, как каждое устройство взаимодействует с файлами.

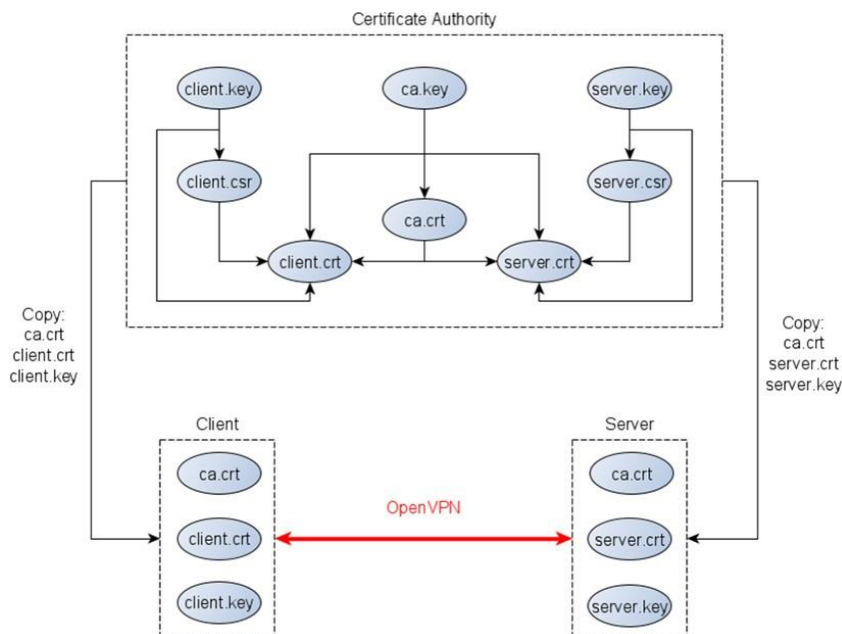


Рисунок 182: Процесс создания сертификата TLS

Прежде всего, сгенерированные файлы являются приватными ключами. Каждое устройство имеет свой ключевой файл, созданный либо объектом ЦС, либо самим устройством. Самый важный файл — закрытый ключ ЦС `ca.key`, который не должен покидать сущность. Объект СА создает свой сертификат на основе своего закрытого ключа `ca.crt`. Этот сертификат является общедоступным файлом, используемым устройствами для проверки VPN-подключения. Для создания сертификатов с устройства сначала требуется файл запроса (`.csr` или `.req` в зависимости от инструмента) на основе закрытого ключа устройства. В этом документе представлены два возможных инструмента для создания файлов сертификатов: Easy-RSA и OpenSSL.

Убедитесь, что дата и время установлены правильно в объекте ЦС, чтобы генерация сертификатов основывалась на текущей настройке.

#### 7.1.1. Создание сертификата Easy-RSA

Проект OpenVPN предоставляет этот инструмент для помощи с сертификатом и ключами. Easy-RSA доступен для Windows и Linux. Ниже приведены пошаговые инструкции по созданию файлов в конфигурации Windows:

1- Откройте приглашение Windows в папке Easy-RSA и запустите `.\Easy-RSA-Start.bat`, чтобы войти в оболочку инструмента.

```

C:\Users\igor.franco\Downloads\EasyRSA-3.0.8-win64\EasyRSA-3.0.8>.\EasyRSA-Start.bat

Welcome to the EasyRSA 3 Shell for Windows.
EasyRSA 3 is available under a GNU GPLv2 license.

Invoke './easyrsa' to call the program. Without commands, help is displayed.

EasyRSA Shell
# _
  
```

Рисунок 183: Генерация сертификата с помощью Easy-RSA (шаг 1)

2 - Скопируйте файл vars.example и переименуйте его в vars в папке инструментов.

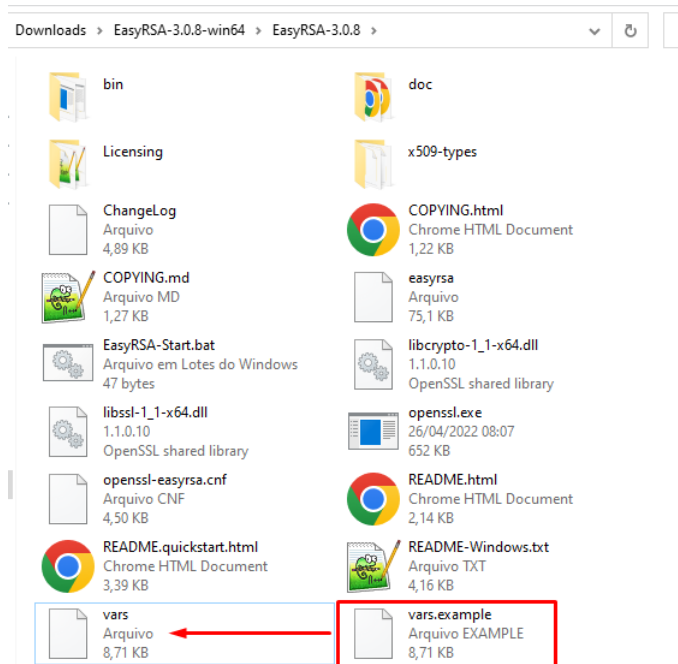


Рисунок 184: Генерация сертификата с помощью Easy-RSA (шаг 2)

3- Откройте файл vars с помощью текстового редактора и измените информацию центра сертификации.

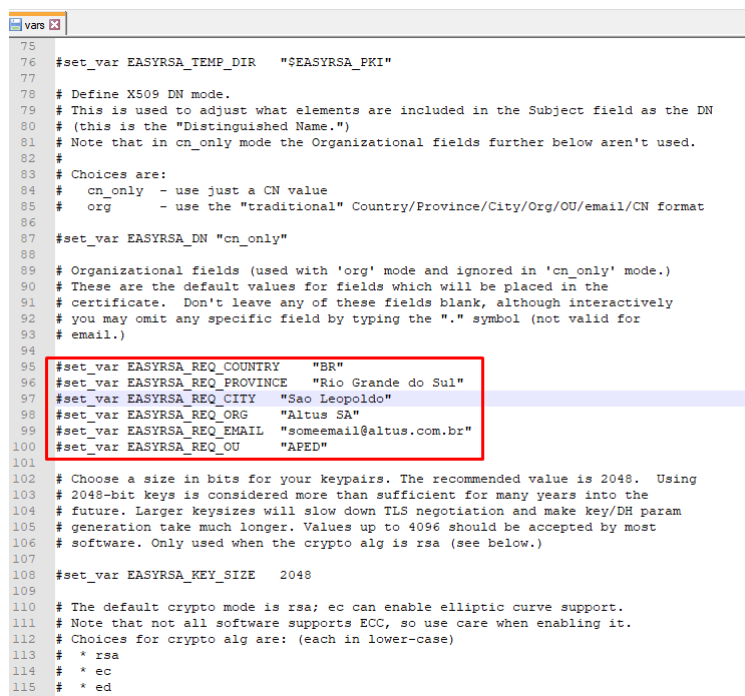


Рисунок 185: Генерация сертификата с помощью Easy-RSA (шаг 3)

4- Используйте команду ./easyrsa init-pki для подготовки конфигурации.

```
# ./easysrsa init-pki

Note: using Easy-RSA configuration from: ./vars
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-13020.a14492/tmp.XXXXXX
lpPathBuffer = C:\Users\IGOR~1.FRA\AppData\Local\Temp\
szTempName = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmpC051.tmp
path = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmpC051.tmp
fd = 3

init-pki complete; you may now create a CA or requests.
Your newly created PKI dir is: C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki

EasyRSA Shell
#
```

Рисунок 186: Генерация сертификата с помощью Easy-RSA (шаг 4)

5- Затем введите `/easysrsa build-ca nopass`, чтобы сгенерировать сертификат ЦС. Удалите аргумент `nopass`, если вы хотите установить пароль для файла. При появлении запроса введите общее имя сертификата ЦС (нажмите Enter, чтобы использовать ЦС Easy-RSA по умолчанию в качестве общего имени).

```
# ./easysrsa build-ca nopass

Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.0j 20 Nov 2018
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-6996.a08916/tmp.XXXXXX
lpPathBuffer = C:\Users\IGOR~1.FRA\AppData\Local\Temp\
szTempName = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmp4FB0.tmp
path = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmp4FB0.tmp
fd = 3
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-6996.a08916/tmp.XXXXXX
lpPathBuffer = C:\Users\IGOR~1.FRA\AppData\Local\Temp\
szTempName = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmp505C.tmp
path = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmp505C.tmp
fd = 3
Generating RSA private key, 2048 bit long modulus
.....+++++
e is 65537 (0x010001)
path = C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-6996.a08916/tmp.XXXXXX
lpPathBuffer = C:\Users\IGOR~1.FRA\AppData\Local\Temp\
szTempName = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmp5194.tmp
path = C:\Users\IGOR~1.FRA\AppData\Local\Temp\tmp5194.tmp
fd = 3
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:CA-Entity

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
C:/Users/igor.franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/ca.crt

EasyRSA Shell
#
```

Рисунок 187: Генерация сертификата с помощью Easy-RSA (шаг 5)

6 - Сгенерируйте ключ устройства и запросите файлы с помощью команды `/easysrsa gen-req DeviceName nopass`. Измените `DeviceName` на желаемое общее имя. Снова удалите аргумент `nopass`, чтобы использовать пароль для файла сертификата. При вводе общего имени в качестве аргумента просто нажмите Enter при появлении запроса (красный квадрат).

```
# ./easysrsa gen-req DeviceName nopass

Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.0j 20 Nov 2018
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmp1508.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmp1508.tmp
fd = 3
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmp1696.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmp1696.tmp
fd = 3
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmp1742.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmp1742.tmp
fd = 3
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-16216.a13904/tmp.a02420'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [DeviceName]:
-----
Key pair and certificate request completed. Your files are:
req: C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/reqs/DeviceName.req
key: C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/private/DeviceName.key

EasyRSA Shell
#
```

Рисунок 188: Генерация сертификата с помощью Easy-RSA (шаг 6)

7- Наконец, введите `easysrsa sign-req server DeviceName` для создания сертификата устройства. `DeviceName` — желаемое общее имя, а `server` — тип (используйте `client`, если вы создаете для VPN-клиента)

```
# ./easysrsa sign-req server DeviceName

Note: using Easy-RSA configuration from: ./vars
Using SSL: openssl OpenSSL 1.1.0j 20 Nov 2018

You are about to sign the following certificate.
Please check over the details shown below for accuracy. Note that this request
has not been cryptographically verified. Please be sure it came from a trusted
source or that you have verified the request checksum with the sender.

Request subject, to be signed as a server certificate for 825 days:

subject=
  commonName = DeviceName

Type the word 'yes' to continue, or any other input to abort.
Confirm request details: yes
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmpC79.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmpC79.tmp
fd = 3
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmpF29.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmpF29.tmp
fd = 3
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmp10FE.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmp10FE.tmp
fd = 3
path = C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.XXXXXX
lpPathBuffer = C:/Users/Igor.Franco/AppData/Local/Temp\
szTempName = C:/Users/Igor.Franco/AppData/Local/Temp/tmp11AA.tmp
path = C:/Users/Igor.Franco/AppData/Local/Temp/tmp11AA.tmp
fd = 3
Using configuration from C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/easy-rsa-9368.a06604/tmp.p.10308
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName : ASN.1 12: 'DeviceName'
Certificate is to be certified until Jul 29 12:59:53 2024 GMT (825 days)

Write out database with 1 new entries
Data Base Updated

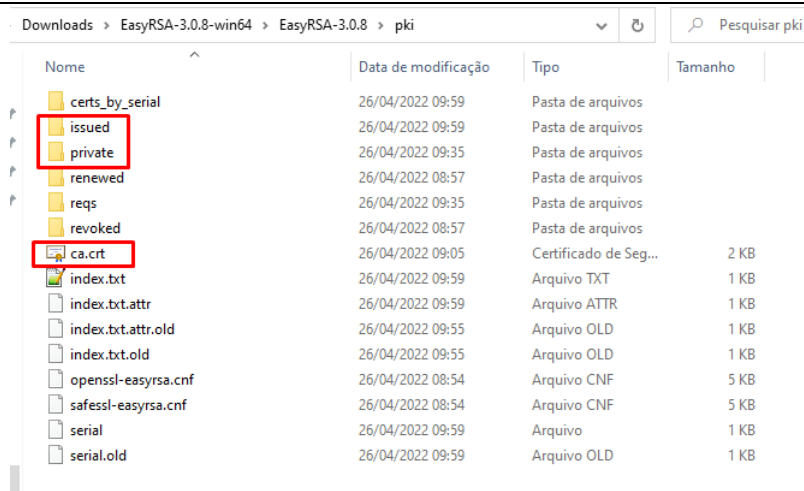
Certificate created at: C:/Users/Igor.Franco/Downloads/EasyRSA-3.0.8-win64/EasyRSA-3.0.8/pki/issued/DeviceName.crt

EasyRSA Shell
#
```

Рисунок 189: Генерация сертификата с помощью Easy-RSA (шаг 7)

8- Повторите шаги 6 и 7, чтобы сгенерировать дополнительные сертификаты устройств.

9- Найдите файл `sa.crt` в папке `pki`, закрытые ключи устройства в пути `pki/private` и сертификаты устройства в каталоге `pki/issued`.



Nome	Data de modificação	Tipo	Tamanho
certs_by_serial	26/04/2022 09:59	Pasta de arquivos	
issued	26/04/2022 09:59	Pasta de arquivos	
private	26/04/2022 09:35	Pasta de arquivos	
renewed	26/04/2022 08:57	Pasta de arquivos	
reqs	26/04/2022 09:35	Pasta de arquivos	
revoked	26/04/2022 08:57	Pasta de arquivos	
ca.crt	26/04/2022 09:05	Certificado de Seg...	2 KB
index.txt	26/04/2022 09:59	Arquivo TXT	1 KB
index.txt.attr	26/04/2022 09:59	Arquivo ATTR	1 KB
index.txt.attr.old	26/04/2022 09:55	Arquivo OLD	1 KB
index.txt.old	26/04/2022 09:55	Arquivo OLD	1 KB
openssl-easyrsa.cnf	26/04/2022 08:54	Arquivo CNF	5 KB
safessl-easyrsa.cnf	26/04/2022 08:54	Arquivo CNF	5 KB
serial	26/04/2022 09:59	Arquivo	1 KB
serial.old	26/04/2022 09:59	Arquivo OLD	1 KB

Рисунок 190: Генерация сертификата с помощью Easy-RSA (шаг 9)

### 7.1.2. Создание сертификата OpenSSL

OpenSSL — это пакет с открытым исходным кодом с инструментами, которые помогают создавать множество файлов и функций безопасности. Этот пакет является родным для большинства дистрибутивов Linux и доступен для Windows. Просто не забудьте установить папку OpenSSL в PATH (переменная среды), чтобы вы могли использовать команду из любого места через приглашение. Найдите ниже пошаговое использование этой функции (все файлы могут иметь любое имя по желанию, шаги рассматриваются только в качестве примера):

- 1- Откройте приглашение в папке сертификата (где вы будете создавать файлы).
- 2- Сгенерируйте закрытый ключ ЦС с помощью следующей команды: `openssl genrsa -out ca.key 4096`.

```
C:\Users\igor.franco\Downloads\Certificate>openssl genrsa -out ca.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
C:\Users\igor.franco\Downloads\Certificate>
```

Рисунок 191: Генерация сертификата с использованием OpenSSL (шаг 2)

- 3- Затем создайте сертификат CA на основе закрытого ключа, используя команду `openssl req -new -x509 -days 365 -key ca.key -out ca.crt`.

Параметр `-days` представляет время истечения срока действия сертификата. Установите его по желанию. В этом примере сертификат действителен в течение одного года. Заполните значения, запрошенные в подсказке по мере необходимости (нажмите Enter, чтобы использовать значение по умолчанию, которое заключено в квадратные скобки []). Необходимо определить общее имя для работы сертификата.

```
C:\Users\igor.franco\Downloads\Certificate>openssl req -new -x509 -days 365 -key ca.key -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:CA-Entity
Email Address []:
C:\Users\igor.franco\Downloads\Certificate>
```

Рисунок 192: Генерация сертификата с использованием OpenSSL (шаг 3)

4- Теперь сгенерируйте закрытый ключ устройства, аналогично шагу 2, используя команду `openssl genrsa -out DeviceName.key 2048`.

```
C:\Users\igor.franco\Downloads\Certificate>openssl genrsa -out DeviceName.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
C:\Users\igor.franco\Downloads\Certificate>
```

Рисунок 193: Генерация сертификата с использованием OpenSSL (шаг 4)

5- После этого сгенерируйте файл запроса сертификата на основе закрытого ключа с помощью команды `openssl req -new -key DeviceName.key -out DeviceName.csr`.

Введите нужную информацию и не забудьте использовать общее имя, отличное от CA.

```
C:\Users\igor.franco\Downloads\Certificate>openssl req -new -key DeviceName.key -out DeviceName.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:DeviceName
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
C:\Users\igor.franco\Downloads\Certificate>
```

Рисунок 194: Генерация сертификата с использованием OpenSSL (шаг 5)

6- Наконец, сгенерируйте сертификат устройства, используя закрытый ключ CA, сертификат CA и файл запроса сертификата устройства, используя `openssl x509 -req -days 365 -in DeviceName.csr -CA ca.crt -CAkey ca.key -set_serial 01` - выход команды.

Установите желаемую дату истечения срока действия с параметром `-days` и серийный номер сертификата с аргументом `-set_serial`.

```
C:\Users\igor.franco\Downloads\Certificate>openssl x509 -req -days 365 -in DeviceName.csr -CA ca.crt -CAkey ca.key -s
et_serial 01 -out DeviceName.crt
Signature ok
subject=C = AU, ST = Some-State, O = Internet Widgits Pty Ltd, CN = DeviceName
Getting CA Private Key
C:\Users\igor.franco\Downloads\Certificate>
```

Рисунок 195: Генерация сертификата с использованием OpenSSL (шаг 6)

7- Повторите шаги с 4 по 6 для любого нового устройства.

8- (Необязательно) OpenSSL предоставляет инструмент для проверки того, что сертификат устройства работает с CA: Используйте команду `openssl verify -purpose sslserver -CAfile ca.crt DeviceName.crt`.

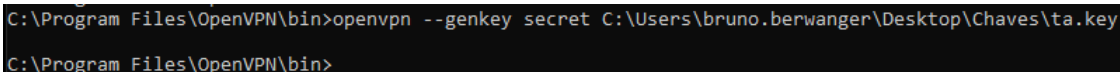
```
C:\Users\igor.franco\Downloads\Certificate>openssl verify -purpose sslserver -CAfile ca.crt DeviceName.crt
DeviceName.crt: OK
```

Рисунок 196: Генерация сертификата с использованием OpenSSL (шаг 8)

### 7.1.3. Генерация ключей TA с помощью OpenVPN

Проект OpenVPN предоставляет инструмент для создания ключа TLS, обычно называемого ta.key. Этот ключ является дополнительным уровнем защиты коммуникационных портов OpenVPN UDP/TCP, поэтому использование этого ключа можно интерпретировать как брандмауэр HMAC для VPN-коммуникации, требующий наличия параметра на обеих сторонах связи для его установки.

Генерацию ключа можно выполнить с помощью следующей команды: `openvpn --genkey secret ta.key`. Пример использования команды в Windows:

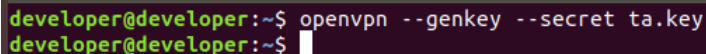


```
C:\Program Files\OpenVPN\bin>openvpn --genkey secret C:\Users\bruno.berwanger\Desktop\Chaves\ta.key
C:\Program Files\OpenVPN\bin>
```

Рисунок 197: Генерация TA-ключа в Windows

Для выполнения команды мы использовали исполняемый файл, установленный с пакетом OpenVPN. Каталог, используемый на изображении выше, является примером и является необязательным. Вы также можете использовать только желаемое имя файла.

Команду можно использовать для создания ключа из Linux, но в команде есть небольшое изменение по сравнению с Windows. Чтобы сгенерировать ключ в Linux, используйте следующую команду: `openvpn --genkey --secret ta.key`. Чтобы запустить его, введите команду в терминале, как показано на примере:



```
developer@developer:~$ openvpn --genkey --secret ta.key
developer@developer:~$
```

Рисунок 198: Генерация ключей TA в Linux

Этот параметр не является обязательным для VPN-связи, но если сервер использует его, все его клиенты также должны его использовать, а ключ для сервера и клиентов должен быть одинаковым.